Coling 2010

23rd International Conference on Computational Linguistics

Posters Volume

Chu-Ren Huang and Dan Jurafsky

23 – 27 August 2010
Beijing International Convention Center
Beijing, China

Produced by
Chinese Information Processing Society of China

To order the CD of Coling 2010 and its Workshop Proceedings, please contact:

Chinese Information Processing Society of China
No.4, Southern Fourth Street
Haidian District, Beijing, 100190
China
Tel: +86-010-62562916
Fax: +86-010-62562916
cips@iscas.ac.cn

Sponsorship

The COLING 2010 very gratefully acknowledges the following commitments in sponsorship:

Platinum Sponsors

-National Natural Science Foundation of China

-Department of Language Information Administration, Ministry of Education, PRC

Gold Sponsor

BaiDu

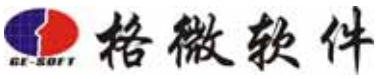
Silver Sponsors

Google

Fujitsu R&D Center CO., LTD.

Microsoft Research

Beijing TRS Information Technology Co., Ltd



Shenyang Globla Envoy software Co.,Ltd.

Supporters

Asian Federation of Natural Language Processing

Institute of Automation
Chinese Academy of Sciences

Institute of Computing Technology
Chinese Academy of Sciences

Institute of Software
Chinese Academy of Sciences

Harbin Institute of Technology

Peking University

Tsinghua University

Preface

You will find in this volume papers from the 23rd International Conference on Computational Linguistics (COLING 2010) held in Beijing, China on August 23-27, 2010 under the auspices of the International Committee on Computational Linguistics (ICCL), and organized by the Chinese Information Processing Society (CIPS) of China. For this prestigious natural language processing conference to be held in China is a significant event for computational linguistics and for colleagues in China, demonstrating both the maturity of our field and the development of academic areas in China.

COLING started as a friendly gathering in New York in 1965, and has grown steadily since. Yet COLINGs aspiration to be a different conference remains the same. COLING strives to maintain its key qualities of embracing different theories and encouraging young scholars in spite of its growing size. A new component introduced at COLING 2010 underlines this quality. A RefreshINGenious (RING) session, organized by Aravind Joshi, our General Chair, allows new and un-orthodox ideas to be presented before they are fully developed in order to generate more discussion and stimulate other new ideas. We hope that this can become an important feature of COLING in the future.

The 155 oral papers included in the hardcopy proceedings published by Tsinghua University Press, as well as the 334 papers included in the electronic proceedings (the same 155 oral papers plus 179 poster papers) are selected from among 815 effective submissions among the more than 840 submissions received. The very selective acceptance rate of 19.02% for oral presentations (155/815 submissions) indicates the extremely high quality of the papers. An additional 21.96% (179/815) are selected for poster presentations to bring the overall acceptance rate to 40.98% (334/815).

We would like to thank the program committee area chairs for their dedicated and efficient review work, and our 738 reviewers for giving us very high quality reviews with a very short turnaround time, allowing us to maintain both the review quality and schedule even given the extraordinary number of submissions. Of course we thank the authors of the 840 papers for submitting their labor of love to COLING. Although we were only able to accept a minority of the submitted papers, we do hope that all authors and reviewers benefit from this process of indirect dialogue. We are especially grateful to the incredibly hard-working team of Stanford volunteers Jenny Finkel, Adam Vogel, and Mengqiu Wang, and HIT volunteers Sam Liang and Lemon Liu, who provided timely and efficient support for the two program chairs at every step of the review and publication processes.

Last but not least, we would like to thank the people who made COLING 2010 and this volume possible. We thank local arrangement committee co-chairs Professor Chengqing Zong and Professor Le Sun for their tireless work which will make COLING-2010 a sure success. Our special appreciation goes to the Chinese Information Processing Society (CIPS) and Professor Youqi Cao for their generous support as the COLING 2010 organizer. Lastly, Professor Qin Lu and Professor Tiejun Zhao should be recognized for their meticulous preparation for editing and publication, which brought this volume to reality.

Chu-Ren Huang and Dan Jurafsky,
COLING 2010 Program Committee Co-chairs

July 8, 2010

COLING 2010 is organized by the Chinese Information Processing Society of China (**CIPS**) and under the auspices of the International Committee on Computational Linguistics (**ICCL**).

General Chair:

Aravind K .Joshi (University of Pennsylvania)

Program Chairs:

Chu-Ren Huang (The Hong Kong Polytechnic University)
Dan Jurafsky (Stanford University)

Advisors to Organizing Committee:

Youqi Cao (The Chinese Information Processing Society of China)
Zhendong Dong (The Chinese Information Processing Society of China)
Changning Huang (Microsoft Research Asia)
Sheng Li (Harbin Institute of Technology)
TianshunYao (Northeastern University)
Shiwen Yu (Peking University)
Zhiwei Feng (Institute of Applied Linguistics, Ministry of Education)
Kaiying Liu (Shanxi University)

OrganizationChairs:

Chengqing Zong (Institute of Automation, Chinese Academy of Sciences)
Le Sun (Institute of Software, Chinese Academy of Sciences)

Publication Chairs:

Qin Lu (The Hong Kong Polytechnic University)
Tiejun Zhao (Harbin Institute of Technology)

Tutorial Chairs:

Dan Gildea (University of Rochester)
Xuanjing Huang (Fudan University)

Workshop Chairs:

Noah Smith (Carnegie Mellon University)
Takenobu Tokunaga (Tokyo Institute of Technology)
Haifeng Wang (BaiDu)

Publicity Chairs:

Hal Daumé III (University of Utah)
Bin Wang (Institute of Computing Technology, Chinese Academy of Sciences)
Minghui Dong (Institute for Infocomm Research)
Monica Monachini (Institute for Computational Linguistics)

Aline Villavicencio (Federal University of Rio Grande do Sul)

Sponsorship Chairs:

Tingting He (Huazhong Normal University)
Hinrich Schütze (University of Stuttgart)
Key-Sun Choi (Korea Advanced Institute of Science and Technology)
Rion Snow (Stanford University)
Takehito Utsuro (University of Tsukuba)
Renata Vieira (Pontifical Catholic University of Rio Grande do Sul)
Hao Yu (Fujitsu(China) R&D Center)

Demo Chairs:

Ting Liu (Harbin Institute of Technology)
Yang Liu (The University of Texas at Dallas)

Program Committee Members/AreaChairs:

Nianwen Xue (Brandeis University)
Rajeev Sangal (India Institute of Information Technology)
Roger Levy (University of California, San Diego)
Justine Cassell (Northwestern University)
Caroline Sporleder (Saarland University)
Gary Geunbae Lee (Pohang University of Science and Technology)
Rosie Jones (Yahoo! Research)
Nicoletta Calzolari (Istituto di Linguistica Computazionale)
Chris Callison-Burch (Johns Hopkins University)
Qun Liu (Institute of Computing Technology)
Pierre Isabelle (National Research Council of Canada)
Sadao Kurohashi (Kyoto University)
Sebastian Padó (Universität Stuttgart)
Takenobu Tokunaga (Tokyo Institute of Technology)
Bo Pang (Yahoo! Research)
Haizhou Li (Institute for Infocomm Research)
David A Smith (University of Massachusetts, Amherst)
Donia Scott (University of Sussex)
Emily Bender (University of Washington)
Ming Zhou (Microsoft Research Asia)

Organization Committee Members:

Juanzi Li (Tsinghua University)
KWONG Olivia (Hong Kong City University)
Bin Sun (Peking University)
Houfeng Wang (Peking University)
Xiaojie Wang (Beijing University of Posts and Telecommunications)
Endong Xun (Beijing Language and Culture University)
Erhong Yang (Beijing Language and Culture University)
Jun Zhao (Institute of Automation, Chinese Academy of Sciences)
Jingbo Zhu (Northeastern University)

Program Committee Members/Reviewers:

Rob Abbott
Takeshi Abekawa
Omri Abend
Steven Abney
Eytan Adar
Guadalupe Aguado
Khalid Al-Kofahi
Cecilia Ovesdotter Alm
Omar Alonso
Bharat Ram Ambati
Massih-Reza Amini
Xiangdong An
Sophia Ananiadou
Jaime Arguello
Masayuki Asahara
Nicholas Asher
Tania Avgustinova
Necip Fazil Ayan
Lakshmi Bai
Jason Baldridge
Timothy Baldwin
Carmen Banea
Srinivas Bangalore
Colin Bannard
Ken Barker
Marco Baroni
Regina Barzilay
John Bateman
Beata Beigman Klebanov
Daisuke Bekki
Nria Bel
Yinon Bentor
Sabine Bergler
Aditya Bhargava
Pushpak Bhattacharyya
Timothy Bickmore
Klinton Bicknell
Alexandra Birch
Philippe Blache
Alan Black
John Blitzer
Michael Bloodgood
Phil Blunsom
Ondrej Bojar
Gemma Boleda
Francis Bond
Johan Bos
Marisa Boston
Pierrette Bouillon
Julien Bourdaillet
Jordan Boyd-Graber
S. R. K. Branavan
Antonio Branco
Eric Breck
Chris Brew
Sabine Buchholz
Stefan Buettcher
Hung Bui
Razvan Bunescu
Aljoscha Burchardt
Stephan Busemann
Miriam Butt
William Byrne
Lynne Cahill
Jamie Callan
Bin Cao
Yunbo Cao
Guiseppe Carenini
Jean Carletta
Marine Carpuat
Xavier Carreras
John Carroll
Ben Carterette
Francisco Casacuberta
Steve Cassidy
Eric Castelli
Alexandru Ceausu
Nick Cercone
Jeong-Won Cha
Vineet Chaitanya
Yllias Chali
Nate Chambers
Baobao Chang
Pi-Chuan Chang
Wanxiang Che
Ciprian Chelba
Aitao Chen
Bin Chen
Boxing Chen
Chien Chin Chen
Harr Chen
Hsin-Hsi Chen
Jinying Chen
Keh-Jiann Chen
Wenliang Chen
Xiao Chen
Charibeth K. Cheng
Colin Cherry
Tee Kiah Chia
David Chiang
Yejin Choi
Kenneth Church
Massimiliano Ciaramita
Philipp Cimiano
Shay Cohen
Kevyn Collins-Thompson
John Conroy
Bonaventura Coppola
Ed Cormany
Dan Cristea
Bruce Croft
Dick Crouch
Montse Cuadros
Silviu-Petru Cucerzan
Oliver Čulo
Aron Culotta
Beatrice Daille
Hercules Dalianis
Sandipan Dandapat
Hoa Dang
Dana Dannells
Dipanjan Das
Sajib Dasgupta
Hal Daume
Dmitry Davidov
Adria de Gispert
Eric De La Clergerie
Valeria de Paiva
Maarten de Rijke
Thierry Declerck
Vera Demberg
Steve DeNeefe
John DeNero
Pascal Denis
Mona Diab
Georgiana Dinu
Pinar Donmez
Iustin Dornescu
Ascander Dost
Antoine Doucet
Doug Downey
Markus Dreyer
Gregory Druck
Xiangyu Duan
Amit Dubey
Kevin Duh
Michael Dukes
Chris Dyer
Marc Dymetman
Markus Egg

Koji Eguchi
David Eichmann
Andreas Eisele
Jacob Eisenstein
Jason Eisner
Noemie Elhadad
Charles Elkan
Micha Elsner
Alexandre entiev
Katrin Erk
Andrea Esuli
Richard Evans
Roger Evans
Patrick Fan
Alex Fang
Benoit Favre
Marcello Federico
Christiane Fellbaum
Katja Filippova
Radu Florian
Sandiway Fong
George Foster
Gil Francopoulo
Stefan Frank
Alex Fraser
Marjorie Freedman
Maria Fuentes
Hagen Fuerstenau
Atsushi Fujii
Sumio Fujita
Pascale Fung
Ryan Gabbard
Evgeniy Gabrilovich
Karthik Gali
Michel Galley
Michael Gamon
Kuzman Ganchev
Jianfeng Gao
Albert Gatt
Eric Gaussier
Ruifang Ge
Byron Georgantopoulos
Pablo Gervas
Sanjukta Ghosh
Daniel Gildea
Dan Gillick
Jesus Gimenez
Kevin Gimpel
Roxana Girju
Sharon Goldwater
Tianxia Gong
Julio Gonzalo
Cyril Goutte
Brigitte Grau
Stephan Greene
Mark Greenwood
Gregory Grefenstette
Ralph Grishman
Cecile Grivaz
Jiafeng Guo
Ben Hachey
Aria Haghighi
Udo Hahn
Jan Hajic
Eva Hajicova
Dilek Hakkani-tur
John Hale
David Hall
Greg Hanneman
Max Harper
Chikara Hashimoto
Ahmed Hassan
Claudia Hauff
Yoshihiko Hayashi
Ben He
daan he
Daqing He
Xiaodong He
Yulan He
John Henderson
Iris Hendrickx
Amac Herdagdelen
Ulf Hermjakob
Raquel Hervas
Dirk Heylen
Andrew Hickl
Graeme Hirst
Jerry Hobbs
Katja Hofmann
Steven Chu Hong Hoi
Kristy Hollingshead
Mark Hopkins
Eric Horvitz
Veronique Hoste
Yunhua Hu
Jimmy Huang
XuanJing Huang
Yifen Huang
Yun Huang
Matt Huenerfauth
Juan Huete
Sarmad Hussain
Rebecca Hwa
Gonzalo Iglesias
Steven Ikier
Diana Inkpen
Kentaro Inui
Elena Irimia
Amy Isard
Abe Ittycheriah
Tatsuya Izuha
Adam Jatowt
Jiwoon Jeon
Heng Ji
Sittichai Jiampojamarn
Daxin Jiang
Hongfei Jiang
Jiepu Jiang
Jing Jiang
Long Jiang
Wenbin Jiang
Valentin Jijkoun
Richard Johansson
Howard Johnson
Rie Johnson
Kristiina Jokinen
Doug Jones
Rosie Jones
Hanmin Jung
Vijay K. Shanker
Min-Yen Kan
Hiroshi Kanayama
Damianos Karakos
Nikiforos Karamanis
Rohit Kate
Tsuneaki Kato
Daisuke Kawahara
Tatsuya Kawahara
Junichi Kazama
Shahram Khadivi
Chloe Kiddon
Jin-Dong Kim
Min Kim
Chunyu Kit
Kevin Knight
Youngjoong Ko
Philipp Koehn
Rob Koeling
Dimitrios Kokkinakis
Greg Kondrak

Moshe Koppel
Valia Kordoni
Lili Kotlerman
Eric Kow
Zornitsa Kozareva
Emiel Kramer
Steven Krauwer
Gerhard Kremer
Canasai Kruengkrai
Lun-Wei Ku
Taku Kudo
Jonas Kuhn
Roland Kuhn
Peter Khnlein
Amba Kulkarni
Ravi Kumar
Shankar Kumar
A Kumaran
Oren Kurland
K. L. Kwok
Olivia Kwong
Wai Lam
Man Lan
Philippe Langlais
Francois Lareau
Martha Larson
Alex Lascarides
Alberto Lavelli
Alon Lavie
Florian Laws
Guy Lebanon
Changki Lee
John Lee
Joo Young Lee
Lillian Lee
Seungwoo Lee
Tan Lee
Jochen Leidner
Yves Lepage
Kevin Lerman
James Lester
Gregor Leusch
Gina-Anne Levow
Hang Li
Jiye Li
Lei Li
Mu Li
Shiqi Li
Shoushan Li
Wei Li
Wen Jie Li
Zhifei Li
Dekang Lin
Lucian Vlad Lita
Diane Litman
Bing Liu
Shui Liu
Ting Liu
Xiaohua Liu
Yan Liu
Yang Liu
Yang Liu
Yupeng Liu
Zhanyi Liu
Zhaopengx Liu
Anna Lobanova
Qiu Long
Adam Lopez
Qin Lu
Yue Lu
Harald Lngen
Xiaoqiang Luo
Tomoya Lwakura
Bin Ma
Qing Ma
Yanjun Ma
Wolfgang Macherey
Nitin Madnani
Hala Maghout
B. Mallikarjun
Gideon Mann
Daniel Marcu
Mitch Marcus
Joseph Mariani
Katja Markert
Erwin Marsi
M. Antonia Marti Antonin
Jean-Claude MARTIN
Andre Martins
Yuval Marton
Yuji Masumoto
Shigeki Matsubara
Tomoko Matsui
Yuji Matsumoto
Takuya Matsuzaki
Evgeny Matusov
Mausam
Arne Mauser
Marshall Mayberry
Diana McCarthy
David McClosky
Ryan McDonald
Tara McIntosh
Kathy McKeown
Susan McRoy
Yashar Mehdad
Qiaozhu Mei
Chris Mellish
Amlia Mendes
Helen Meng
Donald Metzler
Haitao Mi
Jun Miao
Lukas Michelbacher
Jeffrey Micher
Eleni Miltsakaki
David Mimno
Zhaoyan Ming
Shachar Mirkin
Jeff Mitchell
Vibhu Mittal
Yusuke Miyao
Marie-Francine Moens
Dan Moldovan
Diego Moll Aliod
Christof Monz
Raymond J. Mooney
Bob Moore
Roser Morante
Louis-Philippe Morency
Alessandro Moschitti
Isabelle Moulinier
Animesh Mukherjee
Stefan Mller
Art Munson
Dragos Munteanu
Masaki Murata
Vanessa Murdock
Smaranda Muresan
Gabriel Murray
Pradeep Muthukrishnan
Sobha Nair
Testuji Nakagawa
Vivi Nastase
Martina Naughton
Roberto Navigli
Mark-Jan Nederhof
Vasek Nemcik
Ani Nenkova
Hwee Tou Ng

Vincent Ng
Huyen Nguyen Thi Minh
Patrick Nguyen
Alex Niculescu-Mizil
Jian-Yun Nie
Rodney Nielsen
Takashi Ninomiya
Toyoaki Nishida
Cheng Niu
Zheng-Yu Niu
Diarmuid Saghdha
Franz Och
Michael ODonnell
Stephan Oepen
Naoki Okazaki
Lilja Ovrelid
Derya Ozkan
Ulrike Pado
Alexis Palmer
Sinno Jialin Pan
Patrick Pantel
Simone Paolo Ponzetto
Jong Park
Kristen Parton
Marius Pasca
Siddharth Patwardhan
Michael Paul
Soma Paul
Matthias Paulik
Yves Peirsman
Gerald Penn
Marco Pennacchiotti
Bhaskararao Peri
Aina Peris
Wim Peters
Kay Peterson
Slav Petrov
Daniele Pighin
Prasad Pingali
Stelios Piperidis
Guillaume Pitel
Paul Piwek
Luiz Pizzato
Massimo Poesio
Hoifung Poon
Ana-Maria Popescu
Maja Popovic
Christopher Potts
Richard Power
Rashmi Prasad
Laurent Prevot
Matthew Purver
Haoliang Qi
Toh Zhi Qiang
Tao Qin
Yang Qu
Chris Quirk
Bhiksha Raj
S. Rajendran
Dan Ramage
S.V. Ramanan
P. V. S. Rambabu
Owen Rambow
Delip Rao
Ari Rappoport
Paul Rayson
Michaela Regneri
Nils Reiter
Norbert Reithinger
Giuseppe Riccardi
Sebastian Riedel
German Rigau
Hae-Chang Rim
Lucia Rino
Hammam Riza
Horacio Rodrłguez
Gerhard Rolletschek
Lorenza Romano
Mathias Rossignol
Antti-Veikko Rosti
Michael Roth
Alex Rudnick
Marta Ruiz
Kenji Sagae
Horacio Saggion
Harabagiu Sanda
Baskaran Sankaran
Ratna Sanyal
Murat Saraclar
Sudeshna Sarkar
Manabu Sassano
Sekine Satoshi
Roser Sauri
Helmut Schmid
William Schuler
Sabine Schulte im Walde
Ineke Schuurman
Lane Schwartz
Wolfgang Seeker
sekine Sekine
Stephanie Seneff
Violeta Seretan
Hendra Setiawan
Chirag Shah
Dipti Sharma
Dou Shen
Libin Shen
Wade Shen
Kiyoaki Shirai
Eyal Shnarch
Advaith Siddharthan
Candy Sidner
Khalil Simaan
Michel Simard
Kiril Simov
Anil Kumar Singh
Smriti Singh
Samar Sinha
Sharon Small
Jason Smith
Nathaniel Smith
Matthew Snover
Benjamin Snyder
Stephen Soderland
Swapna Somasundaran
Yan Song
Young-In Song
Virach Sornlertlamvanich
Lucia Specia
Jennifer Spenader
Valentin Spitkovsky
Richard Sproat
Ed Stabler
Manfred Stede
Mark Steedman
Amanda Stent
Mark Stevenson
Matthew Stone
Svetlana Stoyanchev
Veselin Stoyanov
Kristina Striegnitz
Michael Strube
Tomek Strzalkowski
Jian Su
Jinsong Su
Keh-Yih Su
Nam Kim Su
Rajen Subba
K. V. Subbarao
L. V. Subramaniam

Zhifang Sui
Eiichiro Sumita
Congkai Sun
Jun Sun
Maosong Sun
Hisami Suzuki
Gyorgy Szarvas
Hiroya Takamura
Koichi Takeuchi
David Talbot
Chew Lim Tan
Jie Tang
Chua Tat-Seng
Paul Tepper
Simone Teufel
Stefan Thater
Mariet Theune
Paul Thomas
Vu Thuy
Joerg Tiedemann
Christoph Tillmann
Ivan Titov
Katrin Tomanek
Yoichi Tomiura
Sara Tonelli
Kentaro Torisawa
Kristina Toutanova
Roy Tromble
Huihsin Tseng
Shu-Chuan Tseng
Benjamin Tsou
Yuta Tsuboi
Jun-ichi Tsujii
Koji Tsukamoto
Yoshimasa Tsuruoka
Dan Tufis
Gokhan Tur
Kiyotaka Uchimoto
Yuya Unno
Lonneke Van der Plas
ielka van der sluis
Josef van Genabith
Gertjan van Noord
Lucy Vanderwende
Sebastian Varges
Sriram Venkatapathy
Marc Verhagen
Yannick Versley
Cristina Vertan
Marta Vila
David Vilar
Aline Villavicencio
Stephan Vogel
Clare Voss
Piek Vossen
Marilyn Walker
Hanna Wallach
Xiaojun Wan
Wenting Wang
Haifeng Wang
Hsin-Min Wang
Kai Wang
Rui Wang
Wei Wang
Xuanhui Wang
Leo Wanner
Nigel Ward
Taro Watanabe
Yotaro Watanabe
Nick Webb
Bonnie Webber
Eric Wehrli
Kilian Weinberger
David Weir
Michael White
Michael Wick
Jan Wiebe
Yorick Wilks
Theresa Wilson
Shuly Wintner
Andreas Witt
Peter Wittek
Magda Wolska
Chung-Hsien Wu
Dan Wu
Dekai Wu
Hua Wu
Xiaoyun Wu
Yunfang Wu
Fei Xia
Rui Xia
Xinyan Xiao
Deyi Xiong
Peng Xu
Ruifeng Xu
Bert Xue
Yongxin Yan
Muyun Yang
Qiang Yang
Ainur Yessenalina
Bei Yu
Jianxing Yu
Kun Yu
Mo Yu
Yisong Yue
Zdenek Zabokrtsky
Taras Zagibalov
Omar Zaidan
Roberto Zanoli
Fabio Zanzotto
Alessandra Zarcone
Richard Zens
Torsten Zesch
Luke Zettlemoyer
Hao Zhang
Hui Zhang
Jiajun Zhang
Min Zhang
Ruiqiang Zhang
Yujie Zhang
Ziqi Zhang
Hai Zhao
Jun Zhao
Bowen Zhou
Guodong Zhou
Yiping Zhou
Conghui Zhu
Jerry Zhu
Tao Zhuang
Heike Zinsmeister
Imed Zitouni
Andreas Zollmann
Chengqing Zong
Ingrid Zukerman

Table of Contents

Author Index

Towards the Adequate Evaluation of Morphosyntactic Taggers

Szymon Acedański
Institute of Computer Science,
Polish Academy of Sciences
Institute of Informatics,
University of Warsaw
accek@mimuw.edu.pl

Adam Przepiórkowski
Institute of Computer Science,
Polish Academy of Sciences
Institute of Informatics,
University of Warsaw
adamp@ipipan.waw.pl

Abstract

There exists a well-established and almost unanimously adopted measure of tagger performance, namely, accuracy. Although it is perfectly adequate for small tagsets and typical approaches to disambiguation, we show that it is deficient when applied to rich morphological tagsets and propose various extensions designed to better correlate with the real usefulness of the tagger.

1 Introduction

Part-of-Speech (PoS) tagging is probably the most common and best researched NLP task, the first step in many higher level processing solutions such as parsing, but also information retrieval, speech recognition and machine translation. There are also well established evaluation measures, the foremost of which is accuracy, i.e., the percent of words for which the tagger assigns the correct — in the sense of some gold standard — interpretation.

Accuracy works well for the original PoS tagging task, where each word is assumed to have exactly one correct tag, and where the information carried by a tag is limited roughly to the PoS of the word and only very little morphosyntactic information, as in typical tagsets for English. However, there are two cases where accuracy becomes less than adequate: the situation where the gold standard and / or the tagging results contain multiple tags marked as correct for a single word, and the use of a rich morphosyntactic (or morphological) tagset.

The first possibility is discussed in detail in (Karwańska and Przepiórkowski, 2009), but the need for an evaluation measure for taggers which do not necessarily fully disambiguate PoS was already noted in (van Halteren, 1999), where the use of standard information retrieval measures precision and recall (as well as their harmonic mean, the F-measure) is proposed. Other natural generalisations of the accuracy measure, able to deal with non-unique tags either in the gold standard[1] or in the tagging results, are proposed in (Karwańska and Przepiórkowski, 2009).

Standard accuracy is less than adequate also in case of rich morphosyntactic tagsets, where the full tag carries information not only about PoS, but also about case, number, gender, etc. Such tagsets are common for Slavic languages, but also for Hungarian, Arabic and other languages. For example, according to one commonly used Polish tagset (Przepiórkowski and Woliński, 2003), the form *uda* has the following interpretations: `fin:sg:ter:perf` (a finite singular 3rd person perfective form of the verb UDAĆ 'pretend'), `subst:pl:nom:n` and

[1] There are cases were it makes sense to manually assign a number of tags as correct to a given word, as any decision would be fully arbitrary, regardless of the amount of context and world knowledge available. For example, in some Slavic languages, incl. Polish, there are verbs which optionally subcategorise for an accusative or a genitive complement, without any variation in meaning, and there are nouns which are syncretic between these two cases, so for such "verb + $\text{noun}_{\text{acc/gen}}$" sequences it is impossible to fully disambiguate case; see also (Oliva, 2001).

Coling 2008: Poster Volume, pages 1–8,
Beijing, August 2010

subst:pl:acc:n (nominative or accusative plural form of the neuter noun UDO 'thigh'). Now, assuming that the right interpretation in a given context is subst:pl:acc:n, accuracy will equally harshly penalise the other nominal interpretation (subst:pl:nom:n), which shares with the correct interpretation not only PoS, but also the values of gender and number, and the completely irrelevant verbal interpretation. A more accurate tagger evaluation measure should distinguish these two non-optimal assignments and treat subst:pl:nom:n as partially correct.

Similarly, the Polish tagset mentioned above distinguishes between nouns and gerunds, with some forms actually ambiguous between these two interpretations. For example, *zadanie* may be interpreted as a nominative or accusative form of the noun ZADANIE 'task', or a nominative or accusative form of the gerund derived from the verb ZADAĆ 'assign'. Since gerunds and nouns have very similar distributions, any error in the assignment of part of speech, noun vs. gerund, will most probably not matter for a parser of Polish — it will still be able to construct the right tree, provided the case is correctly disambiguated. However, the "all-or-nothing" nature of the accuracy measure regards the tag differing from the correct one only in part of speech or in case as harshly, as it would regard an utterly wrong interpretation, say, as an adverb.

In what follows we propose various evaluation measures which differentiate between better and worse incorrect interpretations, cf. § 2. The implementation of two such measures is described in § 3. Finally, § 4 concludes the paper.

2 Proposed Measures

2.1 Full Interpretations and PoS

The first step towards a better accuracy measure might consist in calculating two accuracy measures: one for full tags, and the other only for fragments of tags representing parts of speech. Two taggers wrongly assigning either fin:sg:ter:perf (T1) or subst:pl:nom:n (T2) instead of the correct subst:pl:acc:n would fare equally well with respect to the tag-level accuracy, but T2 would be — rightly — evaluated as better with respect to the PoS-level accuracy.

The second example given in § 1 shows, however, that the problem is more general and that a tagger which gets the PoS wrong (say, gerund instead of noun) but all the relevant categories (case, number, gender) right may actually be more useful in practice than the one that gets the PoS right at the cost of confusing cases (say, accusative instead of nominative).

2.2 Positional Accuracy

A generalisation of the idea of looking separately at parts of speech is to split tags into their components (or positions) and measure the correctness of the tag by calculating the F-measure. For example, if the (perfective, affirmative) gerundial interpretation ger:sg:nom:n:perf:aff is assigned instead of the correct nominal interpretation subst:sg:nom:n, the tags agree on 3 positions (sg, nom, n), so the precision is $\frac{3}{6}$, the recall — $\frac{3}{4}$, which gives the F-measure of 0.6. Obviously, the assignment of the correct interpretation results in F-measure equal 1.0, and the completely wrong interpretation gives F-measure 0.0. Taking these values instead of the "all-or-nothing" 0 or 1, accuracy is reinterpreted as the average F-measure over all tag assignments.

Note that while this measure, let us call it *positional accuracy* (PA), is more fine-grained than the standard accuracy, it wrongly treats all components of tags as of equal importance and difficulty. For example, there are many case syncretisms in Polish, but practically no ambiguities concerning the category of negation (see the value aff above), so case is inherently much more difficult than negation, and also much more important for syntactic parsing, and as such it should carry more weight when evaluating tagging results.

2.3 Weighted Positional Accuracy

In the current section we make a simplifying assumption that weights of positions are absolute, rather than conditional, i.e., that the weight of, say, case does not depend on part of speech, word or context. Once the weights are attained, weighted precision and recall may be used as in the following example.

Assume that PoS, case, number and gender have the same weight, say 2.0, which is 4 times larger than that of any other category. Then, in case `ger:sg:nom:n:perf:aff` is assigned instead of the correct `subst:sg:nom:n`, precision and recall are given by:

$$P = \frac{3 \times 2.0}{4 \times 2.0 + 2 \times 0.5} = \frac{2}{3},$$
$$R = \frac{3 \times 2.0}{4 \times 2.0} = \frac{3}{4}.$$

This results in a higher F-measure than in case of non-weighted positional accuracy.

The following subsections propose various ways in which the importance of particular grammatical categories and of the part of speech may be estimated.

2.3.1 Average Ambiguity

The average number of morphosyntactic interpretations per word is sometimes given as a rough measure of the difficulty of tagging. For example, tagging English texts with the Penn Treebank tagset is easier than tagging Czech or Polish, as the average number of possible tags per word is 2.32 in English (Hajič, 2004, p. 171), while it is 3.65 (Hajič and Hladká, 1997, p. 113) and 3.32 (Przepiórkowski, 2008, p. 44) for common tagsets for Czech and Polish, respectively.

By analogy, one measure of the difficulty of assigning the right value of a given category or part of speech is the average number of different values of the category per word.

2.3.2 Importance for Parsing

All measures mentioned so far are *intrinsic* (*in vitro*) evaluation measures, independent — but hopefully correlated with — the usefulness of the results in particular applications. On the other hand, *extrinsic* (*in vivo*) evaluation estimates the usefulness of tagging in larger systems, e.g., in parsers. Full-scale extrinsic evaluation is rarely used, as it is much more costly and often requires user evaluation of the end system.

In this and the next subsections we propose evaluation measures which combine the advantages of both approaches. They are variants of the weighted positional accuracy (WPA) measure, where weights correspond to the usefulness of a given category (or PoS) for a particular task.

Probably the most common task taking advantage of morphosyntactic tagging is syntactic parsing. Here, weights should indicate to what extent the parser relies on PoS and particular categories to arrive at the correct parse. Such weights may be estimated from an automatically parsed corpus in the following way:

for each category (including PoS) c **do**
  $\mathrm{count}(c) = 0$ {Initialise counts.}
end for
for each sentence s **do**
  for each rule r used in s **do**
    for each terminal symbol (word) t in the RHS of r **do**
      for each category c referred to by r in t **do**
        increase $\mathrm{count}(c)$
      end for
    end for
  end for
end for
{Use $\mathrm{count}(c)$'s as weights.}

In prose: whenever a syntactic rule is used, increase counts of all morphosyntactic categories (incl. PoS) mentioned in the terminal symbols occurring in this rule. These counts may be normalised or used directly as weights.

We assume here that either the parser produces a single parse for any sentence (assumption realistic only in case of shallow parsers), or that the best or at least most probable parse may be selected automatically, as in case of probabilistic grammars, or that parses are disambiguated manually. In case only a non-probabilistic deep parser is available, and parses are not disambiguated manually, the Expectation-Maximisation method may be used to select a probable parse (Dębowski, 2009) or all parses might be taken into account.

Note that, once a parser is available, such weights may be calculated automatically and used repeatedly for tagger evaluation, so the cost of using this measure is not significantly higher than the cost of intrinsic measures, while at the same time the correlation of the evaluation results with the extrinsic application is much higher.

2.3.3 Importance for Corpus Search

The final variant (many more are imaginable) of WPA that we would like to describe here concerns another application of tagging, namely, for the annotation of corpora. Various corpus search engines, including the IMS Open Corpus Workbench (`http://cwb.sourceforge.net/`) and Poliqarp (`http://poliqarp.sourceforge.net/`) allow the user to search for particular parts of speech and grammatical categories. Obviously, the tagger should maximise the quality of the disambiguation of those categories which occur frequently in corpus queries, i.e., the weights should correspond to the frequencies of particular categories (and PoS) in user queries. Note that the only resource needed to calculate weights are the logs of a corpus search engine.

An experiment involving an implementation of this measure is described in detail in § 3.

2.4 Conditional Weighted Positional Accuracy

The importance and difficulty of a category may depend on the part of speech. For example, after case syncretisms, gender ambiguity is one of the main problems for the current taggers of Polish. But this problem concerns mainly pronouns and adjectives, where the systematic gender syncretism is high. On the other hand, nouns do not inflect for gender, so only some nominal forms are ambiguous with respect to gender. Moreover, gerunds, which also bear gender, are uniformly neuter, so here part of speech alone uniquely determines the value of this category.

A straightforward extension of WPA capitalising on these observations is what we call *conditional weighted positional accuracy* (CWPA), where weights of morphosyntactic categories are conditioned on PoS.

Note that not all variants of WPA may be easily generalised to CWPA; although such an extension is obvious for the average ambiguity (§ 2.3.1), it is less clear for the other two variants. For parsing-related WPA, we assume that, even if a given rule does not mention the PoS of a terminal symbol,[2] that PoS may be read off the parse tree, so the conditional weights may still be calculated. On the other hand, logs of a corpus search engine are typically not sufficient to calculate such conditional weights; e.g., a query for a sequence of 5 genitive words occurring in logs would have to be rerun on the corpus again in order to find out parts of speech of the returned 5-word sequences. For a large number of queries on a large corpus, this is a potentially costly operation.

It is also not immediately clear how to generalise precision and recall from WPA to CWPA. Returning to the example above, where t_1 = `ger:sg:nom:n:perf:aff` is assigned instead of the correct t_2 = `subst:sg:nom:n`, we note that the weights of number, case and gender may now (and should, at least in case of gender!) be different for the two parts of speech involved. Hence, precision needs to be calculated with respect to the weights for the automatically assigned part of speech, and recall — taking into account weights for the gold standard part of speech:

$$P = \frac{\delta_{t_1^* t_2^*} w(t_1^*) + \sum_{c \in C(t_1,t_2)} \delta_{t_1^c t_2^c} w(c|t_1^*)}{w(t_1^*) + \sum_{c \in C(t_1)} w(c|t_1^*)},$$
$$R = \frac{\delta_{t_1^* t_2^*} w(t_2^*) + \sum_{c \in C(t_1,t_2)} \delta_{t_1^c t_2^c} w(c|t_2^*)}{w(t_2^*) + \sum_{c \in C(t_2)} w(c|t_2^*)},$$

where t^* is the PoS of tag t, $w(p)$ is the weight of the part of speech p, $w(c|p)$ is the conditional weight of the category c for PoS p, $C(t)$ is the set of morphosyntactic categories of tag t, $C(t_1, t_2)$ is the set of morphosyntactic categories common to tags t_1 and t_2, t^c is the value of category c in tag t, and δ_{ij} is the Kronecker delta (equal to 1 if $i = j$, and to 0 otherwise). Hence, for the example above, these formulas may be simplified to:

$$P = \frac{\sum_{c \in \{n,c,g\}} w(c|\texttt{ger})}{w(\texttt{ger}) + \sum_{c \in \{n,c,g,a,neg\}} w(c|\texttt{ger})},$$
$$R = \frac{\sum_{c \in \{n,c,g\}} w(c|\texttt{subst})}{w(\texttt{subst}) + \sum_{c \in \{n,c,g\}} w(c|\texttt{subst})},$$

where *n*, *c*, *g*, *a* and *neg* stand for number, case, gender, aspect and negation.

[2] For example, in unification grammars and constraint-based grammars a terminal may be identified only by the values of some of its categories, as in the following simple rule, specifying prepositional phrases as a preposition governing a specific case and a non-empty sequence of words bearing that case: $\mathrm{PP}_{\mathrm{case=C}} \rightarrow \mathrm{P}_{\mathrm{case=C}}\ \mathrm{X}^{+}_{\mathrm{case=C}}$.

3 Experiment

To evaluate behaviour of the proposed metrics, a number of experiments were performed using the manually disambiguated part of the IPI PAN Corpus of Polish (Przepiórkowski, 2005). This subcorpus consists of 880 000 segments. Two taggers of Polish were tested. TaKIPI (Piasecki and Godlewski, 2006) is a tagger which was used for automatic disambiguation of the remaining part of the aforementioned corpus. It is a statistical classifier based on decision trees combined with some automatically extracted, hand-crafted rules. This tagger by default sometimes assigns more than one tag to a segment, what is consistent with the golden standard. There is a setting which allows this behaviour to be switched off. This tagger was tested with both settings. The other tagger is a prototype version of this Brill tagger, presented by Acedański and Gołuchowski in (Acedański and Gołuchowski, 2009).

For comparison, four metrics were used: standard metrics for full tags and only parts of speech, as well as Positional Accuracy and Weighted Positional Accuracy. For the last measure, the weights were obtained by analysing logs of user queries of the Poliqarp corpus search engine. Occurrences of queries involving particular grammatical categories were counted and used as weights. Obtained results are presented in Table 1.

Table 1: Occurrences of particular grammatical categories in query logs of the Poliqarp corpus search engine.

Category	# occurrences
POS	37771
CASE	14055
NUMBER	2074
GENDER	552
ASPECT	222
PERSON	186
DEGREE	81
ACCOMMODABILITY	25
POST-PREP.	8
NEGATION	7
ACCENTABILITY	5
AGGLUTINATION	4

3.1 Scored information retrieval metrics

In § 2 a number of methods of assigning a score to a pair of tags were presented. From now on, let name them *scoring functions*. One could use them directly for evaluation, given that both the tagger and the golden standard always assign a single interpretation to each segment. This is not the case for the corpus we use, hence we propose generalisation of standard information retrieval metrics (precision, recall and F-measure) as well as strong and weak correctness (Karwańska and Przepiórkowski, 2009) to account for scoring functions.

Denote by T_i and G_i the sets of tags assigned by the tagger and the golden standard, accordingly, to the i-th segment of the tagged text. The set of all tags in the tagset is denoted by $\mathbf{T}$. The scoring function used is $score\colon \mathbf{T} \times \mathbf{T} \to [0, 1]$. Also, to save up on notation, we define

$$score(t, A) := \max_{t' \in A} score(t, t')$$

Now, given the text has n segments, we take

$$P = \frac{\sum_{i=1}^{n} \sum_{t \in T_i} score(t, G_i)}{\sum_{i=1}^{n} |T_i|}$$

$$R = \frac{\sum_{i=1}^{n} \sum_{g \in G_i} score(g, T_i)}{\sum_{i=1}^{n} |G_i|}$$

$$F = \frac{2 \cdot P \cdot R}{P + R}$$

$$WC = \frac{\sum_{i=1}^{n} \max_{t \in T_i} score(t, G_i)}{n}$$

$$SC = \frac{\sum_{i=1}^{n} \min(\{score(t, G_i)\colon t \in T_i\} \cup \{score(g, T_i)\colon g \in G_i\})}{n}$$

Intuitions for scored precision and recall are that precision specifies the percent of tags assigned by the tagger which have a high score with some corresponding golden tag. Analogously recall estimates the percent of golden tags which have high scores with some corresponding tag assigned by the tagger. The definition of recall is slightly different than proposed by Ziółko et al. (Ziółko et al., 2007) so that recall is never greater than one.[3]

[3] For example if the golden standard specifies a single tag and the tagger determines two tags which all score 0.6 when compared with the golden, then if we used equations from Ziółko et al., we would get the recall of 1.2.

3.2 Evaluation results

Now the taggers were trained on the same data consisting of 90% segments of the corpus and then tested on the remaining 10%. Results were 10-fold cross-validated. They are presented in Tables 2, 3, 4 and 5.

As expected, the values obtained with PA and WPA fall between the numbers for standard metrics calculated with full tags and only the part of speech. What is worth observing is that the use of WPA makes values for scored precision and recall much closer together. This can be justified by the fact that the golden standard relatively frequently contains more than one interpretation for some tags, which differ only in values of less important grammatical categories. WPA is resilient to such situations.

One may argue that such scoring functions may hide a large number of tagging mistakes occurring in low-weighted categories. But this is not the case as the clearly most common tagging errors reported in both (Piasecki and Godlewski, 2006) and (Acedański and Gołuchowski, 2009) are for CASE, GENDER and NUMBER. Also, the motivation for weighting grammatical categories is to actually ignore errors in not important ones. To be fair, though, one should make sure that the weights used for evaluation match the actual application domain of the analysed tagger, and if no specific domain is known, using a number of measures is recommended.

It should also be noted that for classic information retrieval metrics, the result of weak correctness for TaKIPI is more similar to 92.55% reported by the authors (Piasecki and Godlewski, 2006) than 91.30% shown in (Karwańska and Przepiórkowski, 2009) despite using the same test corpus and very similar methodology[4] as the second paper presents.

4 Conclusion

This paper stems from the observation that the commonly used measure for tagger evaluation, i.e., accuracy, does not distinguish between completely incorrect and partially correct interpretations, even though the latter may be sufficient for some applications. We proposed a way of grading tag assignments, by weighting the importance of particular categories (case, number, etc.) and the part of speech. Three variants of the weighted positional accuracy were presented: one intrinsic and two application-oriented, and an extension of WPA to conditional WPA was discussed. The variant of WPA related to the needs of the users of a corpus search engine for the National Corpus of Polish was implemented and its usefulness was demonstrated. We plan to implement the parsing-oriented WPA in the future.

We conclude that tagger evaluation is far from being a closed chapter and the time has come to adopt more subtle approaches than sheer accuracy, approaches able to cope with morphological richness and oriented towards real applications.

[4]The only difference was not contracting the grammatical category of ACCOMMODABILITY present for masculine numerals in the golden standard.

References

Acedański, Szymon and Konrad Gołuchowski. 2009. A morphosyntactic rule-based Brill tagger for Polish. In Kłopotek, Mieczysław A., Adam Przepiórkowski, Sławomir T. Wierzchoń, and Krzysztof Trojanowski, editors, *Advances in Intelligent Information Systems — Design and Applications*, pages 67–76. Akademicka Oficyna Wydawnicza EXIT, Warsaw.

Dębowski, Łukasz. 2009. Valence extraction using the EM selection and co-occurrence matrices. *Language Resources and Evaluation*, 43:301–327.

Hajič, Jan and Barbora Hladká. 1997. Probabilistic and rule-based tagger of an inflective language - a comparison. In *Proceedings of the 5th Applied Natural Language Processing Conference*, pages 111–118, Washington, DC. ACL.

Hajič, Jan. 2004. *Disambiguation of Rich Inflection*. Karolinum Press, Prague.

Janus, Daniel and Adam Przepiórkowski. 2007. Poliqarp: An open source corpus indexer and search engine with syntactic extensions. In *Proceedings of the ACL 2007 Demo and Poster Sessions*, pages 85–88, Prague.

Karwańska, Danuta and Adam Przepiórkowski. 2009. On the evaluation of two Polish taggers. In Goźdź-Roszkowski, Stanisław, editor, *The proceedings of Practical Applications in Language and Computers PALC 2009*, Frankfurt am Main. Peter Lang. Forthcoming.

Oliva, Karel. 2001. On retaining ambiguity in disambiguated corpora: Programmatic reflections on why's and how's. *TAL (Traitement Automatique des Langues)*, 42(2):487–500.

Piasecki, Maciej and Grzegorz Godlewski. 2006. Effective Architecture of the Polish Tagger. In Sojka, Petr, Ivan Kopecek, and Karel Pala, editors, *TSD*, volume 4188 of *Lecture Notes in Computer Science*, pages 213–220. Springer.

Przepiórkowski, Adam and Marcin Woliński. 2003. The unbearable lightness of tagging: A case study in morphosyntactic tagging of Polish. In *Proceedings of the 4th International Workshop on Linguistically Interpreted Corpora (LINC-03), EACL 2003*, pages 109–116.

Przepiórkowski, Adam. 2005. The IPI PAN Corpus in Numbers. In *Proceedings of the* 2nd Language & Technology Conference, Poznań, Poland.

Przepiórkowski, Adam. 2008. *Powierzchniowe przetwarzanie języka polskiego*. Akademicka Oficyna Wydawnicza EXIT, Warsaw.

van Halteren, Hans. 1999. Performance of taggers. In van Halteren, Hans, editor, *Syntactic Wordclass Tagging*, volume 9 of *Text, Speech and Language Technology*, pages 81–94. Kluwer, Dordrecht.

Ziółko, Bartosz, Suresh Manandhar, and Richard Wilson. 2007. Fuzzy Recall and Precision for Speech Segmentation Evaluation. In *Proceedings of 3rd Language & Technology Conference, Poznan, Poland*, October.

Table 2: Evaluation results — standard information retrieval metrics, full tags

Tagger	C (%)	WC (%)	P (%)	R (%)	F (%)
TaKIPI — defaults	87.67%	92.10%	89.93%	84.72%	87.25%
TaKIPI — one tag per seg.	88.68%	91.06%	90.94%	83.78%	87.21%
Brill	90.01%	92.44%	92.26%	85.00%	88.49%

Table 3: Evaluation results — standard information retrieval metrics, PoS only

Tagger	C (%)	WC (%)	P (%)	R (%)	F (%)
TaKIPI — defaults	95.56%	97.52%	95.71%	97.61%	96.65%
TaKIPI — one tag per seg.	96.53%	96.54%	96.58%	96.71%	96.65%
Brill	98.17%	98.18%	98.20%	98.26%	98.23%

Table 4: Evaluation results — scored metrics, Positional Accuracy

Tagger	C (%)	WC (%)	P (%)	R (%)	F (%)
TaKIPI — defaults	95.23%	96.58%	95.69%	95.44%	95.57%
TaKIPI — one tag per seg.	95.69%	96.10%	96.12%	95.00%	95.56%
Brill	97.02%	97.43%	97.42%	96.27%	96.84%

Table 5: Evaluation results — scored metrics, Weighted PA, Poliqarp weights

Tagger	C (%)	WC (%)	P (%)	R (%)	F (%)
TaKIPI — defaults	95.20%	96.62%	95.34%	96.56%	95.95%
TaKIPI — one tag per seg.	95.88%	95.93%	95.97%	95.94%	95.95%
Brill	97.34%	97.40%	97.41%	97.34%	97.38%

Document Expansion Based on WordNet for Robust IR

Eneko Agirre
IXA NLP Group
Univ. of the Basque Country
e.agirre@ehu.es

Xabier Arregi
IXA NLP Group
Univ. of the Basque Country
xabier.arregi@ehu.es

Arantxa Otegi
IXA NLP Group
Univ. of the Basque Country
arantza.otegi@ehu.es

Abstract

The use of semantic information to improve IR is a long-standing goal. This paper presents a novel Document Expansion method based on a WordNet-based system to find related concepts and words. Expansion words are indexed separately, and when combined with the regular index, they improve the results in three datasets over a state-of-the-art IR engine. Considering that many IR systems are not robust in the sense that they need careful fine-tuning and optimization of their parameters, we explored some parameter settings. The results show that our method is specially effective for realistic, non-optimal settings, adding robustness to the IR engine. We also explored the effect of document length, and show that our method is specially successful with shorter documents.

1 Introduction

Since the earliest days of IR, researchers noted the potential pitfalls of keyword retrieval, such as synonymy, polysemy, hyponymy or anaphora. Although in principle these linguistic phenomena should be taken into account in order to obtain high retrieval relevance, the lack of algorithmic models prohibited any systematic study of the effect of this phenomena in retrieval. Instead, researchers resorted to distributional semantic models to try to improve retrieval relevance, and overcome the brittleness of keyword matches. Most research concentrated on Query Expansion (QE) methods, which typically analyze term co-occurrence statistics in the corpus and in the highest scored documents for the original query in order to select terms for expanding the query terms (Manning et al., 2009). Document expansion (DE) is a natural alternative to QE, but surprisingly it was not investigated until very recently. Several researchers have used distributional methods from similar documents in the collection in order to expand the documents with related terms that do not actually occur in the document (Liu and Croft, 2004; Kurland and Lee, 2004; Tao et al., 2006; Mei et al., 2008; Huang et al., 2009). The work presented here is complementary, in that we also explore DE, but use WordNet instead of distributional methods.

Lexical semantic resources such as WordNet (Fellbaum, 1998) might provide a principled and explicit remedy for the brittleness of keyword matches. WordNet has been used with success in psycholinguistic datasets of word similarity and relatedness, where it often surpasses distributional methods based on keyword matches (Agirre et al., 2009b). WordNet has been applied to IR before. Some authors extended the query with related terms (Voorhees, 1994; Liu et al., 2005), while others have explicitly represented and indexed word senses after performing word sense disambiguation (WSD) (Gonzalo et al., 1998; Stokoe et al., 2003; Kim et al., 2004). More recently, a CLEF task was organized (Agirre et al., 2008; Agirre et al., 2009a) where queries and documents were semantically disambiguated, and participants reported mixed results.

This paper proposes to use WordNet for document expansion, proposing a new method: given

Coling 2008: Poster Volume, pages 9–17,
Beijing, August 2010

a full document, a random walk algorithm over the WordNet graph ranks concepts closely related to the words in the document. This is in contrast to previous WordNet-based work which focused on WSD to replace or supplement words with their senses. Our method discovers important concepts, even if they are not explicitly mentioned in the document. For instance, given a document mentioning *virus*, *software* and *DSL*, our method suggests related concepts and associated words such us *digital subscriber line*, *phone company* and *computer*. Those expansion words are indexed separately, and when combined with the regular index, we show that they improve the results in three datasets over a state-of-the-art IR engine (Boldi and Vigna, 2005). The three datasets used in this study are ResPubliQA (Peñas et al., 2009), Yahoo! Answers (Surdeanu et al., 2008) and CLEF-Robust (Agirre et al., 2009a).

Considering that many IR systems are not robust in the sense that they need careful fine-tuning and optimization of their parameters, we decided to study the robustness of our method, exploring some alternative settings, including default parameters, parameters optimized in development data, and parameters optimized in other datasets. The study reveals that the additional semantic expansion terms provide robustness in most cases.

We also hypothesized that semantic document expansion could be most profitable when documents are shorter, and our algorithm would be most effective for collections of short documents. We artificially trimmed documents in the Robust dataset. The results, together with the analysis of document lengths of the three datasets, show that document expansion is specially effective for very short documents, but other factors could also play a role.

The paper is structured as follows. We first introduce the document expansion technique. Section 3 introduces the method to include the expansions in a retrieval system. Section 4 presents the experimental setup. Section 5 shows our main results. Sections 6 and 7 analyze the robustness and relation to document length. Section 8 compares to related work. Finally, the conclusions and future work are mentioned.

2 Document Expansion Using WordNet

Our key insight is to expand the document with related words according to the background information in WordNet (Fellbaum, 1998), which provides generic information about general vocabulary terms. WordNet groups nouns, verbs, adjectives and adverbs into sets of synonyms (synsets), each expressing a distinct concept. Synsets are interlinked with conceptual-semantic and lexical relations, including hypernymy, meronymy, causality, etc.

In contrast with previous work, we select those concepts that are most closely related to the document as a whole. For that, we use a technique based on random walks over the graph representation of WordNet concepts and relations.

We represent WordNet as a graph as follows: graph nodes represent WordNet concepts (synsets) and dictionary words; relations among synsets are represented by undirected edges; and dictionary words are linked to the synsets associated to them by directed edges. We used version 3.0, with all relations provided, including the gloss relations. This was the setting obtaining the best results in a word similarity dataset as reported by Agirre et al. (2009b).

Given a document and the graph-based representation of WordNet, we obtain a ranked list of WordNet concepts as follows:

1. We first pre-process the document to obtain the lemmas and parts of speech of the open category words.
2. We then assign a uniform probability distribution to the terms found in the document. The rest of nodes are initialized to zero.
3. We compute personalized PageRank (Haveliwala, 2002) over the graph, using the previous distribution as the reset distribution, and producing a probability distribution over WordNet concepts The higher the probability for a concept, the more related it is to the given document.

Basically, personalized PageRank is computed by modifying the random jump distribution vector in the traditional PageRank equation. In our case, we concentrate all probability mass in the concepts corresponding to the words in the docu-

ment.

Let G be a graph with N vertices $v_1, \ldots, v_N$ and d_i be the outdegree of node i; let M be a $N \times N$ transition probability matrix, where $M_{ji} = \frac{1}{d_i}$ if a link from i to j exists, and zero otherwise. Then, the calculation of the *PageRank vector* $\mathbf{Pr}$ over G is equivalent to resolving Equation (1).

$$\mathbf{Pr} = cM\mathbf{Pr} + (1-c)\mathbf{v} \quad (1)$$

In the equation, $\mathbf{v}$ is a $N \times 1$ vector and c is the so called *damping factor*, a scalar value between 0 and 1. The first term of the sum on the equation models the voting scheme described in the beginning of the section. The second term represents, loosely speaking, the probability of a surfer randomly jumping to any node, e.g. without following any paths on the graph. The damping factor, usually set in the $[0.85..0.95]$ range, models the way in which these two terms are combined at each step.

The second term on Eq. (1) can also be seen as a smoothing factor that makes any graph fulfill the property of being aperiodic and irreducible, and thus guarantees that PageRank calculation converges to a unique stationary distribution.

In the traditional PageRank formulation the vector $\mathbf{v}$ is a stochastic normalized vector whose element values are all $\frac{1}{N}$, thus assigning equal probabilities to all nodes in the graph in case of random jumps. In the case of personalized PageRank as used here, $\mathbf{v}$ is initialized with uniform probabilities for the terms in the document, and 0 for the rest of terms.

PageRank is actually calculated by applying an iterative algorithm which computes Eq. (1) successively until a fixed number of iterations are executed. In our case, we used a publicly available implementation[1], with default values for the damping value (0.85) and the number of iterations (30). In order to select the expansion terms, we chose the 100 highest scoring concepts, and get all the words that lexicalize the given concept.

Figure 1 exemplifies the expansion. Given the short document from Yahoo! Answers (cf. Section 4) shown in the top, our algorithm produces the set of related concepts and words shown in the bottom. Note that the expansion produces synonyms, but also other words related to concepts that are not mentioned in the document.

[1] http://ixa2.si.ehu.es/ukb/

3 Including Expansions in a Retrieval System

Once we have the list of words for document expansion, we create one index for the words in the original documents and another index with the expansion terms. This way, we are able to use the original words only, or to also include the expansion words during the retrieval.

The retrieval system was implemented using MG4J (Boldi and Vigna, 2005), as it provides state-of-the-art results and allows to combine several indices over the same document collection. We conducted different runs, by using only the index made of original words (baseline) and also by using the index with the expansion terms of the related concepts.

BM25 was the scoring function of choice. It is one of the most relevant and robust scoring functions available (Robertson and Zaragoza, 2009).

$$w_{Dt}^{BM25} := \frac{tf_{D_t}}{k_1\left((1-b)+b\frac{dl_D}{avdl_D}\right)+tf_{Dt}} idf_t \quad (2)$$

where tf_{D_t} is the term frequency of term t in document D, dl_D is the document length, idf_t is the inverted document frequency (or more specifically the RSJ weight, (Robertson and Zaragoza, 2009)), and k_1 and b are free parameters.

The two indices were combined linearly, as follows (Robertson and Zaragoza, 2009):

$$score(d,e,q) := \sum_{t \in q \cap d} w_{D_t}^{BM25} + \lambda \sum_{t \in q \cap e} w_{E_t}^{BM25} \quad (3)$$

where *D* and *E* are the original and expanded indices, *d, e* and *q* are the original document, the expansion of the document and the query respectively, t is a term, and λ is a free parameter for the relative weight of the expanded index.

You should only need to turn off virus and anti-spy not uninstall. And that's done within each of the softwares themselves. Then turn them back on later after installing any DSL softwares.

06566077-n → *computer software, package*, **software**, *software package, software program, software system*

03196990-n → *digital subscriber line*, **dsl**

01569566-v → *instal*, **install**, *put in, set up*

04402057-n → line, phone line, suscriber line, telephone circuit, telephone line

08186221-n → phone company, phone service, telco, telephone company, telephone service

03082979-n → computer, computing device, computing machine, data processor, electronic computer

Figure 1: Example of a document expansion, with original document on top, and some of the relevant WordNet concepts identified by our algorithm, together with the words that lexicalize them. Words in the original document are shown in bold, synonyms in italics, and other related words underlined.

4 Experimental Setup

We chose three data collections. The first is based on a traditional news collection. DE could be specially interesting for datasets with short documents, which lead our choice of the other datasets: the second was chosen because it contains shorter documents, and the third is a passage retrieval task which works on even shorter paragraphs. Table 1 shows some statistics about the datasets.

One of the collections is the English dataset of the **Robust** task at CLEF 2009 (Agirre et al., 2009a). The documents are news collections from LA Times 94 and Glasgow Herald 95. The topics are statements representing information needs, consisting of three parts: a brief title statement; a one-sentence description; a more complex narrative describing the relevance assessment criteria. We use only the title and the description parts of the topics in our experiments.

The **Yahoo! Answers** corpus is a subset of a dump of the Yahoo! Answers web site[2] (Surdeanu et al., 2008), where people post questions and answers, all of which are public to any web user willing to browse them. The dataset is a small subset of the questions, selected for their linguistic properties (for example they all start with "how {to||do||did||does||can||would||could||should}"). Additionally, questions and answers of obvious low quality were removed. The document set was created with the best answer of each question (only one for each question).

	docs	length	q. train	q. test
Robust	166,754	532	150	160
Yahoo!	89610	104	1000	88610
ResPubliQA	1,379,011	20	100	500

Table 1: Number of documents, average document length, number of queries for train and test in each collection.

The other collection is the English dataset of **ResPubliQA** exercise at the Multilingual Question Answering Track at CLEF 2009 (Peñas et al., 2009). The exercise is aimed at retrieving paragraphs that contain answers to a set of 500 natural language questions. The document collection is a subset of the JRC-Acquis Multilingual Parallel Corpus, and consists of 21,426 documents for English which are aligned to a similar number of documents in other languages[3]. For evaluation, we used the gold standard released by the organizers, which contains a single correct passage for each query. As the retrieval unit is the passage, we split the document collection into paragraphs. We applied the expansion strategy only to passages which had more than 10 words (half of the passages), for two reasons: the first one was that most of these passages were found not to contain relevant information for the task (e.g. "Article 2" or "Having regard to the proposal from the Commission"), and the second was that we thus saved some computation time.

In order to evaluate the quality of our expansion in practical retrieval settings, the next Section re-

[2]Yahoo! Webscope dataset "ydata-yanswers-manner-questions-v1_0" http://webscope.sandbox.yahoo.com/

[3]Note that Table 1 shows the number of paragraphs, which conform the units we indexed.

		base.	expa.	Δ
Robust	MAP	.3781	**.3835****	1.43%
Yahoo!	MRR	.2900	**.2950****	1.72%
	P@1	.2142	**.2183****	1.91%
ResPubliQA	MRR	.3931	**.4077****	3.72%
	P@1	.2860	**.3000****	4.90%

Table 2: Results using default parameters.

		base.	expa.	Δ
Robust	MAP	.3740	**.3823****	2.20%
Yahoo!	MRR	.3070	**.3100****	0.98%
	P@1	.2293	**.2317***	1.05%
ResPubliQA	MRR	.4970	.4942	-0.56%
	P@1	.3980	.3940	-1.01%

Table 3: Results using optimized parameters.

port results with respect to several parameter settings. Parameter optimization is often neglected in retrieval with linguistic features, but we think it is crucial since it can have a large effect on relevance performance and therefore invalidate claims of improvements over the baseline. In each setting we assign different values to the free parameters in the previous section, k_1, b and λ.

5 Results

The main evaluation measure for Robust is mean Average Precision (MAP), as customary. In two of the datasets (Yahoo! and ResPubliQA) there is a single correct answer per query, and therefore we use Mean Reciprocal Rank (MRR) and Mean Precision at rank 1 (P@1) for evaluation. Note that in this setting MAP is identical to MRR. Statistical significance was computed using Paired Randomization Test (Smucker et al., 2007). In the tables throughout the paper, we use * to indicate statistical significance at 90% confidence level, ** for 95% and *** for 99%. Unless noted otherwise, base. refers to MG4J with the standard index, and expa. refers to MG4J using both indices. Best results per row are in bold when significant. Δ reports relative improvement respect to the baseline.

5.1 Default Parameter Setting

The values for k_1 and b are the default values as provided in the w^{BM25} implementation of MG4J, 1.2 and 0.5 respectively. We could not think of a straightforward value for λ. A value of 1 would mean that we are assigning equal importance to original and expanded terms, which seemed an overestimation, so we used 0.1. Table 2 shows the results when using the default setting of parameters. The use of expansion is beneficial in all datasets, with relative improvements ranging from 1.43% to 4.90%.

Setting	System	k_1	b	λ
Default	base.	1.20	0.50	-
	expa.	1.20	0.50	0.100
Robust	base.	1.80	0.64	-
	expa.	1.66	0.55	0.075
Yahoo!	basel.	0.99	0.82	-
	expa.	0.84	0.87	0.146
ResPubliQA	base.	0.09	0.56	-
	expa.	0.13	0.65	0.090

Table 4: Parameters as in the default setting or as optimized in each dataset. The λ parameter is not used in the baseline systems.

5.2 Optimized Parameter Setting

We next optimized all three parameters using the train part of each collection. The optimization of the parameters followed a greedy method called "promising directions" (Robertson and Zaragoza, 2009). The comparison between the baseline and expansion systems in Table 3 shows that expansion helps in Yahoo! and Robust, with statistical significance. The differences in ResPubliQA are not significant, and indicate that expansion terms were not helpful in this setting.

Note that the optimization of the parameters yields interesting effects in the baseline for each of the datasets. If we compare the results of the baseline with default settings (Table 2) and with optimized setting (Table 3), the baseline improves MRR dramatically in ResPubliQA (26% relative improvement), significantly in Yahoo! (5.8%) and decreases MAP in Robust (-0.01%). This disparity of effects could be explained by the fact that the default values are often approximated using TREC-style news collections, which is exactly the genre of the Robust documents, while Yahoo uses shorter documents, and ResPubliQA has the shortest documents.

Table 4 summarizes the values of the parameters in both default and optimized settings. For k_1, the optimization yields very different values. In Robust the value is similar to the default value, but

		base.	expa.	Δ	λ
Rob	MAP	.3781	**.3881****	2.64%	0.18
Y!	MRR	.2900	**.2980****	2.76%	0.27
	P@1	.2142	**.2212****	3.27%	
ResP.	MRR	.3931	**.4221****	7.39%	0.61
	P@1	.2860	**.3180****	11.19%	

Table 5: Results obtained using the λ optimized setting, including actual values of λ.

in ResPubliQA the optimization pushes it down below the typical values cited in the literature (Robertson and Zaragoza, 2009), which might explain the boost in performance for the baseline in the case of ResPubliQA. When all three parameters are optimized together, the values λ in the table range from 0.075 to 0.146. The values of the optimized λ can be seem as an indication of the usefulness of the expanded terms, so we explored this farther.

5.3 Exploring λ

As an additional analysis experiment, we wanted to know the effect of varying λ keeping k_1 and b constant at their default values. Table 5 shows the best values in each dataset, which that the weight of the expanded terms and the relative improvement are highly correlated.

5.4 Exploring Number of Expansion Concepts

One of the free parameters of our system is the number of concepts to be included in the document expansion. We have performed a limited study with the default parameter setting on the Robust setting, using 100, 500 and 750 concepts, but the variations were not statistically significant. Note that with 100 concepts we were actually expanding with 268 words, with 500 concepts we add 1247 words and with 750 concepts we add 1831 words.

6 Robustness

The results in the previous section indicate that optimization is very important, but unfortunately real applications usually lack training data. In this Section we wanted to study whether the parameters can be carried over from one dataset to the other, and if not, whether the extra terms found by DE would make the system more robust to those sub-optimal parameters.

	train		base.	expa.	Δ
Rob.	def.	MAP	.3781	**.3835****	1.43%
	Rob.	MAP	.3740	**.3823****	2.20%
	Y!	MAP	.3786	.3759	-0.72%
	Res.	MAP	.3146	**.3346****	6.35%
Y!	def.	MRR	.2900	**.2950****	1.72%
	Rob.	MRR	.2920	.2920	0.0%
	Y!	MRR	.3070	**.3100****	0.98%
	Res.	MRR	.2600	**.2750****	5.77%
ResP.	def.	MRR	.3931	**.4077****	3.72%
	Rob.	MRR	.3066	**.3655****	19.22%
	Y!	MRR	.3010	**.3459****	14.93%
	Res.	MRR	.4970	.4942	-0.56%

Table 6: Results optimizing parameters with training from other datasets. We also include default and optimization on the same dataset for comparison. Only MRR and MAP results are given.

Table 6 includes a range of parameter settings, including defaults, and optimized parameters coming from the same and different datasets. The values of the parameters are those in Table 4. The results show that when the parameters are optimized in other datasets, DE provides improvement with statistical significance in all cases, except for the Robust dataset when using parameters optimized from Yahoo! and vice-versa.

Overall, the table shows that our DE method either improves the results significantly or does not affect performance, and that it provides robustness across different parameter settings, even with sub-optimal values.

7 Exploring Document Length

The results in Table 6 show that the performance improvements are best in the collection with shortest documents (ResPubliQA). But the results for Robust and Yahoo! do not show any relation to document length. We thus decided to do an additional experiment artificially shrinking the document in Robust to a certain percentage of its original length. We create new pseudo-collection with the shrinkage factors of 2.5%, 10%, 20% and 50%, keeping the first N% words in the document and discarding the rest. In all cases we used the same parameters, as optimized for Robust.

Table 7 shows the results (MAP), with some clear indication that the best improvements are ob-

tained for the shortest documents.

	length	base.	expa.	Δ
2.5%	13	.0794	.0851	7.18%
10%	53	.1757	.1833	4.33%
20%	107	.2292	.2329	1.61%
50%	266	.3063	.3098	1.14%
100%	531	.3740	.3823	2.22%

Table 7: Results (MAP) on Robust when artificially shrinking documents to a percentage of their length. In addition to the shrinking rate we show the average lengths of documents.

8 Related Work

Given the brittleness of keyword matches, most research has concentrated on Query Expansion (QE) methods. These methods analyze the user query terms and select automatically new related query terms. Most QE methods use statistical (or distributional) techniques to select terms for expansion. They do this by analyzing term co-occurrence statistics in the corpus and in the highest scored documents of the original query (Manning et al., 2009). These methods seemed to improve slightly retrieval relevance on average, but at the cost of greatly decreasing the relevance of difficult queries. But more recent studies seem to overcome some of these problems (Collins-Thompson, 2009).

An alternative to QE is to perform the expansion in the document. Document Expansion (DE) was first proposed in the speech retrieval community (Singhal and Pereira, 1999), where the task is to retrieve speech transcriptions which are quite noisy. Singhal and Pereira propose to enhance the representation of a noisy document by adding to the document vector a linearly weighted mixture of related documents. In order to determine related documents, the original document is used as a query into the collection, and the ten most relevant documents are selected.

Two related papers (Liu and Croft, 2004; Kurland and Lee, 2004) followed a similar approach on the TREC ad-hoc document retrieval task. They use document clustering to determine similar documents, and document expansion is carried out with respect to these. Both papers report significant improvements over non-expanded baselines. Instead of clustering, more recent work (Tao et al., 2006; Mei et al., 2008; Huang et al., 2009) use language models and graph representations of the similarity between documents in the collection to smooth language models with some success. The work presented here is complementary, in that we also explore DE, but use WordNet instead of distributional methods. They use a tighter integration of their expansion model (compared to our simple two-index model), which coupled with our expansion method could help improve results further. We plan to explore this in the future.

An alternative to statistical expansion methods is to use lexical semantic knowledge bases such as WordNet. Most of the work has focused on query expansion and the use of synonyms from WordNet after performing word sense disambiguation (WSD) with some success (Voorhees, 1994; Liu et al., 2005). The short context available in the query when performing WSD is an important problems of these techniques. In contrast, we use full document context, and related words beyond synonyms. Another strand of WordNet based work has explicitly represented and indexed word senses after performing WSD (Gonzalo et al., 1998; Stokoe et al., 2003; Kim et al., 2004). The word senses conform a different space for document representation, but contrary to us, these works incorporate concepts for all words in the documents, and are not able to incorporate concepts that are not explicitly mentioned in the document. More recently, a CLEF task was organized (Agirre et al., 2009a) where terms were semantically disambiguated to see the improvement that this would have on retrieval; the conclusions were mixed, with some participants slightly improving results with information from WordNet. To the best of our knowledge our paper is the first on the topic of document expansion using lexical-semantic resources.

We would like to also compare our performance to those of other systems as tested on the same datasets. The systems which performed best in the Robust evaluation campaign (Agirre et al., 2009a) report 0.4509 MAP, but note that they deployed a complex system combining probabilistic and monolingual translation-based models. In ResPubliQA (Peñas et al., 2009), the official eval-

uation included manual assessment, and we cannot therefore reproduce those results. Fortunately, the organizers released all runs, but only the first ranked document for each query was included, so we could only compute P@1. The P@1 of best run was 0.40. Finally (Surdeanu et al., 2008) report MRR figure around 0.68, but they evaluate only in the questions where the correct answer is retrieved by answer retrieval in the top 50 answers, and is thus not comparable to our setting.

Regarding the WordNet expansion technique we use here, it is implemented on top of publicly available software[4], which has been successfully used in word similarity (Agirre et al., 2009b) and word sense disambiguation (Agirre and Soroa, 2009). In the first work, a single word was input to the random walk algorithm, obtaining the probability distribution over all WordNet synsets. The similarity of two words was computed as the similarity of the distribution of each word, obtaining the best results for WordNet-based systems on the word similarity dataset, and comparable to the results of a distributional similarity method which used a crawl of the entire web. Agirre et al. (2009) used the context of occurrence of a target word to start the random walk, and obtained very good results for WordNet WSD methods.

9 Conclusions and Future Work

This paper presents a novel Document Expansion method based on a WordNet-based system to find related concepts and words. The documents in three datasets were thus expanded with related words, which were fed into a separate index. When combined with the regular index we report improvements over MG4J using w^{BM25} for those three datasets across several parameter settings, including default values, optimized parameters and parameters optimized in other datasets. In most of the cases the improvements are statistically significant, indicating that the information in the document expansion is useful. Similar to other expansion methods, parameter optimization has a stronger effect than our expansion strategy. The problem with parameter optimization is that in most real cases there is no tuning dataset available. Our analysis shows that our expansion method is more effective for sub-optimal parameter settings, which is the case for most real-live IR applications. A comparison across the three datasets and using artificially trimmed documents indicates that our method is particularly effective for short documents.

As document expansion is done at indexing time, it avoids any overhead at query time. It also has the advantage of leveraging full document context, in contrast to query expansion methods, which use the scarce information present in the much shorter queries. Compared to WSD-based methods, our method has the advantage of not having to disambiguate all words in the document. Besides, our algorithm picks the most relevant concepts, and thus is able to expand to concepts which are not explicitly mentioned in the document. The successful use of background information such as the one in WordNet could help close the gap between semantic web technologies and IR, and opens the possibility to include other resources like Wikipedia or domain ontologies like those in the Unified Medical Language System.

Our method to integrate expanded terms using an additional index is simple and straightforward, and there is still ample room for improvement. A tighter integration of the document expansion technique in the retrieval model should yield better results, and the smoothed language models of (Mei et al., 2008; Huang et al., 2009) seem a natural choice. We would also like to compare with other existing query and document expansion techniques and study whether our technique is complementary to query expansion approaches.

[4] http://ixa2.si.ehu.es/ukb

Acknowledgments

This work has been supported by KNOW2 (TIN2009-14715-C04-01) and KYOTO (ICT-2007-211423) projects. Arantxa Otegi's work is funded by a PhD grant from the Basque Government. Part of this work was done while Arantxa Otegi was visiting Yahoo! Research Barcelona.

References

Agirre, E. and A. Soroa. 2009. Personalizing PageRank for Word Sense Disambiguation. In *Proc. of*

EACL 2009, Athens, Greece.

Agirre, E., G. M. Di Nunzio, N. Ferro, T. Mandl, and C. Peters. 2008. CLEF 2008: Ad-Hoc Track Overview. In *Working Notes of the Cross-Lingual Evaluation Forum*.

Agirre, E., G. M. Di Nunzio, T. Mandl, and A. Otegi. 2009a. CLEF 2009 Ad Hoc Track Overview: Robust - WSD Task. In *Working Notes of the Cross-Lingual Evaluation Forum*.

Agirre, E., A. Soroa, E. Alfonseca, K. Hall, J. Kravalova, and M. Pasca. 2009b. A Study on Similarity and Relatedness Using Distributional and WordNet-based Approaches. In *Proc. of NAACL*, Boulder, USA.

Boldi, P. and S. Vigna. 2005. MG4J at TREC 2005. In *The Fourteenth Text REtrieval Conference (TREC 2005) Proceedings*, number SP 500-266 in Special Publications. NIST.

Collins-Thompson, Kevyn. 2009. Reducing the risk of query expansion via robust constrained optimization. In *Proceedings of CIKM '09*, pages 837–846.

Fellbaum, C., editor. 1998. *WordNet: An Electronic Lexical Database and Some of its Applications*. MIT Press, Cambridge, Mass.

Gonzalo, J., F. Verdejo, I. Chugur, and J. Cigarran. 1998. Indexing with WordNet synsets can improve text retrieval. In *Proceedings ACL/COLING Workshop on Usage of WordNet for Natural Language Processing*.

Haveliwala, T. H. 2002. Topic-sensitive PageRank. In *Proceedings of WWW '02*, pages 517–526.

Huang, Yunping, Le Sun, and Jian-Yun Nie. 2009. Smoothing document language model with local word graph. In *Proceedings of CIKM '09*, pages 1943–1946.

Kim, S. B., H. C. Seo, and H. C. Rim. 2004. Information retrieval using word senses: root sense tagging approach. In *Proceedings of SIGIR '04*, pages 258–265.

Kurland, O. and L. Lee. 2004. Corpus structure, language models, and ad hoc information retrieval. In *Proceedings of SIGIR '04*, pages 194–201.

Liu, X. and W. B. Croft. 2004. Cluster-based retrieval using language models. In *Proceedings of SIGIR '04*, pages 186–193.

Liu, S., C. Yu, and W. Meng. 2005. Word sense disambiguation in queries. In *Proceedings of CIKM '05*, pages 525–532.

Manning, C. D., P. Raghavan, and H. Schütze. 2009. *An introduction to information retrieval*. Cambridge University Press, UK.

Mei, Qiaozhu, Duo Zhang, and ChengXiang Zhai. 2008. A general optimization framework for smoothing language models on graph structures. In *Proceedings of SIGIR '08*, pages 611–618.

Peñas, A., P. Forner, R. Sutcliffe, A. Rodrigo, C. Forăscu, I. Alegria, D. Giampiccolo, N. Moreau, and P. Osenova. 2009. Overview of ResPubliQA 2009: Question Answering Evaluation over European Legislation. In *Working Notes of the Cross-Lingual Evaluation Forum*.

Robertson, S. and H. Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389.

Singhal, A. and F. Pereira. 1999. Document expansion for speech retrieval. In *Proceedings of SIGIR '99*, pages 34–41, New York, NY, USA. ACM.

Smucker, M. D., J. Allan, and B. Carterette. 2007. A comparison of statistical significance tests for information retrieval evaluation. In *Proc. of CIKM 2007*, Lisboa, Portugal.

Stokoe, C., M. P. Oakes, and J. Tait. 2003. Word sense disambiguation in information retrieval revisited. In *Proceedings of SIGIR '03*, page 166.

Surdeanu, M., M. Ciaramita, and H. Zaragoza. 2008. Learning to Rank Answers on Large Online QA Collections. In *Proceedings of ACL 2008*.

Tao, T., X. Wang, Q. Mei, and C. Zhai. 2006. Language model information retrieval with document expansion. In *Proceedings of HLT/NAACL*, pages 407–414, June.

Voorhees, E. M. 1994. Query expansion using lexical-semantic relations. In *Proceedings of SIGIR '94*, page 69.

Cross-Market Model Adaptation with Pairwise Preference Data for Web Search Ranking

Jing Bai
Microsoft Bing
1065 La Avenida
Mountain View, CA 94043
jbai@microsoft.com

Fernando Diaz, Yi Chang, Zhaohui Zheng
Yahoo! Labs
701 First Avenue
Sunnyvale, CA 94089
{diazf,yichang,zhaohui}@yahoo-inc.com

Keke Chen
Computer Science
Wright State
Dayton, Ohio 45435
keke.chen@wright.edu

Abstract

Machine-learned ranking techniques automatically learn a complex document ranking function given training data. These techniques have demonstrated the effectiveness and flexibility required of a commercial web search. However, manually labeled training data (with multiple absolute grades) has become the bottleneck for training a quality ranking function, particularly for a new domain. In this paper, we explore the adaptation of machine-learned ranking models across a set of geographically diverse markets with the market-specific pairwise preference data, which can be easily obtained from clickthrough logs. We propose a novel adaptation algorithm, PairwiseTrada, which is able to adapt ranking models that are trained with multi-grade labeled training data to the target market using the target-market-specific pairwise preference data. We present results demonstrating the efficacy of our technique on a set of commercial search engine data.

1 Introduction

Web search algorithms provide methods for ranking web scale collection of documents given a short query. The success of these algorithms often relies on the rich set of document properties or *features* and the complex relationships between them. Increasingly, machine learning techniques are being used to learn these relationships for an effective ranking function (Liu, 2009). These techniques use a set of labeled *training data* labeled with multiple relevance grades to automatically estimate parameters of a model which directly optimizes a performance metric. Although training data often is derived from editorial labels of document relevance, it can also be inferred from a careful analysis of users' interactions with a working system (Joachims, 2002). For example, in web search, given a query, document preference information can be derived from user clicks. This data can then be used with an algorithm which learns from pairwise preference data (Joachims, 2002; Zheng et al., 2007). However, automatically extracted pairwise preference data is subject to noise due to the specific sampling methods used (Joachims et al., 2005; Radlinski and Joachim, 2006; Radlinski and Joachim, 2007).

One of the fundamental problems for a web search engine with global reach is the development of ranking models for different regional markets. While the approach of training a single model for all markets is attractive, it fails to fully exploit of specific properties of the markets. On the other hand, the approach of training market-specific models requires the huge overhead of acquiring a large training set for each market. As a result, techniques have been developed to create a model for a small market, say a Southeast Asian country, by combining a strong model in another market, say the United States, with a

Coling 2008: Poster Volume, pages 18–26,
Beijing, August 2010

small amount of manually labeled training data in the small market (Chen et al., 2008b). However, the existing Trada method takes only multi-grade labeled training data for adaptation, making it impossible to take advantage of the easily harvested pairwise preference data. In fact, to our knowledge, there is no adaptation algorithm that is specifically developed for pairwise data.

In this paper, we address the development market-specific ranking models by leveraging pairwise preference data. The pairwise preference data contains most market-specific training examples, while a model from a large market may capture the common characteristics of a ranking function. By combining them algorithmically, our approach has two unique advantages. (1) The biases and noises of the pairwise preference data can be depressed by using the base model from the large market. (2) The base model can be tailored to the characteristics of the new market by incorporating the market specific pairwise training data. As the pairwise data has the particular form, the challenge is how to effectively use pairwise data in adaptation. This appeals to the following objective of many web search engines: design algorithms which minimize manually labeled data requirements while maintaining strong performance.

2 Related Work

In recent years, the ranking problem is frequently formulated as a supervised machine learning problem, which combines different kinds of features to train a ranking function. The ranking problem can be formulated as learning a function with pair-wise preference data, which is to minimize the number of contradicting pairs in training data. For example, RankSVM (Joachims, 2002) uses support vector machines to learn a ranking function from preference data; RankNet (Burges et al., 2005a) applies neural network and gradient descent to obtain a ranking function; RankBoost (Freund et al., 1998) applies the idea of boosting to construct an efficient ranking function from a set of weak ranking functions; GBRank (Zheng et al., 2007; Xia et al., 2008) using gradient descent in function spaces, which is able to learn relative ranking information in the context of web search. In addition, Several studies have been focused on learning ranking functions in semi-supervised learning framework (Amini et al., 2008; Duh and Kirchhoff, 2008), where unlabeled data are exploited to enhance ranking function. Another approach to learning a ranking function addresses the problem of optimizing the list-wise performance measures of information retrieval, such as mean average precision or Discount Cumulative Gain (Cao et al., 2007; Xu et al., 2008; Wu et al., 2009; Chen et al., 2008c). The idea of these methods is to obtain a ranking function that is optimal with respect to some information retrieval performance measure.

Model adaptation has previously been applied in the area of natural language processing and speech recognition. This approach has been successfully applied to parsing (Hwa, 1999), tagging (Blitzer et al., 2006), and language modeling for speech recognition (Bacchiani and Roark, 2003). Until very recently, several works have been presented on the topic of model adaptation for ranking (Gao et al., 2009; Chen et al., 2008b; Chen et al., 2009), however, none of them target the model adaptation with the pair-wise learning framework. Finally, multitask learning for ranking has also been proposed as a means of addressing problems similar to those we have encountered in model adaptation (Chen et al., 2008a; Bai et al., 2009; Geng et al., 2009).

3 Background

3.1 Gradient Boosted Decision Trees for Ranking

Assume we have a training data set, $\mathcal{D} = \{\langle (q,d), y\rangle_1, \ldots, \langle (q,d), y\rangle_n\}$, where $\langle (q,d), t\rangle_i$ encodes the labeled relevance, y, of a document, d, given query, q. Each query-document pair, (q,d), is represented by a set of features, $(q,d) = \{x_{i1}, x_{i2}, x_{i3}, ..., x_{im}\}$. These features include, for example, query-document match features, query-specific features, and document-specific features. Each relevance judgment, y, is a relevance grade mapped (e.g. "relevant", "somewhat relevant", "non-relevant") to a real

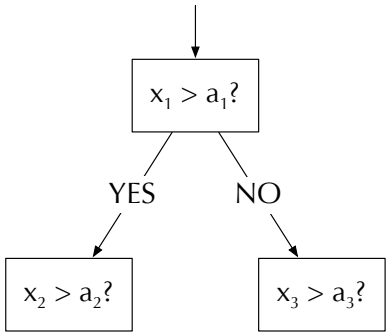


Figure 1: An example of base tree, where x_1, x_2 and x_3 are features and a_1, a_2 and a_3 are their splitting values.

number. Given this representation, we can learn a *gradient boosted decision tree* (GBDT) which models the relationship between document features, (q, d), and the relevance score, y, as a decision tree (Friedman, 2001). Figure 1 shows a portion of such a tree. Given a new query document pair, the GBDT can be used to predict the relevance grade of the document. A ranking is then inferred from these predictions. We refer to this method as GBDT_{reg}.

In the training phase, GBDT_{reg} iteratively constructs regression trees. The initial regression tree minimizes the L_2 loss with respect to the targets, y,

$$L_2(f, y) = \sum_{\langle (q,d), y \rangle} (f(q, d) - y)^2 \qquad (1)$$

As with other boosting algorithms, the subsequent trees minimize the L_2 loss with respect to the residuals of the predicted values and the targets. The final prediction, then, is the sum of the predictions of the trees estimated at each step,

$$f(x) = f^1(x) + \ldots + f^k(x) \qquad (2)$$

where $f^i(x)$ is the prediction of the ith tree.

3.2 Pairwise Training

As alternative to the absolute grades in $\mathcal{D}$, we can also imagine assembling a data set of *relative judgments*. In this case, assume we have a training data set $\mathcal{D}^{\succ} = \{\langle (q, d), (q, d'), \rho \rangle_1, \ldots, \langle (q, d), (q, d'), \rho \rangle_n\}$, where $\langle (q, d), (q, d'), \rho \rangle_i$ encodes the preference, of a document, d, to a second document, d', given query, q. Again, each query-document pair is represented by a set of features. Each preference judgment, $\rho \in \{\succ, \prec\}$, indicates whether document d is preferred to document d' ($d \succ d'$) or not ($d \prec d'$).

Preference data is attractive for several reasons. First, editors can often more easily determine preference between documents than the absolute grade of single documents. Second, relevance grades can often vary between editors. Some editors may tend to overestimate relevance compared to another editor. As a result, judgments need to be rescaled for editor biases. Although preference data is not immune to inter-editor inconsistency, absolute judgments introduce two potential sources of noise: determining a relevance ordering and determining a relevance grade. Third, even if grades can be accurately labeled, mapping those grades to real values is often done in a heuristic or *ad hoc* manner. Fourth, GBDT_{reg} potentially wastes modeling effort on predicting the grade of a document as opposed to focusing on optimizing the rank order of documents, the real goal a search engine. Finally, preference data can often be mined from a production system using assumptions about user clicks.

In order to support preference-based training data, (Zheng et al., 2007) proposed GBRANK based on GBDT_{reg}. The GBRANK training algorithm begins by constructing an initial tree which predicts a constant score, c, for all instances. A pair is contradicting if the $\langle (q, d), (q, d'), \succ \rangle$ and prediction $f(q, d) < f(q, d')$. At each boosting stage, the algorithm constructs a set of contradicting pairs, $\mathcal{D}^{\succ}_{\text{contra}}$. The GBRANK algorithm then adjusts the response variables, $f(q, d)$ and $f(q, d')$, so that $f(q, d) > f(q, d')$. Assume that $(q, d) \succ (q, d')$ and $f(q, d) < f(q, d')$. To correct the order, we modify the target values,

$$\tilde{f}(q, d) = f(q, d) + \tau \qquad (3)$$

$$\tilde{f}(q, d') = f(q, d') - \tau \qquad (4)$$

where $\tau > 0$ is a margin parameter that we

need to assign. In our experiments, we set τ to 1. Note that if preferences are inferred from absolute grades, $\mathcal{D}$, minimizing the L_2 to 0 also minimizes the contradictions.

3.3 Tree Adaptation

Recall that we are also interested in using the information learned from one market, which we will call the *source market*, on a second market, which we will call the *target market*. To this end, the Trada algorithm adapts a GBDT_{reg} model from the source market for the target market by using a small amount of target market absolute relevance judgments (Chen et al., 2008b). Let the $\mathcal{D}_s$ be the data in the source domain and $\mathcal{D}_t$ be the data in target domain. Assume we have trained a model using GBDT_{reg}. Our approach will be to use the decision tree structure learned from $\mathcal{D}_s$ but to adapt the thresholds in each node's feature. We will use Figure 1 to illustrate Trada. The splitting thresholds are a_1, a_2 and a_3 for rank features x_1, x_2 and x_3. Assume that the data set $\mathcal{D}_t$ is being evaluated at the root node v in Figure 1. We will split the using the feature $v_x = x_1$ but will compute a new threshold v'_a using $\mathcal{D}_t$ and the GBDT_{reg} algorithm. Because we are discussing the root node, when we select a threshold b, $\mathcal{D}_t$ will be partitioned into two sets, $\mathcal{D}_t^{>b}$ and $\mathcal{D}_t^{<b}$ representing those instances whose feature x_1 has a value greater and lower than b. The response value for each partition will be the uniform average of instances in that partition,

$$f = \begin{cases} \frac{1}{|\mathcal{D}_t^{>b}|} \sum_{d_i \in \mathcal{D}_t^{>b}} y_i & \text{if } d_i \in \mathcal{D}_t^{>b} \\ \frac{1}{|\mathcal{D}_t^{<b}|} \sum_{d_i \in \mathcal{D}_t^{<b}} y_i & \text{if } d_i \in \mathcal{D}_t^{<b} \end{cases} \quad (5)$$

We would like to select a value for b which minimizes the L_2 loss between y and f in Equation 5; equivalently, b can be selected to minimize the variance of y in each partition. In our implementation, we compute the L_2 loss for all possible values of the feature v'_x and select the value which minimizes the loss.

Once b is determined, the adaptation consists of performing a linear interpolation between the original splitting threshold v_a and the new splitting threshold b as follows:

$$v'_a = pv_a + (1-p)b \quad (6)$$

where p is an adaptation parameter which determines the scale of how we want to adapt the tree to the new task. If there is no additional information, we can select p according to the size of the data set,

$$p = \frac{|\mathcal{D}_s^{<a}|}{|\mathcal{D}_s^{<a}| + |\mathcal{D}_t^{<b}|} \quad (7)$$

In practice, we often want to enhance the adaptation scale since the training data of the extended task is small. Therefore, we add a parameter β to boost the extended task as follows:

$$p = \frac{|\mathcal{D}_s^{<a}|}{|\mathcal{D}_s^{<a}| + \beta|\mathcal{D}_t^{<b}|} \quad (8)$$

The value of β can be determined by cross-validation. In our experiments, we set β to 1.

The above process can also be applied to adjust the response value of nodes as follows:

$$v'_f = pv_f + (1-p)f \quad (9)$$

where v'_f is the adapted response at a node, v_f is its original response value of source model, and f is the response value (Equation 5).

The complete Trada algorithm used in our experiments is presented in Algorithm 1.

Algorithm 1 Tree Adaptation Algorithm

TRADA$(v, \mathcal{D}_t, p)$
1 $b \leftarrow$ COMPUTE-THRESHOLD$(v_x, \mathcal{D}_t)$
2 $v'_a \leftarrow pv_a + (1-p)b$
3 $v'_f \leftarrow pv_f + (1-p)$MEAN-RESPONSE$(\mathcal{D}_t)$

4 $\mathcal{D}'_t \leftarrow \{x \in \mathcal{D}_t : x_i < v'_a\}$
5 $v'_< \leftarrow$ TRADA$(v_<, \mathcal{D}'_t, p)$

6 $\mathcal{D}''_t \leftarrow \{x \in \mathcal{D}_t : x_i > v'_a\}$
7 $v'_> \leftarrow$ TRADA$(v_>, \mathcal{D}''_t, p)$

8 **return** v'

The Trada algorithm can be augmented with a second phase which directly incorporates the target training data. Assume that our source model, $\mathcal{M}_s$, was trained using source data, $\mathcal{D}_s$. Recall that $\mathcal{M}_s$ can be decomposed as a sum of regression tree output, $f_{\mathcal{M}_s}(x) = f^1_{\mathcal{M}_s}(x) + \ldots + f^k_{\mathcal{M}_s}(x)$. *Additive tree adaptation* refers augmenting this summation with a set of regression trees trained on the residuals between the model, $\mathcal{M}_s$, and the target training data, $\mathcal{D}_t$. That is, $f_{\mathcal{M}_t}(x) = f^1_{\mathcal{M}_s}(x) + \ldots + f^k_{\mathcal{M}_s}(x) + f_{\mathcal{M}_t}(x)^{k+1} + \ldots + f_{\mathcal{M}_t}(x)^{k+k'}$. In order for us to perform additive tree adaptation, the source and target data must use the same absolute relevance grades.

4 Pairwise Adaptation

Both GBRANK and Trada can be used to reduce the requirement on editorial data. GBRANK achieves the goal by leveraging preference data, while Trada does so by leveraging data from a different search market. A natural extension to these methods is to leverage both sources of data simultaneously. However, no algorithm has been proposed to do this so far in the literature. We propose an adaptation method using pairwise preference data.

Our approach shares the same intuition as Trada: maintain the tree structure but adjust decision threshold values against some target value. However, an important difference is that our adjustment of threshold values does not regress against some target grade values; rather its objective is to improve the ordering of documents. To make use of preference data in the tree adaptation, we follow the method used in GBRANK to adjust the target values whenever necessary to preserve correct document order. Given a base model, $\mathcal{M}_s$, and preference data, $\mathcal{D}_t^{\succ}$, we can use Equations 3 and 4 to *infer* target values. Specifically, we construct a set $\mathcal{D}^{\succ}_{\text{contra}}$ from $\mathcal{D}_t^{\succ}$ and $\mathcal{M}_s$. For each item (q, d) in $\mathcal{D}^{\succ}_{\text{contra}}$, we use the value of $\tilde{f}(q, d)$ as the target. These tuples, $\langle (q, d), \tilde{f}(q, d) \rangle$ along with $\mathcal{M}_s$ are then are provided as input to Trada. Our approach is described in Algorithm 2.

Compared to Trada, Pairwise-Trada has two important differences. First, Pairwise-Trada can use a source GBDT model trained either against absolute or pairwise judgments. When an organization maintains a set of ranking models for different markets, although the underlying modeling method may be shared (e.g. GBDT), the learning algorithm used may be market-specific (e.g. GBRANK or GBDT_{reg}). Unfortunately, classic Trada relies on the source model being trained using GBDT_{reg}. Second, Pairwise-Trada can be adapted using pairwise judgments. This means that we can expand our adaptation data to include click feedback, which is easily obtainable in practice.

Algorithm 2 Pairwise Tree Adaptation Algorithm

PAIRWISE-TRADA($\mathcal{M}_s, \mathcal{D}_t^{\succ}, p$)

1 $\mathcal{D}_{\text{contra}} \leftarrow$ FIND-CONTRADICTIONS($\mathcal{M}_s, \mathcal{D}_t^{\succ}$)
2 $\tilde{\mathcal{D}}_t \leftarrow \{\langle (q, d), \tilde{f}(q, d) \rangle : (q, d) \in \mathcal{D}_{\text{contra}}\}$
3 **return** TRADA(ROOT($\mathcal{M}_s$), $\tilde{\mathcal{D}}_t, p$)

5 Methods and Materials

The proposed algorithm is a straightforward modification of previous ones. The question we want to examine in this section is whether this simple modification is effective in practice. In particular, we want to examine whether pairwise adaptation is better than the original adaptation Trada using grade data, and whether the pairwise data from a market can help improve the ranking function on a different market.

Our experiments evaluate the performance of Pairwise-Trada for web ranking in ten target markets. These markets, listed in Table 1, cover a variety of languages and cultures. Furthermore, resources, in terms of documents, judgments, and click-through data, also varies across markets. In particular, editorial query-document judgments range from hundreds of thousands (e.g. SEA_1) to tens of thousands (e.g. SEA_5). Editors graded query-document pairs on a five-point relevance scale, resulting in our data set $\mathcal{D}$. Preference labels, $\mathcal{D}^{\succ}$, are inferred from these judgments.

We also include a second set of experiments which incorporate click data.[1] In these experiments, we infer a preference from click data by assuming the following model. The user is presented with ten results. An item $i \succ j$ if i the following conditions hold: i is positioned below j, i receives a click, and j does not receive a click.

In our experiments, we tested the following runs,

- GBDT_{reg} trained using only $\mathcal{D}_s$ or $\mathcal{D}_t$
- GBRANK trained using only $\mathcal{D}_s^{\succ}$ or $\mathcal{D}_t^{\succ}$
- GBRANK trained using only $\mathcal{D}_s^{\succ}$, $\mathcal{D}_t^{\succ}$, and $\mathcal{C}_t$
- Trada with both GBDT_s and GBRANK_s, adapted with $\mathcal{D}_t$.
- Pairwise-Trada with both GBDT_s and GBRANK_s, adapted with $\mathcal{D}_t^{\succ}$ and $\mathcal{C}_t$ at different ratios.

In the all experiments, we use 400 additive trees when additive adaptation is used.

All models are evaluated using discounted cumulative gain (DCG) at rank cutoff 5 (Järvelin and Kekäläinen, 2002).

6 Results

6.1 Adaptation with Manually Labeled Data

In Table 1, we show the results for all of our experimental conditions.

We can make a few observations about the non-adaptation baselines. First, models trained on the (limited) target editorial data, GBDT_t and GBRANK_t, tend to outperform those trained only on the source editorial data, GBDT_s and GBRANK_s. The critical exception is SEA_5, the market with the fewest judgments. We believe that this behavior is a result of similarity between the United States source data and the SEA_5 target market; both the source and target query populations share the same language, a property not exhibited in other markets. Notice that other small markets such as LA_2 and LA_3 see modest improvements when using target-only runs compared to source-only runs. Second, GBRANK tends to outperform GBDT when only trained on the source data. This implies that we should prefer a base model which is based on GBRANK, something that is difficult to combine with classic Trada. Third, by comparing GBRANK and GBDT when only trained on the target data, we notice that the effectiveness of GBRANK depends on the amount of training data. For markets where there training data is plentiful (e.g. SEA_1), GBRANK outperforms GBDT. On the other hand, for smaller markets (e.g. LA_3), GBDT outperforms GBRANK.

In general, the results confirm the hypothesis that adaptation runs outperform all of non-adaptation baselines. This is the case for both Trada and Pairwise-Trada. As with the baseline runs, the Australian market sees different performance as a result of the combination of a small target editorial set and a representative source domain. This effect has been observed in previous results (Chen et al., 2009).

We can also make a few observations by comparing the adaptation runs. Trada works better with a GBDT base model than with a GBRANK base model. We We believe this is the case because the absolute regression targets are difficult to compare with the unbounded output of GBRANK. Pairwise-Trada on the other hand tends to perform better with a GBRANK base model than with a GBDT base model. There are a few exceptions, SEA_3 and LA_2, where Pairwise-Trada works better with a GBDT base model. Comparing Trada to Pairwise-Trada, we find that using preference targets tends to improve performance for some markets but not all. The underperformance of Pairwise-Trada tends to occur in smaller markets such as LA_1, LA_2, and LA_3. This is similar to the behavior we observed in the non-adaptation runs and suggests that, in operation, a modeler may have to decide on the training algorithm based on the amount of data available.

[1] For technical reasons, this data set is slightly different from the results we show with the purely editorial data. Therefore the size of the training and testing sets are different, but not to a significant degree.

	SEA_1	SEA_2	EU_1	SEA_3	EU_2	SEA_4	LA_1	LA_2	LA_3	SEA_5
training size	243,790	174,435	137,540	135,066	101,076	100,846	91,638	75,989	66,151	37,445
testing size	18,652	26,752	11,431	13,839	12,118	12,214	11,038	16,339	10,379	21,034
$GBDT_s$	9.4483	8.1271	9.0018	10.0630	8.5339	5.9176	6.1699	11.4167	8.1416	10.5356
$GBDT_t$	9.6011	8.6225	9.3310	10.7591	9.0323	6.4185	6.8441	11.8553	8.5702	10.4561
$GBRANK_s$	9.6059	8.1784	9.0775	10.2486	8.6248	6.1298	6.2614	11.5186	8.2851	10.5915
$GBRANK_t$	9.6952	8.6225	9.3575	10.8595	9.0384	6.4620	6.8543	11.7086	8.4825	10.3469
Trada										
$GBDT_s, \mathcal{D}_t$	**9.6718**	**8.6120**	**9.3086**	**10.8001**	**9.1024**	**6.3440**	**6.9444**	**11.9513**	**8.6519**	10.6279
$GBRANK_s, \mathcal{D}_t$	**9.6116**	**8.5681**	**9.2125**	**10.7597**	**8.9675**	**6.4110**	**6.8286**	**11.7326**	**8.5498**	10.6508
Pairwise-Trada										
$GBDT_s, \mathcal{D}_t$	**9.7364**	**8.6261**	**9.3824**	**10.8549**	**9.0842**	**6.4705**	**6.9438**	**11.8255**	**8.5323**	10.4655
$GBRANK_s, \mathcal{D}_t$	**9.7539**	**8.6538**	**9.4269**	**10.8362**	**9.1044**	**6.4716**	**6.9438**	**11.8034**	**8.6187**	10.6564

Table 1: Adaptation using manually labeled training data Southeast Asia (SEA), Europe (EU), and Latin America (LA) markets. Markets are sorted by target training set size. Significance tests use a t-test. Bolded numbers indicate statistically significant improvements over the respective source model.

	SEA_1	SEA_2	EU_1	SEA_3	EU_2	SEA_4	LA_1	LA_2	LA_3	SEA_5
training size	194,114	166,396	136,829	161,663	94,875	96,642	73,977	108,350	64,481	71,549
testing size	15,655	11,844	11,028	11,839	11,118	5,092	10,038	12,246	10,201	7,477
$GBRANK_s$	9.0159	8.5763	8.7119	11.4512	9.7641	6.5941	6.894	7.9366	8.058	10.7935
Pairwise-Trada										
$GBRANK_s, \mathcal{D}_t, \mathcal{C}_t$										
editorial	**9.3577**	**8.9205**	**8.901**	**12.2247**	9.9531	6.7421	**7.1455**	**8.2811**	**8.2503**	10.7973
click	9.1149	8.7622	8.8187	**11.9361**	9.8818	6.7703	**7.1812**	**8.264**	**8.2485**	10.9042
editorial+click	**9.4898**	**9.0177**	**8.945**	**12.3172**	**10.1156**	6.8459	**7.2414**	**8.4111**	**8.292**	**11.1407**

Table 2: Adaptation incorporating click data. Bolded numbers indicate statistically significant improvements over the baseline. Markets ordered as in Table 1.

6.2 Incorporating Click Data

One of the advantages of Pairwise-Trada is the ability to incorporate multiple sources of pairwise preference data. In this paper, we use the heuristic rule approach which is introduced by (Dong et al., 2009) to extract pairwise preference data from the click log of the search engine. This approach yields both skip-next and skip-above pairs (Joachims et al., 2005), which are sorted by confidence descending order respectively. In these experiments, we combine manually generated preferences with those gathered from click data. We present these results in Table 2.

We notice that no matter the source of preference data, Pairwise-Trada outperforms the baseline GBRANK model. The magnitude of the improvement depends on the source data used. Comparing the editorial-only to the click-only models, we notice that click-only models outperform editorial-only models for smaller markets (SEA_4, LA_1, and SEA_5). This is likely the case because the relative quantity of click data with respect to editorial data is higher in these markets. This is despite the fact that the click data may be noisier than the editorial data. The best performance, though, comes when we combine both editorial and click data.

6.3 Additive tree adaptation

Recall that Pairwise-Trada consists of two parts: parameter adaptation and additive tree adaptation. In this section, we examine the contribution to performance each part is responsible for. Figure 2 illustrates the adaptation results for the LA_1 market. In this experiment, we use a United States base model and 100K LA_1 editorial judgments for adaptation. Pairwise-Trada is performed on top of differently sized base models with 600, 900 and 1200 trees. The original base model has 1200 trees; we selected the first 600, 900 or full 1200 trees for experiments. The number of trees used in the additive tree adaptation step ranges up to 600 trees. From Figure 2 we can see that the additive adaptation can

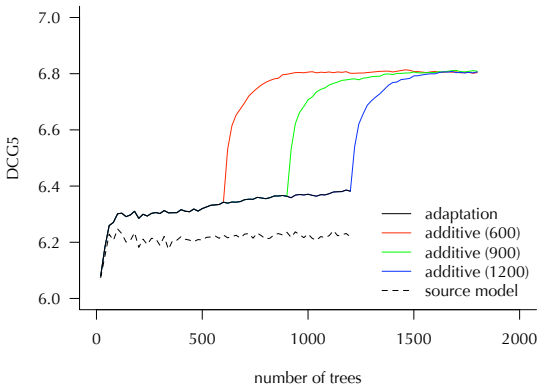


Figure 2: Illustration of additive tree adaptation for LA_1. The curves are average performance over a range of parameter settings.

significantly increase DCG over simple parameter adaptation and is therefore a critical step of Pairwise-Trada. When the number of trees in the additive tree adaptation step reaches roughly 400, the DCG plateaus.

7 Conclusion

We have proposed a model for adapting retrieval models using preference data instead of absolute relevance grades. Our experiments demonstrate that, when much editorial data is present, our method, Pairwise-Trada, may be preferable to competing methods based on absolute relevance grades. However, in real world systems, we often have access to sources of preference data beyond those resulting from editorial judgments. We demonstrated that Pairwise-Trada can exploit such data and boost performance significantly. In fact, *if we omit editorial data altogether we see performance improvements over the baseline model*. This suggests that, in principle, we can train a single, strong source model and improve it using target click data alone. Despite the fact that the modification we made is quite simple, we showed that modification is effective in practice. This tends to validate the general principle of using pairwise data from a different market. This principle can be easily used in other frameworks such as neural networks (Burges et al., 2005b). Therefore, the proposed method also points to a new direction for future improvements of search engines.

There are several areas of future work. First, we believe that detecting other sources of preference data from user behavior can further improve the performance of our model. Second, we only used a single source model in our experiments. We would also like to explore the effect of learning from an ensemble of source models. The importance of each may depend on the similarity to the target domain. Finally, we would also like to more accurately understand the queries where click data improves adaptation and those where editorial judgments is required. This sort of knowledge will allow us to train systems which maximally exploit our editorial resources.

References

Amini, M.-R., T.-V. Truong, and C. Goutte. 2008. A boosting algorithm for learning bipartite ranking functions with partially labeled data. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*.

Bacchiani, M. and B. Roark. 2003. Unsupervised language model adaptation. In *ICASSP '03: Proceedings of the International Conference on Acoustics, Speech and Signal Processing*.

Bai, J., K. Zhou, H. Zha, B. Tseng, Z. Zheng, and Y. Chang. 2009. Multi-task learning for learning to rank in web search. In *CIKM '09: Proceeding of the 18th ACM conference on Information and knowledge management*.

Blitzer, J., R. McDonald, and F. Pereira. 2006. Domain adaptation with structural correspondence learning. In *EMNLP '06: Proceedings of the 2006 Conference on Empirical Methods on Natural Language Processing*.

Burges, C., T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender. 2005a. Learning to rank using gradient descent. In *ICML '05: Proceedings of the 22nd International Conference on Machine learning*.

Burges, Chris, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. 2005b. Learning to rank using gradient descent. In *ICML '05: Proceedings of the*

22nd international conference on Machine learning, pages 89–96. ACM.

Cao, Z., T. Qin, T.-Y. Liu, M.-F. Tsai, and H. Li. 2007. from pairwise approach to listwise approach. In *ICML '07: Proceedings of the 24th international conference on Machine learning*.

Chen, D., J. Yan, G. Wang, Y. Xiong, W. Fan, and Z. Chen. 2008a. Transrank: A novel algorithm for transfer of rank learning. In *ICDM workshop '08: Proceeding of IEEE Conference on Data Mining*.

Chen, K., R. Lu, C. K. Wong, G. Sun, L. Heck, and B. Tseng. 2008b. Trada: tree based ranking function adaptation. In *CIKM '08: Proceeding of the 17th ACM conference on Information and knowledge management*, pages 1143–1152, New York, NY, USA. ACM.

Chen, W., T.-Y. Liu, Y. Lan, Z. Ma, and H. Li. 2008c. Measures and loss functions in learning to rank. In *NIPS '08: Proceedings of the Twenty-Second Annual Conference on Neural Information Processing Systems*.

Chen, K., J. Bai, S. Reddy, and B. Tseng. 2009. On domain similarity and effectiveness of adapting-to-rank. In *CIKM '09: Proceeding of the 18th ACM conference on Information and knowledge management*, pages 1601–1604, New York, NY, USA. ACM.

Dong, A., Y. Chang, S. Ji, C. Liao, X. Li, and Z. Zheng. 2009. Empirical exploitation of click data for query-type-based ranking. In *EMNLP '09: Proceedings of the 2009 Conference on Empirical Methods on Natural Language Processing*.

Duh, K. and K. Kirchhoff. 2008. Learning to rank with partially-labeled data. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*.

Freund, Y., R. D. Iyer, R. E. Schapire, and Y. Singer. 1998. An efficient boosting algorithm for combining preferences. In *ICML '98: Proceedings of the Fifteenth International Conference on Machine Learning*.

Friedman, J. H. 2001. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5):1189–1232.

Gao, J., Q. Wu, C. Burges, K. Svore, Y. Su, N. Khan, Shah S., and H. Zhou. 2009. Model adaptation via model interpolation and boosting for web search ranking. In *EMNLP '09: Proceedings of the 2009 Conference on Empirical Methods on Natural Language Processing*.

Geng, B., L. Yang, C. Xu, and X.-S. Hua. 2009. Ranking model adaptation for domain-specific search. In *CIKM '09: Proceeding of the 18th ACM conference on Information and knowledge management*, pages 197–206, New York, NY, USA. ACM.

Hwa, R. 1999. Supervised grammar induction using training data with limited constituent information. In *ACL '99: Proceedings of the Conference of the Association for Computational Linguistics*.

Järvelin, Kalervo and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of ir techniques. *TOIS*, 20(4):422–446.

Joachims, T., L. Granka, B. Pan, and G. Gay. 2005. Accurately interpreting clickthrough data as implicit feedback.

Joachims, T. 2002. Optimizing search engines using clickthrough data. In *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 133–142. ACM Press.

Liu, T.-Y. 2009. *Learning to Rank for Information Retrieval*. Now Publishers.

Radlinski, F. and T. Joachim. 2006. Minimally invasive randomization for collecting unbiased preferences from clickthrough logs.

Radlinski, F. and T. Joachim. 2007. Active exploration for learning rankings from clickthrough data.

Wu, M., Y. Chang, Z. Zheng, and H. Zha. 2009. Smoothing dcg for learning to rank: A novel approach using smoothed hinge functions. In *CIKM '09: Proceeding of the 18th ACM conference on Information and knowledge management*.

Xia, F., T.-Y. Liu, J. Wang, W. Zhang, and H. Li. 2008. Listwise approach to learning to rank: Theorem and algorithm. In *ICML '08: Proceedings of the 25th international conference on Machine learning*.

Xu, J., T.Y. Liu, M. Lu, H. Li, and W.Y. Ma. 2008. Directly optimizing evaluation measures in learning to rank. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*.

Zheng, Z., K. Chen, G. Sun, and H. Zha. 2007. A regression framework for learning ranking functions using relative relevance judgments. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 287–294. ACM.

Going Beyond Traditional QA Systems: Challenges and Keys in Opinion Question Answering

Alexandra Balahur
Dept. of Software and Computing Systems
University of Alicante
abalahur@dlsi.ua.es

Ester Boldrini
Dept. of Software and Computing Systems
University of Alicante
eboldrini@dlsi.ua.es

Andrés Montoyo
Dept. of Software and Computing Systems
University of Alicante
montoyo@dlsi.ua.es

Patricio Martínez-Barco
Dept. of Software and Computing Systems
University of Alicante
patricio@dlsi.ua.es

Abstract

The treatment of factual data has been widely studied in different areas of Natural Language Processing (NLP). However, processing subjective information still poses important challenges. This paper presents research aimed at assessing techniques that have been suggested as appropriate in the context of subjective - Opinion Question Answering (OQA). We evaluate the performance of an OQA with these new components and propose methods to optimally tackle the issues encountered. We assess the impact of including additional resources and processes with the purpose of improving the system performance on two distinct blog datasets. The improvements obtained for the different combination of tools are statistically significant. We thus conclude that the proposed approach is adequate for the OQA task, offering a good strategy to deal with opinionated questions.

1 Introduction

The State of the Blogosphere 2009 survey published by Technorati [1] concludes that in the past years the blogosphere has gained a high influence on a high variety of topics, ranging from cooking and gardening, to economics, politics and scientific achievements. The development of the Social Web and the new communication frameworks also influenced the way information is transmitted through communities. Blogs are part of the so-called new textual genres. They have distinctive features when compared to the traditional ones, such as newspaper articles. Blog language contains formal and informal expressions, and other elements, as repeated punctuation or emoticons (used to stress upon different text elements). With the growth in the content of the blogosphere, the quantity of subjective data of the Web is increasing exponentially (Cui et al., 2006). As it is being updated in real-time, this data becomes a source of timely information on many topics, exploitable by different applications. In order to properly manage the content of this subjective information, its processing must be automated. The NLP task, which deals with the classification of opinionated content is called Sentiment Analysis (SA). Research in this field aims at discovering appropriate mechanisms to properly retrieve, extract and classify opinions expressed in text. While techniques to retrieve objective information have been widely studied, implemented and evaluated, opinion-related tasks still represent an important challenge. As a consequence, the aim of our research is to study, implement and evaluate appropriate methods for the task of Question Answering (QA) in the opinion treatment framework.

2 Motivation and Contribution

Research in opinion-related tasks gained importance in the past years. However, there are still many aspects that require analysis and im-

[1] http://technorati.com/

Coling 2008: Poster Volume, pages 27–35,
Beijing, August 2010

provement, especially for approaches that combine SA with other NLP tasks such as QA or automatic summarization. The TAC 2008 Opinion Pilot task and the subsequent research performed on the competition data have demonstrated that answering opinionated questions and summarizing subjective information are significantly different from the equivalent tasks in the same context, but dealing with factual data. This finding was confirmed by the recent work by (Kabadjov et al., 2009). The first motivation of our work is the need to detect and explore the challenges raised by opinion QA (OQA), as compared to factual QA. To this aim, we analyze the improvements that can be brought at the different steps of the OQA process: *question treatment* (identification of expected polarity – EPT, expected source – ES and expected target –ET-), *opinion retrieval* (at the level of one and three-sentences long snippets, using topic-related words or using paraphrases), *opinion analysis* (using topic detection and anaphora resolution). This preliminary research is motivated by the conclusions drawn by previous studies (Balahur et al., 2009). Our purpose is to verify if the inclusion of new elements and methods - source and target detection (using semantic role labeling (SRL)), topic detection (using Latent Semantic Analysis), paraphrasing and joint topic-sentiment analysis (classification of the opinion expressed only in sentences related to the topic), followed by anaphora resolution (using a system whose performance is not optimal), affects the results of the system and how. Our contribution to this respect is the identification of the challenges related to OQA compared to traditional QA. A further contribution consists in adding the appropriate methods, tools and resources to resolve the identified challenges. With the purpose of testing the effect of each tool, resource and technique, we carry out a separate and a global evaluation. An additional motivation of our work is the fact that although previous approaches showed that opinion questions have longer answers than factual ones, the research done in OQA so far has only considered a sentence-level approach. Another contribution this paper brings is the retrieval at 1 and 3-sentence level and the retrieval based on similarity to query paraphrases enriched with topic-related words). We believe retrieving longer text could cause additional problems such as redundancy, coreference and temporal expressions or the need to apply contextual information. Paraphrasing, on the other hand, had account for language variability in a more robust manner; however, the paraphrase collections that are available at the moment are known to be noisy. The following sections are structured as follows: Section 3 presents the related work in the field and the competitions organized for systems tackling the OQA task. In Section 4 we describe the corpora used for the experiments we carried out and the set of questions asked over each of them. Section 5 presents the experimental settings and the different system configurations we assessed. Section 6 shows the results of the evaluations, discusses the improvements and drops in performance using different configurations. We finally conclude on our approaches in Section 7, proposing the lines for future work.

3 Related Work

QA can be defined as the task in which given a set of questions and a collection of documents, an automatic NLP system is employed to retrieve the answer to the queries in Natural Language (NL). Research focused on building factoid QA systems has a long tradition; however, it is only recently that researchers have started to focus on the development of OQA systems. (Stoyanov et al., 2005) and (Pustejovsky and Wiebe, 2006) studied the peculiarities of opinion questions. (Cardie et al., 2003) employed opinion summarization to support a Multi-Perspective QA system, aiming at identifying the opinion-oriented answers for a given set of questions. (Yu and Hatzivassiloglou, 2003) separated opinions from facts and summarized them as answer to opinion questions. (Kim and Hovy, 2005) identified opinion holders, which are a key component in retrieving the correct answers to opinion questions. Due to the realized importance of blog data, recent years have also marked the beginning of NLP research focused on the development of opinion QA systems and the organization of international conferences encouraging the creation of effective QA systems both for fact and subjective texts. The TAC 2008[2] QA track proposed a collection

[2] http://www.nist.gov/tac/

of factoid and opinion queries called "rigid list" (factoid) and "squishy list" (opinion) respectively, to which the traditional QA systems had to be adapted. Some participating systems treated opinionated questions as "other" and thus they did not employ opinion specific methods. However, systems that performed better in the "squishy list" questions than in the "rigid list" implemented additional components to classify the polarity of the question and of the extracted answer snippet. The Alyssa system (Shen et al, 2007) uses a Support Vector Machines (SVM) classifier trained on the MPQA corpus (Wiebe et al., 2005), English NTCIR3 data and rules based on the subjectivity lexicon (Wilson et al., 2005). (Varma et al., 2008) performed query analysis to detect the polarity of the question using defined rules. Furthermore, they filter opinion from fact retrieved snippets using a classifier based on Naïve Bayes with unigram features, assigning for each sentence a score that is a linear combination between the opinion and the polarity scores. The PolyU (Venjie et al., 2008) system determines the sentiment orientation of the sentence using the Kullback-Leibler divergence measure with the two estimated language models for the positive versus negative categories. The QUANTA (Li et al., 2008) system performs opinion question sentiment analysis by detecting the opinion holder, the object and the polarity of the opinion. It uses a semantic labeler based on PropBank[4] and manually defined patterns. Regarding the sentiment classification, they extract and classify the opinion words. Finally, for the answer retrieval, they score the retrieved snippets depending on the presence of topic and opinion words and only choose as answer the top ranking results. Other related work concerns opinion holder and target detection. NTCIR 7 and 8 organized MOAT (the Multilingual Opinion Analysis Task), in which most participants employed machine learning approaches using syntactic patterns learned on the MPQA corpus (Wiebe et al., 2005). Starting from the abovementioned research, our aim is to take a step forward to present approaches and employ opinion specific methods focused on improving the performance of our OQA. We perform the retrieval at 1 sentence and 3 sentence-level and also determine the expected source (ES) and the expected target (ET) of the questions, which are fundamental to properly retrieve the correct answer. These two elements are selected employing semantic roles (SR). The expected answer type (EAT) is determined using Machine Learning (ML) using Support Vector Machine (SVM), by taking into account the interrogation formula, the subjectivity of the verb and the presence of polarity words in the target SR. In the case of expected opinionated answers, we also compute the expected polarity type (EPT) – by applying opinion mining (OM) on the affirmative version of the question (e.g. for the question *"Why do people prefer Starbucks to Dunkin Donuts?"*, the affirmative version is *"People prefer Starbucks to Dunkin Donuts because X"*). These experiments are presented in more detail in Section 5.

4 Corpora

In order to carry out the present research for detecting and solving the complexities of opinion QA, we employed two corpora of blog posts: *EmotiBlog* (Boldrini *et al.*, 2009a) and the TAC 2008 Opinion Pilot test collection (part of the Blog06 corpus).

The TAC 2008 Opinion Pilot test collection is composed by documents with the answers to the opinion questions given on 25 targets. *EmotiBlog* is a collection of blog posts in English extracted form the Web. As a consequence, it represents a genuine example of this textual genre. It consists in a monothematic corpus about the Kyoto Protocol, annotated with the improved version of *EmotiBlog* (Boldrini *et al.*, 2009b). It is well know that Opinion Mining (OM) is a very complex task due to the high variability of the language employed. Thus, our objective is to build an annotation model that is able to capture the whole range of phenomena specific to subjectivity expression. Additional criteria employed when choosing the elements to be annotated were effectiveness and noise minimization. Thus, from the first version of the model, the elements which did not prove to be statistically relevant have been eliminated. The elements that compose the improved version of the annotation model are presented in Table 1.

[3] http://research.nii.ac.jp/ntcir/

[4] http://verbs.colorado.edu/~mpalmer/projects/ace.html

Elements	Description
Obj. speech	Confidence, comment, source, target.
Subj. speech	Confidence, comment, level, emotion, phenomenon, polarity, source and target.
Adjectives/Adverbs	Confidence, comment, level, emotion, phenomenon, modifier/not, polarity, source and target.
Verbs/ Names	Confidence, comment, level, emotion, phenomenon, polarity, mode, source and target.
Anaphora	Confidence, comment, type, source and target.
Capital letter/ Punctuation	Confidence, comment, level, emotion, phenomenon, polarity, source and target.
Phenomenon	Confidence, comment, type, collocation, saying, slang, title, and rhetoric.
Reader/Author Interpr. (obj.)	Confidence, comment, level, emotion, phenomenon, polarity, source and target.
Emotions	Confidence, comment, accept, anger, anticipation, anxiety, appreciation, bad, bewilderment, comfort, compassion…

Table 1: *EmotiBlog* improved structure

The first distinction consists in separating objective and subjective speech. Subsequently, a finer-grained annotation is employed for each of the two types of data. Objective sentences are annotated with source and target (when necessary, also the level of confidence of the annotator and a comment). Subjective elements can be annotated at a sentence level, but they also have to be labeled at a word and/or phrase level. *EmotiBlog* also contains annotations of anaphora at a cross-document level (to interpret the storyline of the posts) and the sentence type (simple sentence or title, but also saying or collocation). Finally, the Reader and the Writer interpretation have to be marked in objective sentences. These elements are employed to mark and interpret correctly an apparent objective discourse, whose aim is to implicitly express an opinion (e.g. *"The camera broke in two days"*). The first is useful to extract what is the interpretation of the reader (for example if the writer says *The result of their governing was an increase of 3.4% in the unemployment rate* instead of *The result of their governing was a disaster for the unemployment rate*) and the second to understand the background of the reader (i.e.. *These criminals are not able to govern* instead of saying *the x party is not able to govern*). From this sentence, for example, the reader can deduce the political ideas of the writer. The questions whose answers are annotated with *EmotiBlog* are the subset of opinion questions in English presented in (Balahur et al., 2009). The complete list of questions is shown in Table 2.

N	Question
2	What motivates people's negative opinions on the Kyoto Protocol?
5	What are the reasons for the success of the Kyoto Protocol?
6	What arguments do people bring for their criticism of media as far as the Kyoto Protocol is concerned?
7	Why do people criticize Richard Branson?
11	What negative opinions do people have on Hilary Benn?
12	Why do Americans praise Al Gore's attitude towards the Kyoto protocol?
15	What alternative environmental friendly resources do people suggest to use instead of gas en the future?
16	Is Arnold Schwarzenegger pro or against the reduction of CO2 emissions?
18	What improvements are proposed to the Kyoto Protocol?
19	What is Bush accused of as far as political measures are concerned?
20	What initiative of an international body is thought to be a good continuation for the Kyoto Protocol?

Table 2: Questions over the *EmotiBlog* corpus

The main difference between the two corpora employed is that *Emotiblog* is monothematic, containing only posts about the Kyoto Protocol, while the TAC 2008 corpus contains documents on a multitude of subjects. Therefore, different techniques must be adjusted in order to treat each of them.

5 Experiments

5.1 Question Analysis

In order to be able to extract the correct answer to opinion questions, different elements must be considered. As stated in (Balahur et al., 2009) we need to determine both the expected answer type (EAT) of the question – as in the case of factoid ones - as well as new elements – such as expected polarity type (EPT). However, opinions are directional – i.e., they suppose the existence of a source and a target to which they are addressed. Thus, we introduce two new elements in the question analysis – expected source (ES) and expected target (ET). These two elements are selected by applying SR and choosing the source as the agent in the sentence and the direct object (patient) as the target of the opinion. Of course, the source and target of the

opinions expressed can also be found in other roles, but at this stage we only consider these cases. The expected answer type (EAT) (e.g. opinion or other) is determined using Machine Learning (ML) using Support Vector Machine (SVM), by taking into account the interrogation formula, the subjectivity of the verb and the presence of polarity words in the target SR. In the case of expected opinionated answers, we also compute the expected polarity type (EPT) – by applying OM on the affirmative version of the question. An example of such a transformation is: given the question *"What are the reasons for the success of the Kyoto Protocol?"*, the affirmative version of the question is *"The reasons for the success of the Kyoto Protocol are X".*

5.2 Candidate Snippet Retrieval

In the answer retrieval stage, we employ four strategies:

1. Using the JIRS (JAVA Information Retrieval System) IR engine (Gómez et al., 2007) to find relevant snippets. JIRS retrieves passages (of the desired length), based on searching the question structures (n-grams) instead of the keywords, and comparing them.

2. Using the "Yahoo" search engine to retrieve the first 20 documents that are most related to the query. Subsequently, we apply LSA on the retrieved documents and extract the words that are most related to the topic. Finally, we expand the query using words that are very similar to the topic and retrieve snippets that contain at least one of them and the ET.
3. Generating equivalent expressions for the query, using the DIRT paraphrase collection (Lin and Pantel, 2001) and retrieving candidate snippets of length 1 and 3 (length refers to the number of sentences retrieved) that are similar to each of the new generated queries and contain the ET. Similarity is computed using the cosine measure. Examples of alternative queries for *"People like George Clooney"* are *"People adore George Clooney", "People enjoy George Clooney", "People prefer George Clooney"*.
4. Enriching the equivalent expressions for the query in 3. with the topic-related words discovered in 2. using LSA.

5.3 Polarity and topic-polarity classification of snippets

In order to determine the correct answers from the collection of retrieved snippets, we must filter for the next processing stage only the candidates that have the same polarity as the question EPT. For polarity detection, we use a combined system employing SVM ML on unigram and bigram features trained on the NTCIR MOAT 7 data and an unsupervised lexicon-based system. In order to compute the features for each of the unigrams and bigrams, we compute the tf-idf scores.

The unsupervised system uses the Opinion Finder lexicon to filter out subjective sentences – that contain more than two subjective words or a subjective word and a valence shifter (obtained from the General Inquirer resource). Subsequently, it accounts for the presence of opinionated words from four different lexicons – MicroWordNet (Cerini et al., 2007), WordNet Affect (Strapparava and Valitutti, 2004) Emotion Triggers (Balahur and Montoyo, 2008) and General Inquirer (Stone et al., 1966). For the joint topic-polarity analysis, we first employ LSA to determine the words that are strongly associated to the topic, as described in Section 5.2 (second list item). Consequently, we compute the polarity of the sentences that contain at least one topic word and the question target.

5.4 Filtering using SR

Finally, answers are filtered using the *Semrol* system for SR labeling described in (Moreda, 2008). Subsequently, we filter all snippets with the required target and source as agent or patient. *Semrol* receives as input plain text with information about grammar, syntax, word senses, Named Entities and constituents of each verb. The system output is the given text, in which the semantic roles information of each constituent is marked. Ambiguity is resolved

depending on the machine algorithm employed, which in this case is TIMBL[5].

6 Evaluation and Discussion

We evaluate our approaches on both the *EmotiBlog* question collection, as well as on the TAC 2008 Opinion Pilot test set. We compare them against the performance of the system evaluated in (Balahur et al., 2009) and the best (Copeck et al., 2008) and worst (Varma et al., 2008) scoring systems (as far as F-measure is concerned) in the TAC 2008 task. For both the TAC 2008 and *EmotiBlog* sets of questions, we employ the SR system in SA and determine the ES, ET and EPT. Subsequently, for each of the two corpora, we retrieve 1-phrase and 3-phrase snippets. The retrieval of the of the *EmotiBlog* candidate snippets is done using query expansion with LSA and filtering according to the ET. Further on, we apply sentiment analysis (SA) using the approach described in Section 5.3 and select only the snippets whose polarity is the same as the determined question EPT. The results are presented in Table 3.

Q No.	**No.** A	**Baseline** (Balahur et al., 2009)				**1 phrase + ET+SA**				**3 phrases +ET+SA**			
		@1	@5	@10	@50	@1	@5	@10	@50	@1	@5	@10	@20
2	5	0	2	3	4	1	2	3	4	1	2	3	4
5	11	0	0	0	0	0	2	2	2	1	2	3	4
6	2	0	0	1	2	1	1	2	2	0	1	2	2
7	5	0	0	1	3	1	1	1	3	0	2	2	4
11	2	1	1	1	1	0	0	0	0	0	0	0	1
12	3	0	1	1	1	0	1	2	3	0	0	1	2
15	1	0	0	1	1	0	0	1	1	1	1	1	1
16	6	1	4	4	4	0	1	1	2	1	2	2	6
18	1	0	0	0	0	0	0	0	0	0	0	0	0
19	27	1	5	6	18	0	1	1	2	0	1	1	1
20	4	0	0	0	0	0	0	1	1	0	0	1	2

Table 3: Results for questions over EmotiBlog

[5] http://ilk.uvt.nl/downloads/pub/papers/Timbl_6.2_Manual.pdf and http://ilk.uvt.nl/timbl/

The retrieval of the TAC 2008 1-phrase and 3-phrase candidate snippets was done using JIRS and, in a second approach, using the cosine similarity measure between alternative queries generated using paraphrases and candidate snippets. Subsequently, we performed different evaluations, in order to assess the impact of using different resources and tools. Since the TAC 2008 had a limit of the output of 7000 characters, in order to compute a comparable F-measure, at the end of each processing chain, we only considered the snippets for the 1-phrase retrieval and for the 3-phases one until this limit was reached.

1. In the first evaluation, we only apply the sentiment analysis tool and select the snippets that have the same polarity as the question EPT and the ET is found in the snippet. (i.e. *What motivates peoples negative opinions on the Kyoto Protocol? The Kyoto Protocol becomes deterrence to economic development and international cooperation/ Secondly, in terms of administrative aspect, the Kyoto Protocol is difficult to implement.* - same EPT and ET)
   We also detected cases of same polarity but no ET, e.g. *These attempts mean annual expenditures of $700 million in tax credits in order to endorse technologies, $3 billion in developing research and $200 million in settling technology into developing countries* – EPT negative but not same ET.
2. In the second evaluation, we add the result of the LSA process to filter out the snippets from 1., containing the words related to the topic starting from the retrieval performed by Yahoo, which extracts the first 20 documents about the topic.
3. In the third evaluation, we filter the results in 2 by applying the *Semrol* system and setting the condition that the ET and ES are the agent or the patient of the snippet.
4. In the fourth evaluation setting, we replaced the set of snippets retrieved using JIRS with the ones obtained by generating alternative queries using paraphrases (as explained in the third method in section 5.2.). We subsequently filtered these results based on their polarity (so that it corresponds to the EPT) and on the condition that the source and target of the opinion (identified through SRL using Semrol) correspond to the ES and ET.

5. In the fourth evaluation setting, we replaced the set of snippets retrieved using JIRS with the ones obtained by generating alternative queries using paraphrases, enriched with the topic words determined using LSA. We subsequently filtered these results based on their polarity (so that it corresponds to the EPT) and on the condition that the source and target of the opinion (identified through SRL using Semrol) correspond to the ES and ET.

System	F-measure
Best TAC	0.534
Worst TAC	0.101
JIRS + SA+ET (1 phrase)	0.377
JIRS + SA+ET (3 phrases)	0.431
JIRS + SA+ET+LSA (1 phrase)	0.489
JIRS + SA+ET+LSA (3 phrases)	0.505
JIRS + SA+ET+LSA+SR (1 phrase)	0. 533
JIRS + SA+ET+LSA+SR (3 phrases)	0.571
PAR+SA+ET+SR(1 phrase)	0.345
PAR+SA+ET+SR(2 phrase)	0.386
PAR_LSA+SA+ET+SR (1 phrase)	0.453
PAR_LSA+SA+ET+SR (3 phrases)	0.434

Table 4: Results for the TAC 2008 test set

From the results obtained (Table 3 and Table 4), we can draw the following conclusions. Firstly, the hypothesis that OQA requires the retrieval of longer snippets was confirmed by the improved results, both in the case of *EmotiBlog*, as well as the TAC 2008 corpus. Secondly, opinion questions require the use of joint topic-sentiment analysis. As we can see from the results, the use of topic-related words when computing of the affect influences the results in a positive manner and joint topic-sentiment analysis is especially useful for the cases of questions asked on a monothematic corpus. Thirdly, another conclusion that we can draw is that target and source detection are highly relevant steps at the time of answer filtering, not only helping the more accurate retrieval of answers, but also at placing at the top of the retrieval the relevant results (as more relevant information is contained within these 7000 characters). The use of paraphrases at the retrieval stage was shown to produce a significant drop in results, which we explain by the noise introduced and the fact that more non-relevant answer candidates were introduced among the results. Nonetheless, as we can see from the overall relatively low improvement in the results, much remains to be done in order to appropriately tackle OQA. As seen in the results, there are still questions for which no answer is found (e.g. 18). This is due to the fact that the treatment of such questions requires the use of inference techniques that are presently unavailable (i.e. define terms such as *"improvement"*, possibly as *"X better than Y"*, in which case opinion extraction from comparative sentences should be introduced in the model).

The results obtained when using all the components for the 3-sentence long snippets significantly improve the results obtained by the best system participating in the TAC 2008 Opinion Pilot competition (determined using a paired t-test for statistical significance, with confidence level 5%). Finally, from the analysis of the errors, we could see that even though some tools are in theory useful and should produce higher improvements – such as SR – their performance in reality does not produce drastically higher results. The idea to use paraphrases for query expansion also proved to decrease the system performance. From preliminary results obtained using JavaRap[6] for coreference resolution, we also noticed that the performance of the OQA lowered, although theoretically it should have improved.

7 Conclusions ad Future Work

In this paper, we presented and evaluated different methods and techniques with the objective of improving the task of QA in the context of opinion data. From the evaluations performed using different NLP resources and tools, we concluded that joint topic-sentiment analysis, as well as the target and source identification, are crucial for the correct performance of this task. We have also demonstrated that by retrieving longer answers, the results have improved. We tested, within a simple setting, the impact of using paraphrases in the context of opinion questions and saw that their use lowered the system results. Although such paraphrase col-

[6]http://wing.comp.nus.edu.sg/~qiu/NLPTools/JavaRAP.htm

lections include a lot of noise and have been shown to decrease system performance even in the case of factual questions, we believe that other types of paraphrasing methods should be investigated in the context of OQA. We thus showed that opinion QA requires the development of appropriate strategies at the different stages of the task (recognition of subjective questions, detection of subjective content of the questions, source and target identification, retrieval and classification of the candidate answer data). Due to the high level of complexity of subjective language, our future work will be focused on testing higher-performing tools for coreference resolution, other (opinion) paraphrases collections and paraphrasing methods and the employment of external knowledge sources that refine the semantics of queries. We also plan to include other SA methods and extend the semantic roles considered for ET and ES, with the purpose of checking if they improve or not the performance of the QA system.

Acknowledgements

This paper has been partially supported by Ministerio de Ciencia e Innovación - Spanish Government (grant no. TIN2009-13391-C04-01), and Conselleria d'Educación - Generalitat Valenciana (grant no. PROMETEO/2009/119 and ACOMP/2010/286).

References

Balahur, A. and Montoyo, A. 2008. *Applying a Culture Dependent Emotion Triggers Database for Text Valence and Emotion Classification.* In Proceedings of the AISB 2008 Symposium on Affective Language in Human and Machine, Aberdeen, Scotland.

Balahur, A., Lloret, E., Ferrández, O., Montoyo, A., Palomar, M., and Muñoz, R. 2008. *The DLSIUAES Team's Participation in the TAC 2008 Tracks.* In Proceedings of the Text Analysis Conference 2008 Workshop.

Balahur, A., Boldrini, E., Montoyo A. and Martínez-Barco P. 2009. *Opinion and Generic Question Answering Systems: a Performance Analysis.* In Proceedings of ACL. Singapur.

Boldrini, E., Balahur, A., Martínez-Barco, P. and Montoyo. A. 2009a. *EmotiBlog: an Annotation Scheme for Emotion Detection and Analysis in Non-traditional Textual Genre.* In Proceedings of DMIN 2009, Las Vegas. Nevada.

Boldrini, E., Balahur, A., Martínez-Barco, P. and Montoyo. A. 2009b. *EmotiBlog: a fine-grained model for emotion detection in non-traditional textual genre.* In Proceedings of WOMSA 2009. Seville.

Cardie, C., Wiebe, J., Wilson, T. and Litman, D. 2003. *Combining Low-Level and Summary Representations of Opinions for Multi-Perspective Question Answering.* AAAI Spring Symposium on New Directions in Question Answering.

Cerini, S., Compagnoni, V., Demontis, A., Formentelli, M. and Gandini, C. 2007. *Micro-WNOp: A gold standard for the evaluation of automatically compiledlexical resources for opinion mining*. In: A.Sanso (ed.): Language resources and linguistic theory: Typology, Second Language Acquisition, English Linguistics. Milano. IT.

Copeck, T., Kazantseva, A., Kennedy, A., Kunadze, A., Inkpen, D. and Szpakowicz, S. 2008. *Update Summary Update*. In Proceedings of the Text Analysis Conference (TAC) 2008.

Cui, H., Mittal, V. and Datar, M. 2006. *Comparative Experiments on Sentiment Classification for Online Product Review.* Proceedings, The Twenty-First National Conference on Artificial Intelligence and the Eighteenth Innovative Applications of Artificial Intelligence Conference. Boston, Massachusetts, USA.

Gómez, J.M., Rosso, P. and Sanchis, E. 2007. *JIRS Language-Independent Passage Retrieval System: A Comparative Study.* 5th International Conference on Natural Language Proceeding (ICON 2007).

Kabadjov, M., Balahur, A. And Boldrini, E. 2009. *Sentiment Intensity: Is It a Good Summary Indicator?.* Proceedings of the 4th Language Technology Conference LTC, pp. 380-384. Poznan, Poland, 6-8.11.2009.

Kim, S. M. and Hovy, E. 2005. *Identifying Opinion Holders for Question Answering in Opinion Texts.* Proceedings of the Workshop on Question Answering in Restricted Domain at the Conference of the American Association of Artificial Intelligence (AAAI-05). Pittsburgh, PA.

Li, F., Zheng, Z.,Yang T., Bu, F., Ge, R., Zhu, X., Zhang, X., and Huang, M. 2008. *THU QUANTA at TAC 2008. QA and RTE track.* In Proceedings of the Text Analysis Conference (TAC).

Lin, D. and Pantel, P. 2001. *Discovery of Inference Rules for Question Answering.* Natural Language Engineering 7(4):343-360.

Moreda. P. 2008. *Los Roles Semánticos en la Tecnología del Lengauje Humano: Anotación y Aplicación.* Doctoral Thesis. University of Alicante.

Pustejovsky, J. and Wiebe, J. 2006. *Introduction to Special Issue on Advances in Question Answering.* Language Resources and Evaluation (2005), (39).

Shen, D., Wiegand, M., Merkel, A., Kazalski, S., Hunsicker, S., Leidner, J. L. and Klakow, D. 2007. *The Alyssa System at TREC QA 2007: Do We Need Blog06?* In Proceedings of the Sixteenth Text Retrieval Conference (TREC 2007), Gaithersburg, MD, USA.

Strapparava, C. and Valitutti, A. 2004. *WordNet-Affect: an affective extension of WordNet*. In Proceedings of 4th International Conference on Language Resources and Evaluation (LREC 2004), pages 1083 – 1086, Lisbon.

Stoyanov, V., Cardie, C., and Wiebe, J. 2005. *Multiperspective question answering using the opqa corpus*. In Proceedings of the Human Language Technology Conference and the Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP 2005).

Varma, V., Pingali, P., Katragadda, S., Krishna, R., Ganesh, S., Sarvabhotla, K. Garapati, H., Gopisetty, H., Reddy, K. and Bharadwaj, R. 2008. *IIIT Hyderabad at TAC 2008*. In Proceedings of Text Analysis Conference (TAC).

Wenjie, L., Ouyang, Y., Hu, Y. and Wei, F. 2008. *PolyU at TAC 2008.* In Proceedings of the Text Analysis Conference (TAC).

Wiebe, J., Wilson, T., and Cardie, C. 2005. *Annotating expressions of opinions and emotions in language.* Language Resources and Evaluation, volume 39, issue 2-3, pp. 165-210.

Wilson, T., J. Wiebe, and Hoffmann, P. 2005. *Recognizing Contextual Polarity in Phrase-level sentiment Analysis.* In Proceedings of the Human Language Technologies Conference/Conference on Empirical Methods in Natural Language Processing (HLT/ EMNLP).

Yu, H. and Hatzivassiloglou, V. 2003. *Towards Answering Opinion Questions: Separating Facts from Opinions.* In Proceedings of EMNLP-03.

Wiebe, J., Wilson, T., and Cardie, C. (2005). *Annotating expressions of opinions and emotions in language.* In Language Resources and Evaluation. Vol. 39.

Robust Sentiment Detection on Twitter from Biased and Noisy Data

Luciano Barbosa
AT&T Labs - Research
lbarbosa@research.att.com

Junlan Feng
AT&T Labs - Research
junlan@research.att.com

Abstract

In this paper, we propose an approach to automatically detect sentiments on Twitter messages (tweets) that explores some characteristics of how tweets are written and meta-information of the words that compose these messages. Moreover, we leverage sources of noisy labels as our training data. These noisy labels were provided by a few sentiment detection websites over twitter data. In our experiments, we show that since our features are able to capture a more abstract representation of tweets, our solution is more effective than previous ones and also more robust regarding biased and noisy data, which is the kind of data provided by these sources.

1 Introduction

Twitter is one of the most popular social network websites and has been growing at a very fast pace. The number of Twitter users reached an estimated 75 million by the end of 2009, up from approximately 5 million in the previous year. Through the twitter platform, users share either information or opinions about personalities, politicians, products, companies, events (Prentice and Huffman, 2008) etc. This has been attracting the attention of different communities interested in analyzing its content.

Sentiment detection of tweets is one of the basic analysis utility functions needed by various applications over twitter data. Many systems and approaches have been implemented to automatically detect sentiment on texts (e.g., news articles, Web reviews and Web blogs) (Pang et al., 2002; Pang and Lee, 2004; Wiebe and Riloff, 2005; Glance et al., 2005; Wilson et al., 2005). Most of these approaches use the raw word representation (n-grams) as features to build a model for sentiment detection and perform this task over large pieces of texts. However, the main limitation of using these techniques for the Twitter context is messages posted on Twitter, so-called tweets, are very short. The maximum size of a tweet is 140 characters.

In this paper, we propose a 2-step sentiment analysis classification method for Twitter, which first classifies messages as subjective and objective, and further distinguishes the subjective tweets as positive or negative. To reduce the labeling effort in creating these classifiers, instead of using manually annotated data to compose the training data, as regular supervised learning approaches, we leverage sources of noisy labels as our training data. These noisy labels were provided by a few sentiment detection websites over twitter data. To better utilize these sources, we verify the potential value of using and combining them, providing an analysis of the provided labels, examine different strategies of combining these sources in order to obtain the best outcome; and, propose a more robust feature set that captures a more abstract representation of tweets, composed by meta-information associated to words and specific characteristics of how tweets are written. By using it, we aim to handle better: the problem of lack of information on tweets, helping on the generalization process of the classification algorithms; and the noisy and biased labels provided by those websites.

The remainder of this paper is organized as follows. In Section 2, we provide some context about messages on Twitter and about the websites used as label sources. We introduce the features used in the sentiment detection and also provide a deep analysis of the labels generated by those sources in Section 3. We examine different strategies of

Coling 2008: Poster Volume, pages 36–44,
Beijing, August 2010

combining these sources and present an extensive experimental evaluation in Section 4. Finally, we discuss previous works related to ours in Section 5 and conclude in Section 6, where we outline directions and future work.

2 Preliminaries

In this section, we give some context about Twitter messages and the sources used for our data-driven approach.

Tweets. The Twitter messages are called tweets. There are some particular features that can be used to compose a tweet (Figure 1 illustrates an example): "RT" is an acronym for retweet, which means the tweet was forwarded from a previous post; "@twUser" represents that this message is a reply to the user "twUser"; "#obama" is a tag provided by the user for this message, so-called hashtag; and "http://bit.ly/9K4n9p" is a link to some external source. Tweets are limited to 140 characters. Due to this lack of information in terms of words present in a tweet, we explore some of the tweet features listed above to boost the sentiment detection, as we will show in detail in Section 3.

Data Sources. We collected data from 3 different websites that provide almost real-time sentiment detection for tweets: Twendz, Twitter Sentiment and TweetFeel. To collect data, we issued a query containing a common stopword "of", as we are interested in collecting generic data, and retrieved tweets from these sites for three weeks, archiving the returned tweets along with their sentiment labels. Table 1 shows more details about these sources. Two of the websites provide 3-class detection: positive, negative and neutral and one of them just 2-class detection. One thing to note is our crawling process obtained a very different number of tweets from each website. This might be a result of differences among their sampling processes of Twitter stream or some kind of filtering process to output. For instance, a site may only present the tweets it has more confidence about their sentiment. In Section 3, we present a deep analysis of the data provided by these sources, showing if they are useful to build a sentiment classification.

```
RT @twUser: Obama  is the first U.S. president not to
have seen a new state added in his lifetime.
http://bit.ly/9K4n9p  #obama
```

Figure 1: Example of a tweet.

3 Twitter Sentiment Detection

Our goal is to categorize a tweet into one of the three sentiment categories: positive, neutral or negative. Similar to (Pang and Lee, 2004; Wilson et al., 2005), we implement a 2-step sentiment detection framework. The first step targets on distinguishing subjective tweets from non-subjective tweets (subjectivity detection). The second one further classifies the subjective tweets into positive and negative, namely, the polarity detection. Both classifiers perform prediction using an abstract representation of the sentences as features, as we show later in this section.

3.1 Features

A variety of features have been exploited on the problem of sentiment detection (Pang and Lee, 2004; Pang et al., 2002; Wiebe et al., 1999; Wiebe and Riloff, 2005; Riloff et al., 2006) including unigrams, bigrams, part-of-speech tags etc. A natural choice would be to use the raw word representation (n-grams) as features, since they obtained good results in previous works (Pang and Lee, 2004; Pang et al., 2002) that deal with large texts. However, as we want to perform sentiment detection on very short messages (tweets), this strategy might not be effective, as shown in our experiments. In this context, we are motivated to develop an abstract representation of tweets. We propose the use of two sets of features: meta-information about the words on tweets and characteristics of how tweets are written.

Meta-features. Given a word in a tweet, we map it to its part-of-speech using a part-of-speech dictionary[1]. Previous approaches (Wiebe and Riloff, 2005; Riloff et al., 2003) have shown that the effectiveness of using POS tags for this task. The intuition is certain POS tags are good indicators for sentiment tagging. For example, opinion messages are more likely containing adjec-

[1]The pos dictionary we used in this paper is available at: http://wordlist.sourceforge.net/pos-readme.

Data sources	URL	# Tweets	Sentiments
Twendz	http://twendz.waggeneredstrom.com/	254081	pos/neg/neutral
Twitter Sentiment	http://twittersentiment.appspot.com/	79696	pos/neg/neutral
TweetFeel	http://www.tweetfeel.com/	13122	pos/neg

Table 1: Information about the 3 data sources.

tives or interjections. In addition to POS tags, we map the word to its prior subjectivity (weak and strong subjectivity), also used by (Wiebe and Riloff, 2005), and polarity (positive, negative and neutral). The prior polarity is switched from positive to negative or vice-versa when a negative expression (as, e.g., "don't", "never") precedes the word. We obtained the prior subjectivity and polarity information from subjectivity lexicon of about 8,000 words used in (Riloff and Wiebe, 2003)[2]. Although this is a very comprehensive list, slang and specific Web vocabulary are not present on it, e.g., words as "yummy" or "ftw". For this reason, we collected popular words used on online discussions from many online sources and added them to this list.

Tweet Syntax Features. We exploited the syntax of the tweets to compose our features. They are: retweet; hashtag; reply; link, if the tweet contains a link; punctuation (exclamation and questions marks); emoticons (textual expression representing facial expressions); and upper cases (the number of words that starts with upper case in the tweet).

The frequency of each feature in a tweet is divided by the number of the words in the tweet.

3.2 Subjectivity Classifier

As we mentioned before, the first step in our tweet sentiment detection is to predict the subjectivity of a given tweet. We decided to create a single classifier by combining the objectivity sentences from Twendz and Twitter Sentiment (objectivity class) and the subjectivity sentences from all 3 sources. As we do not know the quality of the labels provided by these sources, we perform a cleaning process over this data to assure some reasonable quality. These are the steps:

1. Disagreement removal: we remove the tweets that are disagreed between the data sources in terms of subjectivity;

2. Same user's messages: we observed that the users with the highest number of messages in our dataset are usually those ones that post some objective messages, for example, advertising some product or posting some job recruiting information. For this reason, we allowed in the training data only one message from the same user. As we show later, this boosts the classification performance, mainly because it removes tweets labeled as subjective by the data sources but are in fact objective;

3. Top opinion words: to clean the objective training set, we remove from this set tweets that contain the top-n opinion words in the subjectivity training set, e.g., words as cool, suck, awesome etc.

As we show in Section 4, this process is in fact able to remove certain noisy in the training data, leading to a better performing subjectivity classifier.

To illustrate which of the proposed features are more effective for this task, the top-5 features in terms of information gain, based on our training data, are: positive polarity, link, strong subjective, upper case and verbs. Three of them are meta-information (positive polarity, strong subjective and verbs) and the other two are tweet syntax features (link and upper case). Here is a typical example of a objective tweet in which the user pointed an external link and used many upper case words: "Starbucks Expands Pay-By-IPhone Pilot to 1,000 Stores—Starbucks customers with Apple iPhones or iPod touches can .. http://oohja.com/x9UbC".

[2]The subjectivity lexicon is available at http://www.cs.pitt.edu/mpqa/

3.3 Polarity Classifier

The second step of our sentiment detection approach is polarity classification, i.e., predicting positive or negative sentiment on subjective tweets. In this section, first we analyze the quality of the polarity labels provided by the three sources, and whether their combination has the potential to bring improvement. Second, we present some modifications in the proposed features that are more suitable for this task.

3.3.1 Analysis of the Data Sources

The 3 data sources used in this work provide some kind of polarity labels (see Table 1). Two questions we investigate regarding these sources are: (1) how useful are these polarity labels? and (2) does combining them bring improvement in accuracy?

We take the following aspects into consideration:

- Labeler quality: if the labelers have low quality, combine them might not bring much improvement (Sheng et al., 2008). In our case, each source is treated as a labeler;
- Number of labels provided by the labelers: if the labels are informative, i.e., the probability of them being correct is higher than 0.5, the more the number of labels, the higher is the performance of a classifier built from them (Sheng et al., 2008);
- Labeler bias: the labeled data provided by the labelers might be only a subset of the real data distribution. For instance, labelers might be interested in only providing labels that they are more confident about;
- Different labeler bias: if labelers make similar mistakes, the combination of them might not bring much improvement.

We provide an empirical analysis of these datasets to address these points. First, we measure the polarity detection quality of a source by calculating the probability p of a label from this source being correct. We use the data manually labeled for assessing the classifiers' performance (testing data, see Section 4) to obtain the correct labels of a data sample. Table 2 shows their values. We can conclude from these numbers that the 3 sources provide a reasonable quality data. This means that combining them might bring some improvement to the polarity detection instead of, for instance, using one of them in isolation. An aspect that is overlooked by quality is the bias of the data. For instance, by examining the data from TwitterFeel, we found out that only 4 positive words ("awesome","rock","love" and "beat") cover 95% of their positive examples and only 6 negative words ("hate","suck","wtf","piss","stupid" and "fail") cover 96% of their negative set. Clearly, the data provided by this source is biased towards these words. This is probably the reason why this website outputs such fewer number of tweets compared to the other websites (see Table 1) as well as why its data has the smallest entropy among the sources (see Table 2).

Data sources	Quality	Entropy
Twendz	0.77	8.3
TwitterSentiment	0.82	7.9
TweetFeel	0.89	7.5

Table 2: Quality of the labels and entropy of the tweets provided by each data source for the polarity detection.

The quality of the data and its individual bias have certainly impact in the combination of labels. However, there is other important aspect that one needs to consider: different bias between the labelers. For instance, if labelers a and b make similar decisions, we expect that combining their labels would not bring much improvement. Therefore, the diversity of labelers is a key element in combining them (Polikar, 2006). One way to measure this is by calculating the agreement between the labels produced by the labelers. We use the kappa coefficient (Cohen, 1960) to measure the degree of agreement between two sources. Table 3 presents the coefficients for each par of data source. All the coefficients are between 0.4 and 0.6, which represents a moderate agreement between the labelers (Landis and Koch, 1977). This means that in fact the sources provide different bias regarding polarity detection.

Data sources	Kappa
Twendz/TwitterSentiment	0.58
TwitterSentiment/TweetFeel	0.58
Twendz/TweetFeel	0.44

Table 3: Kappa coefficient between pairs of sources.

From this analysis we can conclude that combining the labels provided by the 3 sources can improve the performance of the polarity detection instead of using one of them in isolation because they provide diverse labels (moderate kappa agreement) of reasonable quality, although there is some issues related to bias of the labels provided by them. In our experimental evaluation in Section 4, we present results obtained by different strategies of combining these sources that confirm these findings.

3.3.2 Polarity Features

The features used in the polarity detection are the same ones used in the subjectivity detection. However, as one would expect the set of the most discriminative features is different between the two tasks. For subjectivity detection, the top-5 features in terms of information gain, based on the training data, are: negative polarity, positive polarity, verbs, good emoticons and upper case. For this task, the meta-information of the words (negative polarity, positive polarity and verbs) is more important than specific features from Twitter (good emoticons and upper case), whereas for the subjectivity detection, tweet syntax features have a higher relevance.

This analysis show that prior polarity is very important for this task. However, one limitation of using it from a generic list is its values might not hold for some specific scenario. For instance, the polarity of the word "spot" is positive according to this list. However, looking at our training data almost half of the occurrences of this word appears in the positive set and the other half in the negative set. Thus, it is not correct to assume that prior polarity of "spot" is 1 for this particular data. This example illustrates our strategy to weight the prior polarities: for each word w with prior polarity defined by the list, we calculate the prior polarity of w, $pol(w)$, based on the distribution of w in the positive and negative sets. Thus, $pol_{pos}(w) = count(w, pos)/count(w)$ and $pol_{neg}(w) = 1 - pol_{pos}(w)$. We assume the polarity of a word is associated with the polarity of the sentence, which seems to be reasonable since we are dealing with very short messages. Although simple, this strategy is able to improve the polarity detection, as we show in Section 4.

4 Experiments

We have performed an extensive performance evaluation of our solution for twitter sentiment detection. Besides analyzing its overall performance, our goals included: examining different strategies to combine the labels provided by the sources; comparing our approach to previous ones in this area; and evaluating how robust our solution is to the noisy and biased data described in Section 3.

4.1 Experimental Setup

Data Sets. For the subjectivity detection, after the cleansing processing (see Section 3), the training data contains about 200,000 tweets (roughly 100,000 tweets were labeled by the sources as subjective ones and 100,000 objective ones), and for polarity detection, 71046 positive and 79628 negative tweets. For test data, we manually labeled 1,000 tweets as positive, negative and neutral. We also built a development set (1,000 tweets) to tune the parameters of the classification algorithms.

Approaches. For both tasks, subjectivity and polarity detection, we compared our approach with previous ones reported in the literature. Detailed explanation about them are as follows:

- ReviewSA: this is the approach proposed by Pang and Lee (Pang and Lee, 2004) for sentiment analysis in regular online reviews. It performs the subjectivity detection on a sentence-level relying on the proximity between sentences to detect subjectivity. The set of sentences predicted as subjective is then classified as negative or positive in terms of polarity using the unigrams that

compose the sentences. We used the implementation provided by LingPipe (LingPipe, 2008);

- Unigrams: Pang et al. (Pang et al., 2002) showed unigrams are effective for sentiment detection in regular reviews. Based on that, we built unigram-based classifiers for the subjectivity and polarity detections over the training data. Another approach that uses unigrams is the one used by TwitterSentiment website. For polarity detection, they select the positive examples for the training data from the tweets containing good emoticons and negative examples from tweets containing bad emoticons. (Go et al., 2009). We built a polarity classifier using this approach (Unigrams-TS).

- TwitterSA: TwitterSA exploits the features described in Section 3 in this paper. For the subjectivity detection, we trained a classifier from the two available sources, using the cleaning process described in Section 3 to remove noise in the training data, TwitterSA(cleaning), and other classifier trained from the original data, TwitterSA(no-cleaning). For the polarity detection task, we built a few classifiers to compare their performances: TwitterSA(single) and TwitterSA(weights) are two classifiers we trained using combined data from the 3 sources. The only difference is TwitterSA(weights) uses the modification of weighting the prior polarity of the words based on the training data. TwitterSA(voting) and TwitterSA(maxconf) combine classification outputs from 3 classifiers respectively trained from each source. TwitterSA(voting) uses majority voting to combine them and TwitterSA(maxconf) picks the one with maximum confidence score.

We use Weka (Witten and Frank, 2005) to create the classifiers. We tried different learning algorithms available on Weka and SVM obtained the best results for Unigrams and TwitterSA. Experimental results reported in this section are obtained using SVM.

4.2 Subjectivity Detection Evaluation

Table 4 shows the error rates obtained by the different subjectivity detection approaches. TwitterSA achieved lower error rate than both Unigrams and ReviewSA. As a result, these numbers confirm that features inferred from meta-information of words and specific syntax features from tweets are better indicators of the subjectivity than unigrams. Another advantage of our approach is since it uses only 20 features, the training and test times are much faster than using thousands of features like Unigrams. One of the reasons why TwitterSA obtained such a good performance was the process of data cleansing (see Section 3). The label quality provided by the sources for this task was very poor: 0.66 for Twendz and 0.68 for TwitterSentiment. By cleaning the data, the error decreased from 19.9, TwitterSA(no-cleaning), to 18.1, TwitterSA(cleaning). Regarding ReviewSA, its lower performance is expected since tweets are composed by single sentences and ReviewSA relies on the proximity between sentences to perform subjectivity detection.

We also investigated the influence of the size of training data on classification performance. Figure 2 plots the error rates obtained by TwitterSA and Unigrams versus the number of training examples. The curve corresponding to TwitterSA showed that it achieved good performances even with a small training data set, and kept almost constant as more examples were added to the training data, whereas for Unigrams the error rate decreased. For instance, with only 2,000 tweets as training data, TwitterSA obtained 20% of error rate whereas Unigrams 34.5%. These numbers show that our generic representation of tweets produces models that are able to generalize even with a few examples.

4.3 Polarity Detection Evaluation

We provide the results for polarity detection in Table 5. The best performance was obtained by TwitterSA(maxconf), which combines results of the 3 classifiers, respectively trained from each source, by taking the output by the most confident classifier, as the final prediction. TwitterSA(maxconf) was followed by TwitterSA(weights) and TwitterSA(single), both cre-

ated from a single training data. This result shows that computing the prior polarity of the words based on the training data TwitterSA(weights) brings some improvement for this task. TwitterSA(voting) obtained the highest error rate among the TwitterSA approaches. This implies that, in our scenario, the best way of combining the merits of the individual classifiers is by using a confidence score approach.

Unigrams also achieved comparable performances. However, when reducing the size of the training data, the performance gap between TwitterSA and Unigrams is much wider. Figure 3 shows the error rate of both approaches[3] in function of the training size. Similar to subjectivity detection, the training size does not have much influence in the error rate for TwitterSA. However for Unigrams, it decreased significantly as the training size increased. For instance, for a training size with 2,000 tweets, the error rate for Unigrams was 46% versus 23.8% for our approach. As for subjectivity detection, this occurs because our features are in fact able to capture a more general representation of the tweets.

Another advantage of TwitterSA over Unigrams is that it produces more robust models. To illustrate this, we present the error rates of Unigrams and TwitterSA where the training data is composed by data from each source in isolation. For the TweetFeel website, where data is very biased (see Section 3), Unigrams obtained an error rate of 44.5% whereas over a sample of the same size of the combined training data (Figure 3), it obtained an error rate of around 30%. Our approach also performed worse over this data than the general one, but still had a reasonable error rate, 25.1%. Regarding the Twendz website, which is the noisiest one (Section 3), Unigrams also obtained a poor performance comparing it against its performance over a sample of the general data with a same size (see Table 5 and Figure 3). Our approach, on the other hand, was not much influenced by the noise (22.9% on noisy data and around 20% on the sample of same size of the general data). Finally, since the data quality provided by TwitterSentiment is better than the previous sources (Table 2), there was not much impact over both classifiers created from it.

From this analysis over real data, we can conclude that our approach produces (1) an effective polarity classifier even when only a small number of training data is available; (2) a robust model to bias and noise in the training data; and (3) combining data sources with such distinct characteristics, as our data analysis in Section 3 pointed out, is effective.

[3]For this experiment, we used the TwitterSA(single) configuration.

Approach	Error rate
TwitterSA(cleaning)	18.1
TwitterSA(no-cleaning)	19.9
Unigrams	27.6
ReviewSA	32

Table 4: Results for subjectivity detection.

Approach	Error rate
TwitterSA(maxconf)	18.7
TwitterSA(weights)	19.4
TwitterSA(single)	20
TwitterSA(voting)	22.6
Unigrams	20.9
ReviewSA	21.7
Unigrams-TS	24.3

Table 5: Results for polarity detection.

Site	Training Size	TwitterSA	Unigrams
TweetFeel	13120	25.1	44.5
Twendz	78025	22.9	32.3
TwitterSentiment	59578	22	23.4

Table 6: Training data size for each source and error rates obtained by classifiers built from them.

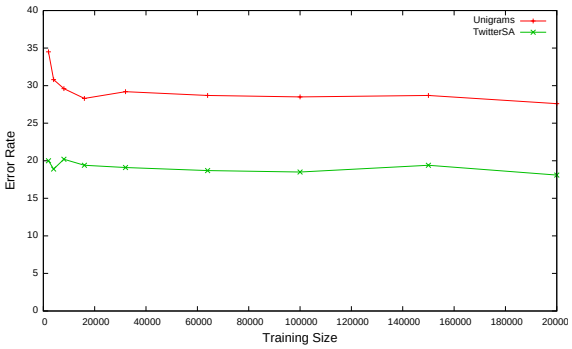


Figure 2: Influence of the training data size in the error rate of subjectivity detection using Unigrams and TwitterSA.

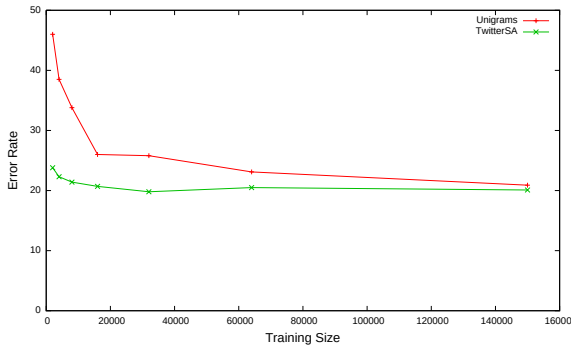


Figure 3: Influence of the training data size in the error rate of polarity detection using Unigrams and TwitterSA.

5 Related Work

There is a rich literature in the area of sentiment detection (see e.g., (Pang et al., 2002; Pang and Lee, 2004; Wiebe and Riloff, 2005; Go et al., 2009; Glance et al., 2005). Most of these approaches try to perform this task on large texts, as e.g., newspaper articles and movie reviews. Another common characteristic of some of them is the use of n-grams as features to create their models. For instance, Pang and Lee (Pang and Lee, 2004) explores the fact that sentences close in a text might share the same subjectivity to create a better subjectivity detector and, similar to (Pang et al., 2002), uses unigrams as features for the polarity detection. However, these approaches do not obtain a good performance on detecting sentiment on tweets, as we showed in Section 4, mainly because tweets are very short messages. In addition to that, since they use a raw word representation, they are more sensible to bias and noise, and need a much higher number of examples in the training data than our approach to obtain a reasonable performance.

The Web sources used in this paper and some other websites provide sentiment detection for tweets. A great limitation to evaluate them is they do not make available how their classification was built. One exception is TwitterSentiment (Go et al., 2009), for instance, which considers tweets with good emoticons as positive examples and tweets with bad emoticons as negative examples for the training data, and builds a classifier using unigrams and bigrams as features. We showed in Section 4 that our approach works better than theirs for this problem, obtaining lower error rates.

6 Conclusions and Future Work

We have presented an effective and robust sentiment detection approach for Twitter messages, which uses biased and noisy labels as input to build its models. This performance is due to the fact that: (1) our approach creates a more abstract representation of these messages, instead of using a raw word representation of them as some previous approaches; and (2) although noisy and biased, the data sources provide labels of reasonable quality and, since they have different bias, combining them also brought some benefits.

The main limitation of our approach is the cases of sentences that contain antagonistic sentiments. As future work, we want to perform a more fine grained analysis of sentences in order to identify its main focus and then based the sentiment classification on it.

References

Cohen, J. 1960. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37.

Glance, N., M. Hurst, K. Nigam, M. Siegler, R. Stockton, and T. Tomokiyo. 2005. Deriving marketing intelligence from online discussion. In *Proceedings of the eleventh ACM SIGKDD*, pages 419–428. ACM.

Go, A., R. Bhayani, and L. Huang. 2009. Twitter sentiment classification using distant supervision. Technical report, Stanford Digital Library Technologies Project.

Landis, J.R. and G.G. Koch. 1977. The measurement of observer agreement for categorical data. *Biometrics*, pages 159–174.

LingPipe. 2008. LingPipe 3.9.1. `http://alias-i.com/lingpipe`.

Pang, B. and L. Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the ACL*, volume 2004.

Pang, B., L. Lee, and S. Vaithyanathan. 2002. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL*, pages 79–86. Association for Computational Linguistics.

Polikar, R. 2006. Ensemble based systems in decision making. *IEEE Circuits and systems magazine*, 6(3):21–45.

Prentice, S. and E. Huffman. 2008. Social Medias New Role In Emergency Management. *Idaho National Laboratory*, pages 1–5.

Riloff, E. and J. Wiebe. 2003. Learning extraction patterns for subjective expressions. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, pages 105–112.

Riloff, E., J. Wiebe, and T. Wilson. 2003. Learning subjective nouns using extraction pattern bootstrapping. In *Proceedings of the 7th Conference on Natural Language Learning*, pages 25–32.

Riloff, E., S. Patwardhan, and J. Wiebe. 2006. Feature subsumption for opinion analysis. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 440–448. Association for Computational Linguistics.

Sheng, V.S., F. Provost, and P.G. Ipeirotis. 2008. Get another label? Improving data quality and data mining using multiple, noisy labelers. In *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 614–622. ACM.

Wiebe, J. and E. Riloff. 2005. Creating subjective and objective sentence classifiers from unannotated texts. *Computational Linguistics and Intelligent Text Processing*, pages 486–497.

Wiebe, J.M., RF Brace, and T.P. O'Hara. 1999. Development and use of a gold-standard data set for subjectivity classifications. In *Proceedings of the ACL*, pages 246–253. Association for Computational Linguistics.

Wilson, T., J. Wiebe, and P. Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *EMNLP*, page 354. Association for Computational Linguistics.

Witten, Ian H. and Eibe Frank. 2005. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann.

Benchmarking for syntax-based sentential inference

Paul Bedaride
INRIA/LORIA
Université Henri Poincaré
paul.bedaride@loria.fr

Claire Gardent
CNRS/LORIA
claire.gardent@loria.fr

Abstract

We propose a methodology for investigating how well NLP systems handle meaning preserving syntactic variations. We start by presenting a method for the semi automated creation of a benchmark where entailment is mediated solely by meaning preserving syntactic variations. We then use this benchmark to compare a semantic role labeller and two grammar based RTE systems. We argue that the proposed methodology (i) supports a modular evaluation of the ability of NLP systems to handle the syntax/semantic interface and (ii) permits focused error mining and error analysis.

1 Introduction

First launched in 2005, the Recognising Textual Inference Challenge (RTE)[1] aims to assess in how far computer systems can emulate a human being in determining whether a short text fragment H referred to as the hypothesis, follows from or is contradicted by a text fragment T. In the RTE benchmarks, the hypothesis is a short constructed sentence whilst the text fragments are short passages of naturally occurring texts. As a result, the RTE challenge permits evaluating the capacity of NLP systems to handle local textual inference on real data, an enabling technology for any applications involving document interpretation.

In this paper, we focus on entailments based on meaning entailing, syntactic transformations such as:

(1) The man gives the woman the flowers that smell nice $\Rightarrow$ The flowers which are given to the woman smell nice

[1] http://www.pascal-network.org/Challenges/RTE

We start (Section 2) by motivating the approach. We argue that the proposed evaluation methodology (i) interestingly complements the RTE challenge in that it permits a modular, analytic evaluation of the ability of NLP systems to handle syntax-based, sentential inference and (ii) permits focused error mining and analysis .

In Section 3, we go on to describe the benchmark construction process. Each item of the constructed benchmark associates two sentences with a truth value (true or false) indicating whether or not the second sentence can be understood to follow from the first. The construction of these benchmark items relies on the use of a grammar based surface realiser and we show how this permits automatically associating with each inference item, an entailment value (true or false) and a detailed syntactic annotation reflecting the syntactic constructs present in the two sentences constituting each benchmark item.

In section 4, we use the benchmark to evaluate and compare three systems designed to recognise meaning preserving syntactic variations namely, a semantic role labeller, Johan Bos' Nutcracker RTE system (where the syntax/semantic interface is handled by a semantic construction module working on the output of combinatory categorial grammar parser) and the Afazio system, a hybrid system combining statistical parsing, symbolic semantic role labelling and sentential entailment detection using first order logic. We give the evaluation figures for each system. Additionally, we show how the detailed syntactic annotations automatically associated with each benchmark item by the surface realiser can be used to identify the most likely source of errors that is, the syntactic constructs that most frequently co-occur with en entailment recognition error.

Coling 2008: Poster Volume, pages 45–53,
Beijing, August 2010

2 Motivations

Arguably focusing on meaning entailing syntactic transformations is very weak. Indeed, one of the key conclusions at the second RTE Challenge Workshop was that entailment modeling requires vast knowledge resources that correspond to different types of entailment reasoning e.g., ontological and lexical relationships, paraphrases and entailment rules, meaning entailing syntactic transformations and last but not least, world knowledge. Further, Manning (2006) has strongly argued against circumscribing the RTE data to certain forms of inference such as for instance, inferences based solely on linguistic knowledge. Finally, it is also often insisted that naturally occurring data should be favored over constructed data.

While we agree that challenges such as the RTE challenge are useful in testing systems abilities to cope with real data, we believe there is also room for more focused evaluation setups.

Focusing on syntax based entailments. As mentioned above, syntax based entailment is only one of the many inference types involved in determining textual entailment. Nevertheless, a manual analysis of the RTE1 data by (Vanderwende et al., 2005) indicates that 37% of the examples could be handled by considering syntax alone. Similarly, (Garoufi, 2007) shows that 37.5% of the RTE2 data does not involve deep reasoning and more specifically, that 33.8% of the RTE2 data involves syntactic or lexical knowledge only. Hence although the holistic, blackbox type of evaluation practiced in the RTE challenge is undeniably useful in assessing the ability of existing systems to handle local textual inference, a more analytic, modular kind of evaluation targeting syntax-based entailment reasoning is arguably also of interest.

Another interesting feature of the SSI (syntax-based sentential entailment) task we propose is that it provides an alternative way of evaluating semantic role labelling (SRL) systems. Typically, the evaluation of SRL systems relies on a hand annotated corpus such as PropBank or the FrameNet corpus. The systems precision and recall are then computed w.r.t. this reference corpus. As has been repeatedly argued (Moll and Hutchinson, 2003; Galliers and Jones, 1993), intrinsic evaluations may be of very limited value. For semantically oriented tools such as SRL systems, it is important to also assess their results w.r.t. the task which they are meant support namely reasoning : Do the semantic representations built by SRL help in making the correct inferences ? Can they be used, for instance, to determine whether a given sentence answers a given question ? or whether the content of one sentence follow from that another ? As explained in (Giampiccolo et al., 2007), entailment recognition is a first, major step towards answering these questions. Accordingly, instead of comparing the representations produced by SRL systems against a gold standard, the evaluation scheme presented here, permits evaluating them w.r.t. their ability to capture syntax based sentential inference.

It is worth adding that, although the present paper focuses on entailments strictly based on syntax, the proposed methodology should straightforwardly extend to further types of entailment such as in particular, entailments involving lexical relations (synonymy, antonymy, etc.) or entailments involving more complex semantic phenomena such as the interplay between different classes of complement taking verbs, polarity and author commitment discussed in (Nairn et al., 2006). This is because as we shall see in section 3, our approach is based on an extensive, hand written grammar of English integrating syntax and semantics. By modifying the grammar, the lexicon and/or the semantics, data of varying linguistic type and complexity can be produced and used for evaluation.

Hand constructed vs. naturally occurring data. Although in the 90s, hand tailored testsuites such as (Lehmann et al., 1996; Cooper et al., 1995) were deemed useful for evaluating NLP systems, it is today generally assumed that, for evaluation purposes, naturally occurring data is best. We argue that constructed data can interestingly complement naturally occurring data.

To start with, we agree with (Crouch et al., 2006; Cohen et al., 2008) that science generally benefits from combining laboratory and field studies and more specifically, that computational linguistics can benefit from evaluating systems on

a combination of naturally occurring and constructed data.

Moreover, constructed data need not be hand constructed. Interestingly, automating the production of this data can help provide better data annotation as well as better and better balanced data coverage than both hand constructed data and naturally occurring data. Indeed, as we shall show in section 4, the benchmark creation process presented here supports a detailed and fully automated annotation of the syntactic properties associated with each benchmark item. As shown in section 5, this in turn allows for detailed error mining making it possible to identify the most likely causes of system errors. Additionally, the proposed methodology permits controlling over such benchmark parameters as the size of the data set, the balance between true and false entailments, the correlation between word overlap and entailment value and/or the specific syntactic phenomena involved. This is in contrast with the RTE data collection process where "the distribution of examples is arbitrary to a large extent, being determined by manual selection[2]" (Giampiccolo et al., 2007). As has been repeatedly pointed out (Burchardt et al., 2007; Garoufi, 2007), the RTE datasets are poorly balanced w.r.t., both the frequency and the coverage of the various phenomena interacting with textual inference.

3 Benchmark

We now present the content of an SSI benchmark and the method for constructing it.

An SSI benchmark item (cf. e.g., Figure 1) consists of two sentences and a truth value (true or false) indicating whether or not the second sentence can be understood to follow from the first. In addition, each sentence is associated with a detailed syntactic annotation describing the syntactic constructs present in the sentence.

The benchmark construction process consists of two main steps. First, a generation bank is built. Second, this generation bank is drawn upon

T: The man gives the woman the flowers that smell nice
smell:{n0Va1,active,relSubj,canAdj}
give:{n0Vn2n1,active,canSubj,canObj,canIObj}
H: The flowers are given to the woman
give:{n0Vn1Pn2,shortPassive,canSubj,canIObj}
Entailment: TRUE

Figure 1: An SSI Benchmark item

to construct a balanced data set for SSI evaluation. We now describe each of these processes in turn.

Constructing a generation bank We use the term "generation bank" to refer to a dataset whose items are produced by a surface realiser i.e., a sentence generator. A surface realiser in turn is a program which associates with a given semantic representation, the set of sentences verbalising the meaning encoded by that representation. To construct our generation bank, we use the GenI surface realiser (Gardent and Kow, 2007). This realiser uses a Feature based Tree Adjoining Grammar (FTAG) augmented with a unification sematics as proposed in (Gardent and Kallmeyer, 2003) to produce all the sentences associated by the grammar with a given semantic representation. Interestingly, the FTAG used has been compiled out of a factorised representation and as a result, each elementary grammar unit (i.e., elementary FTAG tree) and further each parse tree, is associated with a list of items indicating the syntactic construct(s) captured by that unit/tree[3]. In short, GenI permits associating with a given semantics, a set of sentences and further for each of these sentences, a set of items indicating the syntactic construct(s) present in the syntactic tree of that sentence. For instance, the sentences and the syntactic constructs associated by GenI with the semantics given in (2) are those given in (3).

(2) A:give(B C D E) G:the(C) F:man(C) H:the(D) I:woman(D) J:the(E) K:flower(E) L:passive(B) L:smell(M E N) O:nice(N)

(3) a. The flower which smells nice is given to the woman by the man

[2]The short texts of the RTE benchmarks are automatically extracted from real texts using different applications (e.g., Q/A, summarisation, information extraction, information retrieval systems) but the query used to retrieve these texts is either constructed manually or post-edited.

[3]Space is lacking to give a detailed explanation of this process here. We refer the reader to (Gardent and Kow, 2007) for more details on how GenI associates with a given semantics, a set of sentences and for each sentence a set of items indicating the syntactic construct(s) present in the syntactic tree of that sentence.

give:n0Vn1Pn2-Passive-CanSubj-ToObj-ByAgt, smell:n0V-active-OvertSubjectRelative

b. The flower which smells nice is given the woman by the man
give:n0Vn2n1-Passive, smell:n0V-active-OvertSubjectRelative

c. The flower which is given the woman by the man smells nice
give:n0Vn2n1-Passive-CovertSubjectRelative, smell:n0V-active

d. The flower which is given to the woman by the man smells nice
give:n0Vn1Pn2-Passive-OvertSubjectRelative, smell:n0V-active

e. The flower that smells nice is given to the woman by the man
give:n0Vn1Pn2-Passive, smell:n0V-CovertSubjectRelative

f. The flower that smells nice is given the woman by the man
give:n0Vn2n1-Passive, smell:n0V-CovertSubjectRelative

g. The flower that is given the woman by the man smells nice
give:n0Vn2n1-Passive-CovertSubjectRelative, smell:n0V-active

h. The flower that is given to the woman by the man smells nice
give:n0Vn1Pn2-Passive-CovertSubjectRelative, smell:n0V-active

The tagset of syntactic annotation covers the subcategorisation type of the verb, a specification of the verb mood and a description of how arguments are realised.

The semantic representation language used is a simplified version of the flat semantics used in e.g., (Copestake et al., 2005) which is sufficient for the cases handled in the present paper. The grammar and therefore the generator, can however easily be modified to integrate the more sophisticated version proposed in (Gardent and Kallmeyer, 2003) and thereby provide an adequate treatment of scope.

Constructing an SSI benchmark. Given a generation bank, false and true sentential entailment pairs can be automatically produced by taking pairs of sentences $\langle S_1, S_2 \rangle$ and comparing their semantics: if the semantics of S_2 is entailed by the semantics of S_1, the pair is marked as TRUE else as FALSE. The syntactic annotations associated in the generation bank with each sentence are carried over to the SSI benchmark thereby ensuring that the overall information contained in each SSI benchmark is as illustrated in Figure 1 namely, two pairs of syntactically annotated sentences and a truth value indicating (non) entailment.

To determine whether a sentence textually entails another we translate their flat semantic representation into first order logic and check for logical entailment. Differences in semantic representations which are linked to functional surface differences such as active/passive or the presence/absence of a complementizer (*John sees Mary leaving/John sees that Mary leaves*) are dealt with by (automatically) removing the corresponding semantic literals from the semantic representation before translating it to first order logic. In other words, active/passive variants of the same sentence are deemed semantically equivalent.

Note that contrary to what is assumed in the RTE challenge, entailment is here logical rather than textual (i.e., determined by a human) entailment. By using logical, rather than textual (i.e., human based) entailment, it is possible that some cases of syntax mediated textual entailments are not taken into account. However, intuitively, it seems reasonable to assume that for most of the entailments mediated by syntax alone, logical and textual entailments coincide.

3.1 The SSI benchmark

Using the methodology just described, we first produced a generation bank of 226 items using 81 input formula distributed over 4 verb types. From this generation bank, a total of 6 396 SSI-pairs were built with a ratio of 42.6% true and 57.4% false entailments.

For our experiment, we extracted from this SSI-suite, 1000 pairs with an equal proportion of true and false entailments and a 7/23/30/40 distribution of four subcategorisation types namely, adjectival predicative (n0Va1 e.g., *The cake tastes good*), intransitive (n0V), transitive (n0Vn1) and ditransitive (n0Vn2n1)[4]. We furthermore con-

[4]The subcategorisation type of an SSI item is determined manually and refers either to the main verb if the sentence is

strained the suite to respect a neutral correlation between word overlap and entailment. Following (Garoufi, 2007), we define this correlation as follows. The word overlap $wo(T, H)$ between two sentences T and H is the ratio of common lemmas between T and H on the number of lemmas in H (non content words are ignored). If entailment holds, the word overlap/entailment correlation value of the sentence pair is $wo(T, H)$. Otherwise it is $1 - wo(T, H)$. The 1000 items of the SSI suite used in our experiment were chosen in such a way that the word overlap/entailment correlation value of the SSI suite is 0.49.

In sum, the SSI suite used for testing exhibits the following features. First, it is balanced w.r.t. entailment. Second, it displays good syntactic variability based both on the constrained distribution of the four subcategorisation types and on the use of the XTAG grammar to construct sentences from abstract representations (cf. the paraphrases in (3) generated by GenI from the representation given in (2)). Third, it contains 1000 items and could easily be extended to cover more and more varied data. Fourth, it is specifically tailored to check systems on their ability to deal with syntax based sentential entailment: word overlap is high, syntactic variability is provided and the correlation between word overlap and entailment is not biased.

4 System evaluation and comparison

SRL and grammar based systems equipped with a compositional semantics are primary targets for an SSI evaluation. Indeed these systems aim to abstract away from syntactic differences by producing semantic representations of a text which capture predicate/argument relations independent of their syntactic realisation.

We evaluated three such systems on the SSI benchmark namely, NutCracker, (Johansson and Nugues, 2008)'s Semantic Role Labeller and the Afazio RTE system.

4.1 Systems

Nutcracker Nutcracker is a system for recognising textual entailment which uses deep semantic processing and automated reasoning. Deep semantic processing associates each sentence with a Discourse Representation Structure (DRS (Kamp and Reyle, 1993)) by first, using a statistical parser to build the syntactic parse of the sentence and second, using a symbolic semantic construction module to associate a DRS with the syntactic parse. Entailment between two DRSs is then checked by translating this DRS into a first-order logical (FOL) formula and first trying to find a proof. If a proof is found then the entailment is set to true. Otherwise, Nutcracker backs off with a word overlap module computed over an abstract representation of the input sentences and taking into account WordNet related information. Nutcracker was entered in the first RTE challenge and scored an accuraccy (percentage of correct judgments) of 0.562 when used as is and 0.612 when combined with machine learning techniques. For our experiment, we use the online version of Nutcracker and the given default parameters.

Afazio Like Nutcracker, the Afazio system combines a statistical parser (the Stanford parser) with a symbolic semantic component. This component pipelines several rewrite modules which translate the parser output into a first order logic formula intended to abstract away from surface differences and assign syntactic paraphrases the same representation (Bedaride and Gardent, 2009). Special emphasis is placed on capturing syntax based equivalences such as syntactic (e.g., active/passive) variations, redistributions and noun/verb variants. Once the parser output has been normalised into predicate/argument representations capturing these equivalences, the resulting structures are rewritten into first order logic formulae. Like Nutcracker, Afazio checks entailment using first order automated reasoners namely, *Equinox* and *Paradox* [5].

SRL (Johansson and Nugues, 2008)'s semantic role labeller achieved the top score in the closed CoNLL 2008 challenge reaching a labeled semantic F1 of 81.65. To allow for comparison with Nutcracker and Afazio, we adapted the

a clause or to the embedded verb if the sentence is a complex sentence.

[5] http://www.cs.chalmers.se/~koen/folkung/

rewrite module used in Afazio to rewrite Predicate/Argument structures into FOL formula in such a way as to fit (Johansson and Nugues, 2008)'s SRL output. We then use FOL automated reasoner to check entailment.

4.2 Evaluation scheme and results

The results obtained by the three systems are summarised in Table 1. TP (true positives) is the number of entailments recognised as such by the system and TN (true negatives) of non entailments. Conversely, FN and FP indicate how often the systems get it wrong: FP is the number of non entailments labelled as entailments by the system and FN, the number of entailments labelled as non entailments. 'ERROR' refers to cases where the CCG parser used by Nutcracker fails to find a parse. The last three columns indicate the overall ability of the systems to recognise false entailments (TN/N with N the number of false entailment in the benchmark), true entailments (TP/P) and all true and false entailment (Precision).

Overall, Afazio outperforms both Nutcracker and the SRL system. This is unsurprising since contrary to these other two systems, Afazio was specifically designed to handle syntax based sentential entailment. Its strength is that it combines a full SRL system with a semantic construction module designed for entailment detection. More surprisingly, the CCG parser used by Nutcracker often fails to find a parse.

The SRL system has a high rate of false negatives. Using the error mining technique presented in the next section, we found that the most suspicious syntactic constructs all included a relativised argument. A closer look at the analyses showed that this was due to the fact that SRL systems fail to identify the antecedent of a relative pronoun, an identification that is necessary for entailment checking. Another important difference with Afazio is that the SRL system produces a single output. In contrast, Afazio checks entailment for any of the pairs of semantic representations derived from the first 9 parses of the Stanford parser. The number 9 was determined empirically and proved to yield the best results overall although as we shall see in the error mining section, taking such a high number of parses into account often leads to incorrect results when the hypothesis (H) is short.

Nutcracker, on the other hand, produces many false positives. This is in part due to cases where the time bound is reached and the word overlap backoff triggered. Since the overall word overlap of the SSI suite is high, the backoff often predicts an entailment where in fact there is none (for instance, the pair *'John gave flowers to Mary/Mary gave flowers to John* has a perfect word overlap but entailment does not hold). When removing the backoff results i.e., when assigning all backoff cases a negative entailment value, overall precision approximates 60%. In other words, on cases such as those present in the SSI benchmark where word overlap is generally high but the correlation between word overlap and entailment value is neutral, Nutcracker should be used without backoff.

5 Finding the source of errors

The annotations contained in the automatically constructed testsuite can help identify the most likely sources of failures. We use (Sagot and de La Clergerie, 2006)'s suspicion rate to compute the probability that a given pair of sets of syntactic tags is responsible for an RTE detection failure. The tag set pairs with highest suspicion rate indicate which syntactic phenomena often cooccurs with failure.

More specifically, we store for each testsuite item (T,H), all tag pairs (t_j, h_k) such that the syntactic tags t_j and h_k are associated with the same predicate P_i but t_j occurs in T and h_k in H. That is, we collect the tag pairs formed by taking the tags that label the occurrence of the same predicate on both sides of the implication. If a predicate occurs only in H then for each syntactic tag h_k labelling this predicate, the pair (nil, h_k) is created. Conversely, if a predicate occurs only in T, the pair (t_j, nil) is added. Furthermore, the tags describing the subcategorisation type and the form of the verb are grouped into a single tag so as to reduce the tagset and limit data sparseness. For instance, given the pair of sentences in Figure (1), the following tag pairs are produced:
(n0Va1:active:relSubj, nil)
(n0Va1:active:canAdj, nil)

system	ERROR	TN	FN	TP	FP	TN/N	TP/P	Prec
afazio	0	360	147	353	140	0.7200	0.7060	71.3%
nutcracker	155	22	62	312	449	0.0467	0.8342	39.5% (60% w/o B.O.)
srl	0	487	437	63	13	0.9740	0.1260	55.0%

Table 1: Results of the three systems on the SSI-testsuite (TN = true negatives, FN = false negatives, TP = true positives, FP = false positives, N = TN + FP, P = TP + FN, Prec = Precision, ERROR: no parse tree found)

(n0Vn2n1:active:canSubj,n0Vn1Pn2:shortPassive:canSubj)
(n0Vn2n1:active:canSubj,n0Vn1Pn2:shortPassive:canIObj)
(n0Vn2n1:active:canObj,n0Vn1Pn2:shortPassive:canSubj)
(n0Vn2n1:active:canObj,n0Vn1Pn2:shortPassive:canIObj)
(n0Vn2n1:active:canIObj,n0Vn1Pn2:shortPassive:canSubj)
(n0Vn2n1:active:canIObj,n0Vn1Pn2:shortPassive:canIObj)

For each tag pair, we then compute the suspicion rate of that pair using (Sagot and de La Clergerie, 2006)'s fix point algorithm. To also take into account pairs of sets of tags (rather than just pairs of single tags), we furthermore preprocess the data according to (de Kok et al., 2009)'s proposal for handling n-grams.

Computing the suspicion rate of a tag pair. The error mining's suspicion rate algorithm of (Sagot and de La Clergerie, 2006) is a fix point algorithm used to detect the possible cause of parsing failures. We apply this algorithm to the pair of annotated sentences resulting from running the three systems on the automatically created testsuite as follows. Each such pair consists of a pair of sentences, a set of tag pairs, an entailment value (true or false) and a result value namely FP (false positive), FN (false negative), TP (true positive) or TN (true negative). To search for the most likely causes of failure, we consider separately entailments from non entailments. If entailment holds, the suspicion rate of a sentence pair is 0 for true positive and 1 for false positives. Conversely, if entailment does not hold, the suspicion rate of the sentence pair is 0 for true negatives and 1 for false negatives.

The aim is to detect the tag pair most likely to make entailment detection fail[6]. The algorithm iterates between tag pair occurrences and tag pair forms, redistributing probabilities with each iteration as follows. Initially, all tag pair occurrences in a given sentence have the same suspicion rate namely, the suspicion rate of the sentence (1 if the entailment could not be recognised, 0 otherwise) divided by the number of tag pair occurrences in that sentence. Next, the suspicion rate of a tag pair form is defined as the average suspicion rate of all occurrences of that tag pair. The suspicion rate of a tag pair occurrence within each particular sentence is then recalculated as the suspicion rate of that tag pair form normalised by the suspicion rates of the other tag pair forms occurring within the same sentence. The iteration stops when the process reaches a fixed point where the suspicion rates have stabilised.

Extending the approach to pairs of tag sets. To account for entailment recognition due to more than one tag pair, we follow (de Kok et al., 2009) and introduce a preprocessing step which first, expands tag pair unigrams to tag pair n-grams when there is evidence that it is useful that is, when an n-gram has a higher suspicion rate than each of its sub n-grams. For this preprocessing, the suspicion of a tag pair t is defined as the ratio of t occurrences in unrecognised entailments and the total number of t occurrences in the corpus. To compensate for data sparseness, an additional expansion factor is used which depends on the frequency of an n-gram and approaches one for higher frequency. In this way, long n-grams that have low frequency are not favoured. The longer the n-gram is, the more frequent or the more suspicious it needs to be in order to be selected by the preprocessing step.

We apply this extension to the SSI setting. We first extend the set of available tag pairs with tag set pairs such that the suspicion rate of these pairs is higher that the suspicion rate of each of the smaller tagset pairs that can be constructed from these sets. We then apply (Sagot and de La Clerg-

[6]We make the simplifying hypothesis that for each entailment not recognised, a single tag pair or tag pair n-gram is the cause of the failure.

n0Vs1:act:CanSubj	nil	0.85
n0Vn1:act:CanObj	nil	0.46
n0V:betaVn	nil	0.28

Table 2: The first 3 suspects for false positives

n0V:act	n0V:act:RelCSubj	0.73
n0Vs1:act:CanSubj n0V:act:RelOSubj	n0Vs1:act:CanSubj n0V:betaVn	0.69
n0Vs1:act:CanSub n0V:act:CanSubj	n0Vs1:act:CanSubj n0V:betaVn	0.69

Table 3: The first 3 suspects for false negatives

erie, 2006)'s fix point algorithm to compute the suspicion rate of the resulting tag pairs and tag sets pairs.

Results and discussion. We now show how error mining can help shed some light on the most probable sources of error when using Afazio.

For false positives (non entailment labelled as entailment by Afazio), the 3 most suspect tag pairs are given in Table 2. The first pair (n0Vs1:act:CanSubj,nil) points out to cases such as *Bill sees the woman give the flower to the man / The man gives the flower to the woman.* where T contains a verb with a sentential argument not present in H. In such cases, we found that the sentential argument in T is usually incorrectly analysed, the analyses produced are fragmented and entailment goes through. Similarly, the second suspect (n0Vn1:act:CanObj,nil) points to cases such as *a man sees Lisa dancing / a man dances*, where the transitive verb in T has no counterpart in H. Here the high number of analyses relied on by Afazio together with the small size of H leads to entailment detection: because we consider many possible analyses for T and H and because H is very short, one pair of analyses is found to match. Finally, the third suspect (n0V:betaVn,nil) points to cases such as *Bill insists for the singing man to dance / Bill dances* where the gerund is wrongly analysed and a relation is incorrectly established by the parser between *Bill* and *dance* (in H).

For false negatives, the first suspect indicates incorrect analyses for cases where an intransitive with canonical subject in H is matched by an intransitive with covert relative subject (e.g., *Bill sees the woman give the flower to the man / the man gives the flower to the woman.*). The second suspect points to cases such as *Bill insists for the man who sings to dance / Bill insists that the singing man dances.* where an embedded verb with relative overt subject in T (sings) is matched in H by an embedded gerund. Similarly, the third suspect points to embedded verbs with canonical subject matched by gerund verbs as in *the man who Bill insists that dances sings / Bill insists that the singing man dances*.

6 Conclusion

The development of a linguistically principled treatment of the RTE task requires a clear understanding of the strength and weaknesses of RTE systems w.r.t. to the various types of reasoning involved. The main contribution of this paper is the specification of an evaluation methodology which permits a focused evaluation of syntax based reasoning on arbitrarily many inputs. As the results show, there is room for improvment even on that most basic level. In future work, we plan to extend the approach to other types of inferences required for textual entailment recognition. A more sophisticated compositional semantics in the grammar used by the sentence generator would allow for entailments involving more complex semantic phenomena such as the interplay between implicative verbs, polarity and downward/upward monotonicity discussed in (Nairn et al., 2006). For instance, it would allow for sentence pairs such as *Ed did not forget to force Dave to leave / Dave left* to be assigned the correct entailment value.

References

Bedaride, P. and C. Gardent. 2009. Noun/verb entailment. In *4th Language and Technology Conference*, Poznan, Poland.

Burchardt, A., N. Reiter, S. Thater, and A. Frank. 2007. A semantic approach to textual entailment: System evaluation and task analysis. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 10–16.

Cohen, K., W. Baumgartner, and L. Hunter. 2008. Software testing and the naturally occurring data assumption in natural language processing. In *Proc. of "Software engineering, testing, and quality assurance for natural language processing ACL Workshop"*.

Cooper, R., R. Crouch, J. van Eijck, C. Fox, J. van Genabith, J. Jaspars, H. Kamp, M. Pinkal, D. Milward, M. Poesio, and S. Pulman. 1995. A framework for computational semantics, FraCaS. Technical report. MS. Stanford University.

Copestake, A., D. Flickinger, C. Pollard, and I. A. Sag. 2005. Minimal recursion semantics: an introduction. *Research on Language and Computation*, 3.4:281–332.

Crouch, R., L. Karttunen, and A. Zaenen. 2006. Circumscribing is not excluding: A reply to manning. MS. Palo Alto Research Center.

de Kok, D., J. Ma, and G. van Noord. 2009. A generalized method for iterative error mining in parsing results. In *Proceedings of the 2009 Workshop on Grammar Engineering Across Frameworks (GEAF 2009)*, pages 71–79, Suntec, Singapore, August. Association for Computational Linguistics.

Galliers, J. R. and K. Sparck Jones. 1993. Evaluating natural language processing systems. Technical report, Computer Laboratory, University of Cambridge. Technical Report 291.

Gardent, C. and L. Kallmeyer. 2003. Semantic construction in ftag. In *Proceedings of the 10th meeting of the European Chapter of the Association for Computational Linguistics*, Budapest, Hungary.

Gardent, C. and E. Kow. 2007. A symbolic approach to near-deterministic surface realisation using tree adjoining grammar. In *ACL07*.

Garoufi, K. 2007. Towards a better understanding of applied textual entailment: Annotation and evaluation of the rte-2 dataset. Master's thesis, Saarland University, Saarbrcken.

Giampiccolo, D., B. Magnini, I. Dagan, and B. Dolan. 2007. The third pascal recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 1–9.

Johansson, R. and P. Nugues. 2008. Dependency-based syntactic-semantic analysis with propbank and nombank. In *CoNLL '08: Proceedings of the Twelfth Conference on Computational Natural Language Learning*, pages 183–187, Morristown, NJ, USA. Association for Computational Linguistics.

Kamp, H. and U. Reyle. 1993. *From Discourse to Logic. Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory*. Kluwer.

Lehmann, S., S. Oepen, H. Baur, O. Lbdkan, and D. Arnold. 1996. tsnlp — test suites for natural language processing. In *In J. Nerbonne (Ed.), Linguistic Databases*. CSLI Publications.

Manning, C. D. 2006. Local textual inference: It's hard to circumscribe, but you know it when you see it - and nlp needs it. MS. Stanford University.

Moll, D. and B. Hutchinson. 2003. Intrinsic versus extrinsic evaluations of parsing systems. In *Proceedings European Association for Computational Linguistics (EACL), workshop on Evaluation Initiatives in Natural Language Processing*, Budapest.

Nairn, R., C. Condoravdi, and L. Kartunen. 2006. Computing relative polarity for textual inference. In *Proceedings of ICoS-5 (Inference in Computational Semantics)*, Buxton, UK.

Sagot, B. and E. de La Clergerie. 2006. Error mining in parsing results. In *Proceedings of ACL-CoLing 06*, pages 329–336, Sydney, Australie.

Vanderwende, L., D. Coughlin, and B. Dolan. 2005. What syntax can contribute in entailment task. In *Proceedings of the First PASCAL RTE Workshop*, pages 13–17.

Query Expansion based on Pseudo Relevance Feedback from Definition Clusters

Delphine Bernhard
LIMSI-CNRS
Delphine.Bernhard@limsi.fr

Abstract

Query expansion consists in extending user queries with related terms in order to solve the lexical gap problem in Information Retrieval and Question Answering. The main difficulty lies in identifying relevant expansion terms in order to prevent query drift. We propose to use definition clusters built from a combination of English lexical resources for query expansion. We apply the technique of pseudo relevance feedback to obtain expansion terms from definition clusters. We show that this expansion method outperforms both local feedback, based on the document collection, and expansion with WordNet synonyms, for the task of document retrieval in Question Answering.

1 Introduction

Question Answering (QA) systems aim at providing precise answers to user questions. Most QA systems integrate a document retrieval component, which is in charge of retrieving the most relevant documents or passages for a given user question. Since document retrieval is performed in early stages of QA, it is of the uttermost importance that all relevant documents be retrieved, to limit the loss of relevant answers for further processing. However, document retrieval systems have to solve the lexical gap problem, which arises from alternative ways of conveying the same piece of information in questions and answers. One of the solutions proposed to deal with this issue is query expansion (QE), which consists in extending user queries with related terms.

This paper describes a new method for using lexical-semantic resources in query expansion with a focus on QA applications. While some research has been devoted to using explicit semantic relationships for QE, such as synonymy or hypernymy, with rather disappointing results (Voorhees, 1994), we focus on the usefulness of textual and unstructured dictionary definitions for question expansion. Definitions extracted from seven English lexical resources are first grouped to obtain definition clusters, which capture redundancies and sense mappings across resources. Expansion terms are extracted from these definition clusters using pseudo relevance feedback: we first retrieve the definition clusters which are most related to the user query, and then extract the most relevant terms from these definition clusters to expand the query.

The contributions of this work are as follows: (i) we build definition clusters across seven different lexical resources for English, (ii) we thoroughly compare different question expansion methods using local and global feedback, and (iii) we address both the lexical gap and question ambiguity problems by integrating expansion and disambiguation in one and the same step.

In the next section, we describe related work. In Section 3, we describe our method for acquiring definition clusters from seven English lexical resources. In Section 4, we detail query expansion methods. We present experimental results in Section 5 and conclude in Section 6.

2 Related Work

Query expansion attempts to solve the vocabulary mismatch problem by adding new semantically related terms to the query. The goal is to increase recall by retrieving more relevant documents. Two types of query expansion methods are usually distinguished (Manning et al., 2008): *global* techniques, which do not take the results obtained for the original query into account, and

Coling 2008: Poster Volume, pages 54–62,
Beijing, August 2010

local techniques, which expand the query based on an analysis of the documents returned. Local methods are also known as *relevance feedback*.

A first type of global QE methods relies on external hand-crafted lexical-semantic resources such as WordNet. While expansion based on external resources is deemed more efficient than expansion relying on relevance feedback, it also has to tackle problems of semantic ambiguity, which explains why local analysis has been shown to be generally more effective than global analysis (Xu and Croft, 1996). However, recent work by Fang (2008) has demonstrated that global expansion based on WordNet and co-occurrence based resources can lead to performance improvement in an axiomatic model of information retrieval.

Corpus-derived co-occurrence relationships are also exploited for query expansion. Qiu and Frei (1993) build a corpus-based similarity thesaurus using the method described in Schütze (1998) and expand a query with terms which are similar to the query concept based on the similarity thesaurus. Song and Bruza (2003) construct vector representations for terms from the target document collection using the Hyperspace Analogue to Language (HAL) model (Lund and Burgess, 1996). The representations for all the terms in the query are then combined by a restricted form of vector addition. Finally, expansion terms are derived from this combined vector by information flow.

Quasi-parallel monolingual corpora have been recently employed for query expansion, using statistical machine translation techniques. Expansion terms are acquired by training a translation model on question-answer pairs (Riezler et al., 2007) or query-snippets pairs (Riezler et al., 2008) and by extracting paraphrases from bilingual phrase tables (Riezler et al., 2007).

The main difficulty of QE methods lies in selecting the most relevant expansion terms, especially when the query contains ambiguous words. Moreover, even if the original query is not ambiguous, it might become so after expansion. Recent attempts at integrating word sense disambiguation (WSD) in IR within the CLEF Robust WSD track[1] have led to mixed results which show that in most cases WSD does not improve performance of monolingual and cross-lingual IR systems (Agirre et al., 2009). For query expansion based on translation models, ambiguity problems are solved by a language model trained on queries (Riezler et al., 2008), in order to select the most likely expansion terms in the context of a given query.

In this article, we propose to integrate disambiguation and expansion in one and the same step by retrieving expansion terms from definition clusters acquired by combining several English lexical resources.

3 Acquisition of Definition Clusters

Dictionary definitions constitute a formidable resource for Natural Language Processing. In contrast to explicit structural and semantic relations between word senses such as synonymy or hypernymy, definitions are readily available, even for less-resourced languages. Moreover, they can be used for a wide variety of tasks, ranging from word sense disambiguation (Lesk, 1986), to producing multiple-choice questions for educational applications (Kulkarni et al., 2007) or synonym discovery (Wang and Hirst, 2009). However, all resources differ in coverage and word sense granularity, which may lead to several shortcomings when using a single resource. For instance, the sense inventory in WordNet has been shown to be too fine-grained for efficient word sense disambiguation (Navigli, 2006; Snow et al., 2007). Moreover, gloss and definition-based measures of semantic relatedness which rely on the overlap between the definition of a target word and its distributional context (Lesk, 1986) or the definition of another concept (Banerjee and Pedersen, 2003) yield low results when the definitions provided are short and do not overlap sufficiently.

As a consequence, we propose combining lexical resources to alleviate the coverage and granularity problems. To this aim, we automatically build cross-resource sense clusters. The goal of our approach is to capture redundancy in several resources, while improving coverage over the use of a single resource.

[1] http://ixa2.si.ehu.es/clirwsd/

3.1 Resources

In order to build definition clusters, we used the following seven English resources:

WordNet We used WordNet 3.0, which contains 117,659 synset definitions.[2]

GCIDE The GCIDE is the GNU version of the Collaborative International Dictionary of English, derived from Webster's 1913 Revised Unabridged Dictionary. We used a recent XML version of this resource,[3] from which we extracted 196,266 definitions.

English Wiktionary and Simple English Wiktionary Wiktionary is a collaborative online dictionary, which is also available in a simpler English version targeted at children or non-native speakers. We used the English Wiktionary dump dated August 16, 2009 and the Simple English Wiktionary dump dated December 9, 2009. The English Wiktionary comprises 245,078 definitions, while the Simple English Wiktionary totals 11,535 definitions.

English Wikipedia and Simple English Wikipedia Wikipedia is a collaborative online encyclopedia. As Wiktionary, it provides a Simple English version. We used the Mediawiki API to extract 152,923 definitions from the English Wikipedia[4] and 53,993 definitions from the Simple English Wikipedia. Since full Wikipedia articles can be very long in comparison to the other resources, we only retrieved the first sentence of each page to constitute the definition database, following (Kazama and Torisawa, 2007).

OmegaWiki OmegaWiki is a collaborative multilingual dictionary based on a relational database. We used the SQL database dated December 17, 2009,[5] comprising 29,179 definitions.

[2]Statistics obtained from `http://wordnet.princeton.edu/wordnet/man/wnstats.7WN.html`

[3]Retrieved from `http://rali.iro.umontreal.ca/GCIDE/`

[4]As we mainly aimed at capturing the redundancy across resources, we only extracted definitions for the Wikipedia terms which were also found in the GCIDE, Omegawiki, Wiktionary or Simple English Wikipedia.

[5]Retrieved from `http://omegawiki.org/`

3.2 Definition Clustering

In order to cluster definitions, we first build a definition graph: each node in the graph corresponds to a definition in one of our input resources and two definition nodes are linked if they define the same term and their definitions are similar enough. Links are weighted by the cosine similarity of the definition nodes. To compute the cosine similarity, we stem the definition words with the Porter Stemmer and remove stop words. Moreover, we weigh words with their *tf.idf* value in the definitions. Document frequency (*df*) counts are derived from the definitions contained in all our resources.

Definition clusters are identified with a community detection algorithm applied to the definition graph. Communities correspond to groups of nodes with dense interconnections: in our case, we aim at retrieving groups of related definitions. We used the algorithm proposed by Blondel et al. (2008), based on modularity optimisation.[6] The modularity function measures the quality of a division of a graph into communities (Newman and Girvan, 2004).

In order to increase the precision of clustering, we remove edges from the graph whose cosine value is lower than a given threshold.

3.3 Evaluation of Definition Clusters

We built a gold-standard by manually grouping the definitions contained in our source resources for 20 terms from the Basic English Word List,[7] totalling 726 definitions, grouped in 321 classes. We evaluated the definition clusters in terms of clustering purity (Manning et al., 2008), which is a classical evaluation measure to measure clustering quality. Purity is defined as follows:

$$purity(\Omega, C) = \frac{1}{N} \sum_{k} \max_{j} |\omega_k \cap c_j| \qquad (1)$$

where N is the number of clustered definitions, $\Omega = \{\omega_1, \omega_2, \ldots, \omega_K\}$ is the set of definition

[6]We used its Python implementation by Thomas Aynaud, available at `http://perso.crans.org/aynaud/communities/community.py` [Visited on October 26, 2009].

[7]`http://en.wiktionary.org/wiki/Appendix:Basic_English_word_list`

Resource	Definition
WordNet	an arc of colored light in the sky caused by refraction of the sun's rays by rain
Gcide	A bow or arch exhibiting, in concentric bands, the several colors of the spectrum, and formed in the part of the hemisphere opposite to the sun by the refraction and reflection of the sun's rays in drops of falling rain.
Simple Wikipedia	A rainbow is an arc of color in the sky that you can see when the sun shines through falling rain.
Simple Wiktionary	The arch of colours in the sky you can see in the rain when the sun is at your back.

Table 1: Excerpt from a definition cluster.

clusters obtained, w_k is the set of definitions in cluster k, $C = \{c_1, c_2, \ldots, c_J\}$ is the set of definition families expected and c_j is the set of definitions in family j.

We also report the amount of clusters obtained for each cosine threshold value. The evaluation results are detailed in Table 2.

Cosine threshold	Purity	# Clusters
0.0	0.363	73
0.1	0.464	135
0.2	0.644	234
0.3	0.848	384
0.4	0.923	458
0.5	0.957	515

Table 2: Evaluation results for definition clustering.

Overall, the results which account for the best compromise between purity and cluster count are obtained for a threshold of 0.3: for this threshold, we obtain 384 clusters, which is closest to the expected value of 321 classes. The purity obtained for this cosine threshold is very close to the values obtained by Kulkarni et al. (2007), who clustered definitions extracted from only two source dictionaries and report a purity of 0.88 for their best results. In total we obtain 307,570 definition clusters. Table 1 displays an excerpt from one of the definition clusters obtained.

4 Query Expansion Methods

In this section, we describe the methods used for performing query expansion. We first describe two simple baseline methods, one based on local feedback, the other based on WordNet. Then, we detail our method relying on the definition clusters previously described.

4.1 Query Expansion based on Local Feedback

In order to perform local feedback based on the document collection, we used the pseudo relevance feedback methods implemented in the Terrier information retrieval platform (Ounis et al., 2007): Bo1 (Bose-Einstein 1), Bo2 (Bose-Einstein 2) and KL (Kullback-Leibler). These methods extract informative terms from the top-ranked documents retrieved using the original query and use them for query expansion.

4.2 Query Expansion based on WordNet Synonyms

As a second baseline for query expansion, we expand the query terms with their synonyms extracted from WordNet. For each query term t, we retrieve its WordNet synsets and keep the corresponding synset members as expansion terms.[8] We weigh the expansion terms in each synset by the frequency score provided in WordNet, which indicates how often the query term t occurs with the corresponding sense. In the rest of the paper, this method is referred to as **WN-synonyms**.

The expansion terms obtained using WN-synonyms are further reweighted using Rocchio's *beta* formula which computes the weight qtw of

[8] We use NLTK (`http://www.nltk.org/`) to access WordNet.

query term t as follows (Rocchio, 1971; Macdonald et al., 2005):

$$qtw = \frac{qtf}{qtf_{max}} + \beta \frac{w(t)}{w_{max}(t)} \quad (2)$$

where qtf is the frequency of term t in the query, qtf_{max} is the maximum query term frequency among the query terms, $w(t)$ is the expansion weight of t, detailed in Equation 3, and $w_{max}(t)$ is the maximum $w(t)$ of the expansion terms. In all our experiments, β is set to 0.4, which is the default value used in Terrier.

Given this formula, if an original query term occurs among the expansion terms, its weight in the expanded query increases. For expansion terms which do not occur in the original query, $qtf = 0$.

This formula has been proposed in the setting of pseudo relevance feedback, where expansion terms are chosen based on the top documents retrieved for the original query. However, in our WN-synonyms setting, one and the same expansion term might be obtained from different original query terms with different weights. It is therefore necessary to obtain a global similarity weight for one expansion term with respect to the whole query. Following Qiu and Frei (1993), we define $w(t)$ as:

$$w(t) = \frac{\sum_{t_i \in q} qtf_i \cdot s(t, t_i)}{\sum_{t_i \in q} qtf_i} \quad (3)$$

where q is the original query and $s(t, t_i)$ is the similarity between expansion term t and query term t_i, i.e., the frequency score in WordNet.

For final expansion, we keep the top T terms with the highest expansion weight.

4.3 Query Expansion Based on Definition Clusters

In order to use definition clusters (DC) for query expansion, we first use Terrier to index the clusters which obtained the best overall results in our evaluation of definition clustering, corresponding to a cosine threshold of 0.3.[9] For each cluster, we index both the definitions and the list of terms they define, which makes it possible to include synonyms or Wikipedia redirects in the index.

For a given question, we retrieve the top D definition clusters: the retrieval of definition clusters is based on all the question terms, and thus enables indirect contextual word sense disambiguation. Then, we extract expansion terms from these clusters using pseudo relevance feedback (PRF) as implemented in Terrier. The top T most informative terms are retrieved from the top D definition clusters retrieved and used for expansion. The expansion terms are weighted using the KL (Kullback-Leibler) term weighting model in Terrier. We chose this particular weighting model, as it yielded the best results for local feedback (see Table 3).

We name this method **DC-PRF**.

5 Experiments

In this section, we describe the experimental results obtained for the query expansion methods presented in the previous section. We used the Microsoft Research Question-Answering Corpus[10] (MSRQA) as our evaluation dataset.

5.1 Microsoft Research Question-Answering Corpus (MSRQA)

MSRQA provides a fully annotated set of questions and answers retrieved from the Encarta 98 encyclopedia. The Encarta corpus contains 32,715 articles, ranging from very short (3 tokens) to very long (59,798 tokens). QA systems usually split documents into smaller passages. We have therefore segmented the Encarta articles into smaller parts representing subsections of the original article, using a regular expression for identifying section headers in the text. As a result, the dataset comprises 61,604 documents, with a maximum of 2,730 tokens. The relevance judgements provided comprise the document id as well as the sentences (usually one) containing the answer. We processed these sentence level relevance judgements to obtain judgements for documents: a document is considered as relevant if it contains an exact answer sentence. Overall, we obtained relevance judgements for 1,098 questions.

[9] We used the 2.2.1 version of Terrier, downloadable from `http://terrier.org/`

[10] Downloadable from `http://research.microsoft.com/en-us/downloads/88c0021c-328a-4148-a158-a42d7331c6cf/`

	All questions		Easy questions		Medium questions		Hard questions	
Expansion	**MAP**	**MRR**	**MAP**	**MRR**	**MAP**	**MRR**	**MAP**	**MRR**
none	0.2257	0.2681	0.2561	0.3125	0.1720	0.1965	0.1306	0.1392
Terrier-Bo1	0.2268	0.2674	0.2625	0.3157	0.1642	0.1903	0.1222	0.1240
Terrier-Bo2	0.2234	0.2602	0.2581	0.3077	0.1660	0.1872	0.1126	0.1146
Terrier-KL	0.2274	0.2684	0.2635	0.3167	0.1644	0.1915	0.1220	0.1236
WN-synonyms	0.2260	0.2687	0.2536	0.3098	0.1785	0.2055	0.1254	0.1260
DC-PRF	**0.2428**	**0.2929**	**0.2690**	**0.3361**	**0.2004**	**0.2294**	0.1385	0.1472
	+7.6%	+9.2%	+5.0%	+7.5%	+16.5%	+16.7%	+6.0%	+5.7%
DC-PRF + Terrier KL	0.2361	0.2796	0.2625	0.3184	0.1928	0.2213	**0.1389**	**0.1484**

Table 3: Experimental results. The performance gaps between the DC-PRF and the baseline retrieval models without expansion (none), Terrier-KL and WN-synonyms are statistically significant (two-tailed paired t-test, $p < 0.05$), except for hard questions and for the MAP comparison with Terrier-KL for easy questions. We also report the improvement percentage.

Based on the annotations available in the MSRQA dataset, we further distinguish three question types:

- *easy* questions, which have at least one answer with a strong match (two or more query terms in the answer).
- *medium* questions, which have no strong match answer, but at least an answer with a weak match (one query term in the answer).
- *hard* questions, which have neither a strong nor a weak match answer, but only answers which contain no query terms, and at the best synonyms and derivational morphological variants for query terms.

Overall, the evaluation dataset comprises 651 easy questions, 397 medium questions and 64 hard questions (some of these questions have no exact answer).

5.2 Results

As our baseline, we use the BB2 (Bose-Einstein model for randomness) retrieval model in Terrier. We varied the values for the parameters T (number of expansion terms) and D (number of expansion documents) and used the settings yielding the best evaluation results. For the PRF methods implemented in Terrier, the default settings (T=10, D=3) worked best; for DC-PRF, we used T=20 and D=40. Finally, for WN-synonyms we used T=10. We also combined both DC-PRF and Terrier-KL by first applying DC-PRF expansion and then using local Terrier-KL feedback on the retrieved documents (DC-PRF + Terrier KL). Prior to retrieval, all questions are tokenised and part-of-speech tagged using Xerox's Incremental Parser XIP (Aït-Mokhtar et al., 2002). Moreover, we retrieve 100 documents for each question and stem the Encarta document collection. The results shown in Table 3 are evaluated in terms of Mean-Average Precision (MAP) and Mean Reciprocal Rank (MRR). Table 4 provides examples of the top 5 expansion terms obtained for each expansion method.

The DC-PRF expansion method performs best overall, as well as for easy and medium question types. For medium questions, DC-PRF leads to an increase of 16.5% in MAP and 16.7% in MRR, with respect to the 'none' baseline. Local feedback methods, such as Terrier-KL, only bring minor improvements for easy questions, but lead to slightly lower results for medium and hard questions. This might be due to the small size of the document collection, which therefore lacks redundancy. The simple baseline expansion method based on WordNet leads to very slight improvements for medium questions over the setting without expansion. The combination of DC-PRF and Terrier-KL leads to lower results than using only

Terrier-KL	WN-synonyms	DC-PRF
12: *Are there UFOs?*		
sight – unidentifi – report – object – fly	flying – unidentified – object – UFO – saucer	unidentified – ufo – flying – ufology – objects
104: *What is the most deadly insect in the world?*		
speci – plant – feed – anim – liv	cosmos – creation – existence – macrocosm – universe	nightshade – belladonna – mortal – death – lethal
107: *When was the little ice age*		
drift – glacial – ago – sheet – million	small – slight – historic – period – water	floe – period – glacial – cold – interglacial
449: *How does a TV screen get a picture from the air waves?*		
light – beam – televi – electron – signal	moving – ridge – image – icon – ikon	television – movie – image – motion – door
810: *Do aliens really exist?*		
sedition – act – govern – deport – see	live – subsist – survive – alienate – extraterrestrial	alien – extraterrestrial – monsters – dreamworks – animation

Table 4: Expansion examples. The expansion terms produced by Terrier-KL are actually stemmed, as they are retrieved from a stemmed index.

DC-PRF, except for hard questions, for which the combination brings a very slight improvement.

The expansion samples provided in Table 4 exemplify the query drift problem of local feedback methods (Terrier-KL): for question 810, expansion terms focus on the "foreigner" sense of *alien* rather than on the "extraterrestrial" sense. The WN-synonyms method suffers from the problem of weighting synonyms, and mainly focuses on synonyms for the most frequent term of the question, e.g. "world" in question 104. Interestingly, the DC-PRF method accounts for neologisms, such as "ufology" which can be found in new collaboratively constructed resources such as Wikipedia or Wiktionary, but not in WordNet. This is made possible by the combination of diversified resources. It is also able to provide encyclopedic knowledge, such as "dreamworks" and "animation" in question 810, referring to the feature film "Monsters vs. Aliens".

The DC-PRF method also has some limitations. Even though the expansion terms "dreamworks" and "animation" correspond to the intended meaning of the word "alien" in question 810, they nevertheless might introduce some noise in the retrieval. Some other cases exemplify slight drifts in meaning from the query: in question 104, the expansion terms "nightshade" and "belladonna" refer to poisonous plants and not insects; "*deadly* nightshade" is actually the other name of the "belladonna". Similarly, in question 449, the expansion term "door" is obtained, in relation to the word "screen" in the question (as in "screen door"). This might be due to the fact that the terms defined by the definition clusters are indexed as well, leading to a high likelihood of retrieving syntagmatically related terms for multiword expressions. In future work, it might be relevant to experiment with different indexing schemes for definition clusters, e.g. indexing only the definitions, or adding the defined terms to the index only if they are not present in the definitions.

6 Conclusions and Future Work

In this paper, we presented a novel method for query expansion based on pseudo relevance feedback from definition clusters. The definition clusters are built across seven different English lexical resources, in order to capture redundancy while improving coverage over the use of a single resource. The expansions provided by feedback from definition clusters lead to a significant im-

provement of the retrieval results over a retrieval setting without expansion.

In the future, we would like to further ameliorate definition clustering and incorporate other resources, e.g. resources for specialised domains. Moreover, we have shown that query expansion based on definition clusters is most useful when applied to medium difficulty questions. We therefore consider integrating automatic prediction of query difficulty to select the best retrieval method. Finally, we would like to evaluate the method presented in this paper for larger datasets.

Acknowledgments

This work has been partially financed by OSEO under the Quæro program.

References

Agirre, Eneko, Giorgio M. Di Nunzio, Thomas Mandl, and Arantxa Otegi. 2009. CLEF 2009 Ad Hoc Track Overview: Robust - WSD Task. In *Working Notes for the CLEF 2009 Workshop*, Corfu, Greece.

Aït-Mokhtar, Salah, Jean-Pierre Chanod, and Claude Roux. 2002. Robustness beyond shallowness: incremental deep parsing. *Natural Language Engineering*, 8(2-3):121–144.

Banerjee, Satanjeev and Ted Pedersen. 2003. Extended Gloss Overlaps as a Measure of Semantic Relatedness. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*, pages 805–810.

Blondel, Vincent D., Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008+, October.

Fang, Hui. 2008. A Re-examination of Query Expansion Using Lexical Resources. In *Proceedings of ACL-08: HLT*, pages 139–147, Columbus, Ohio, June.

Kazama, Jun'ichi and Kentaro Torisawa. 2007. Exploiting Wikipedia as External Knowledge for Named Entity Recognition. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 698–707.

Kulkarni, Anagha, Jamie Callan, and Maxine Eskenazi. 2007. Dictionary Definitions: The Likes and the Unlikes. In *Proceedings of Speech and Language Technology in Education (SLaTE2007)*, pages 73–76.

Lesk, Michael. 1986. Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In *SIGDOC '86: Proceedings of the 5th annual international conference on Systems documentation*, pages 24–26.

Lund, Kevin and Curt Burgess. 1996. Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior Research Methods, Instruments & Computers*, 28(2):203–208.

Macdonald, Craig, Ben He, Vassilis Plachouras, and Iadh Ounis. 2005. University of Glasgow at TREC 2005: Experiments in Terabyte and Enterprise Tracks with Terrier. In *Proceedings of the 14th Text REtrieval Conference (TREC 2005)*, Gaithersburg, MD, USA.

Manning, Christopher D., Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press.

Navigli, Roberto. 2006. Meaningful clustering of senses helps boost word sense disambiguation performance. In *ACL-44: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 105–112.

Newman, M. E. J. and M. Girvan. 2004. Finding and evaluating community structure in networks. *Physical review E*, 69.

Ounis, Iadh, Christina Lioma, Craig Macdonald, and Vassilis Plachouras. 2007. Research Directions in Terrier: a Search Engine for Advanced Retrieval on the Web. *Novatica/UPGRADE Special Issue on Web Information Access, Ricardo Baeza-Yates et al. (Eds), Invited Paper*.

Qiu, Yonggang and Hans-Peter Frei. 1993. Concept based query expansion. In *SIGIR '93: Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 160–169.

Riezler, Stefan, Alexander Vasserman, Ioannis Tsochantaridis, Vibhu Mittal, and Yi Liu. 2007. Statistical Machine Translation for Query Expansion in Answer Retrieval. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 464–471, Prague, Czech Republic, June.

Riezler, Stefan, Yi Liu, and Alexander Vasserman. 2008. Translating Queries into Snippets for Improved Query Expansion. In *Proceedings of the*

22nd International Conference on Computational Linguistics (Coling 2008), pages 737–744, Manchester, UK, August.

Rocchio, J., 1971. *The SMART Retrieval System*, chapter Relevance Feedback in Information Retrieval, pages 313–323.

Schütze, Hinrich. 1998. Automatic Word Sense Discrimination. *Computational Linguistics*, 24(1):97–123.

Snow, Rion, Sushant Prakash, Daniel Jurafsky, and Andrew Y. Ng. 2007. Learning to Merge Word Senses. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 1005–1014, Prague, Czech Republic, June.

Song, Dawei and Peter D. Bruza. 2003. Towards context sensitive information inference. *Journal of the American Society for Information Science and Technology (JASIST)*, 54(4):321–334.

Voorhees, Ellen M. 1994. Query expansion using lexical-semantic relations. In *SIGIR '94: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 61–69.

Wang, Tong and Graeme Hirst. 2009. Extracting Synonyms from Dictionary Definitions. In *Proceedings of RANLP 2009*.

Xu, Jinxi and W. Bruce Croft. 1996. Query expansion using local and global document analysis. In *SIGIR '96: Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 4–11.

A Formal Scheme for Multimodal Grammars

Philippe Blache & Laurent Prévot
LPL-CNRS, Université de Provence
blache@lpl-aix.fr

Abstract

We present in this paper a formal approach for the representation of multimodal information. This approach, thanks to the to use of *typed feature structures* and *hypergraphs*, generalizes existing ones (typically annotation graphs) in several ways. It first proposes an homogenous representation of different types of information (nodes and relations) coming from different domains (speech, gestures). Second, it makes it possible to specify constraints representing the interaction between the different modalities, in the perspective of developing *multimodal grammars*.

1 Introduction

Multimodality became in the last decade an important challenge for natural language processing. Among the problems we are faced with in this domain, one important is the understanding of how does the different modalities interact in order to produce meaning. Addressing this question requires to collect data (building corpora), to describe them (enriching corpora with annotations) and to organize systematically this information into a homogeneous framework in order to produce, ideally, multimodal grammars.

Many international projects address this question from different perspectives: data representation and coding schemes (cf. ISLE (Dybkjaer, 2001), MUMIN (Allwood, 2005), etc.), corpus annotation (cf. LUNA (Rodriguez, 2007) or DIME (Pineda, 2000), etc.), annotation and editing tools (such as NITE NXT (Carletta, 2003), Anvil (Kipp, 2001), Elan (Wittenburg, 2006), Praat (Boersma, 2009), etc.).

We propose in this paper a generic approach addressing both formal representation and concrete annotation of multimodal data, that relies on *typed-feature structure* (TFS), used as a description language on graphs. This approach is generic in the sense that it answers to different needs: it provides at the same time a formalism directly usable for corpus annotation and a description language making it possible to specify constraints that constitute the core of a *multimodal grammar*.

In the first section, we motivate the use of TFS and present how to concretely implement them for multimodal annotation. We address in the second section one of the most problematic question for multimodal studies: how to represent and implement the relations between the different domains and modalities (a simple answer in terms of time alignment being not powerful enough). In the last section, we describe how to make use of this representation in order to specify multimodal grammars.

2 Typed-feature structures modeling

Information representation is organized in two dimensions: type hierarchies and constituency relations (typically, a prosodic unit is a set of syllables, which in turn are sets of phonemes). The former corresponds to an *is-a* relation, the latter to a *part-of* one. For example *intonational phrase* is a subtype of *prosodic phrase*, and *phonemes* are constituents of *syllables*.

Such an organization is directly represented by means of typed feature structures. They can be considered as a formal annotation schema, used as

Coling 2008: Poster Volume, pages 63–71,
Beijing, August 2010

a preliminary step before the definition of the concrete coding scheme[1]. This step is necessary when bringing together information (and experts) from different fields: it constitutes a common representation framework, homogenizing information representation. Moreover, it allows to clearly distinguish between knowledge representation and annotation. The coding scheme, at the annotation level (labels, features, values), is deduced from this formal level.

The remaining of the section illustrates how to represent objects from different domains by means of TFS. The Figure 1 presents the type hierarchy and the constituency structure of objects taken here as example.

2.1 Phonetics

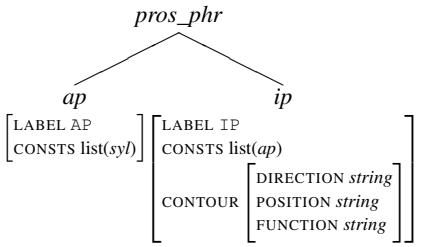
The phoneme is used as primary data: this object is at the lowest level of the constituent hierarchy (most of the objects are set of phonemes). The following feature structure proposes a precise encoding of the main properties describing a phoneme, including articulatory gestures.

$$
\textit{phon}\left[\begin{array}{l}
\text{SAMPA\_LABEL } \textit{sampa\_unit} \\
\text{CAT}\left\{\textit{vowel, consonant}\right\} \\
\text{TYPE}\left\{\textit{occlusive, fricative, nasal, etc.}\right\} \\
\text{ARTICULATION}\left[\begin{array}{l}
\text{LIP}\left[\begin{array}{l}\text{PROTUSION } \textit{string} \\ \text{APERTURE } \textit{aperture}\end{array}\right] \\
\text{TONGUE}\left[\begin{array}{l}\text{TIP}\left[\begin{array}{l}\text{LOCATION } \textit{string} \\ \text{DEGREE } \textit{string}\end{array}\right] \\ \text{BODY}\left[\begin{array}{l}\text{LOCATION } \textit{string} \\ \text{DEGREE } \textit{string}\end{array}\right]\end{array}\right] \\
\text{VELUM } \textit{aperture} \\
\text{GLOTTIS } \textit{aperture}
\end{array}\right] \\
\text{ROLE}\left[\begin{array}{l}\text{EPENTHETIC } \textit{boolean} \\ \text{LIAISON } \textit{boolean}\end{array}\right]
\end{array}\right]
$$

Phonemes being at the lowest level, they do not have any constituents. They are not organized into precise subtypes. The feature structure represent then the total information associated with this type.

2.2 Prosody

As seen above, prosodic phrases are of two different subtypes: *ap* (accentual phrases) and *ip* (intonational phrases). The prosodic type hierarchy is represented as follows:



Accentual phrases have two appropriate features: the label which is simply the name of the corresponding type, and the list of constituents, in this case a list of syllables. The objects of type *ip* contain the list of its constituents (a set of *ap*s) as well as the description of its contour. A contour is a prosodic event, situated at the end of the *ip* and is usually associated to an *ap*.

The prosodic phrases are defined as set of syllables. They are described by several appropriate features: the syllable structure, its position in the word, its possibility to be accented or prominent:

$$
\textit{syl}\left[\begin{array}{l}
\text{STRUCT } \textit{syl\_struct} \\
\text{POSITION}\left[\begin{array}{l}\text{RANK}\left\{\textit{integer}\right\} \\ \text{SYL\_NUMBER}\left\{\textit{integer}\right\}\end{array}\right] \\
\text{ACCENTUABLE } \textit{boolean} \\
\text{PROMINENCE } \textit{boolean} \\
\text{CONSTITUENTS list}(\textit{const\_syl})
\end{array}\right]
$$

Syllable constituents (objects of type *const_syl*) are described by two different features: the set of phonemes (syllable constituents), and the type of the constituent (onset, nucleus and coda). Note that each syllable constituent can contain a set of phonemes.

$$
\textit{const\_syl}\left[\begin{array}{l}
\text{PHON list}(\textit{phon}) \\
\text{CONST\_TYPE}\left\{\textit{onset, nucleus, coda}\right\}
\end{array}\right]
$$

2.3 Disfluencies

We can distinguish two kinds of disfluencies: *non lexicalized* (without any lexical material, such as lengthening, silent pauses or filled pauses) and *lexicalized* (non-voluntary break in the phrasal flow, generating a word or a phrase fragment). Lexicalized disfluencies have a particular organization with three subparts (or constituents):

- *Reparandum*: the word or phrase fragment, in which the break occurs
- *Break*: a point or an interval that can eventually be filled by a fragment repetition, parenthetical elements, etc.

[1]This approach has been first defined and experimented in the XXXX project, not cited for anonymity reasons.

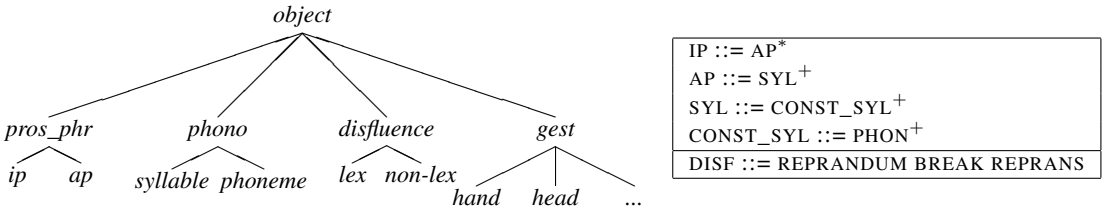
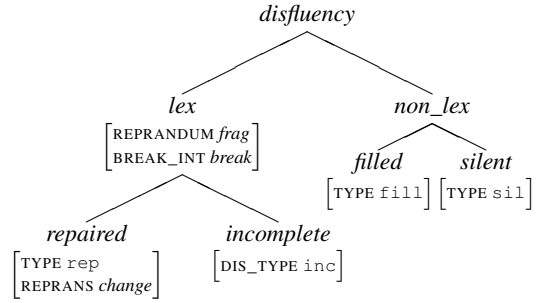
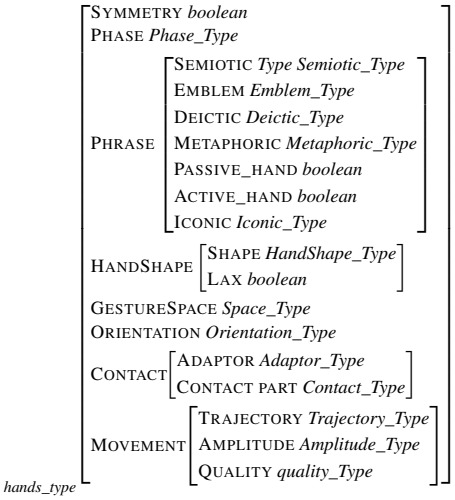


Figure 1: *Type and constituent hierarchies*

- *Reparans*: all that follow the break and recovers the reparandum (in modifying or completing it) or simply left it uncompleted.

The general disfluency type hierarchy, with the appropriate features at each level is given in the following figure:





2.4 Gestures

Besides verbal communication, gestures constitute the main aspect of multimodality. In multimodal annotation, this is probably the most difficult and time-consuming task. Moreover, only few works really focus on a precise description of all the different domains of verbal and non verbal modalities. The TFS-based approach proposed here answers to the first need in such a perspective: a common representation framework.

We give in this section a brief illustration of the representation of one gesture (hands). It relies on adaptation of different proposals, especially (Kipp03) or MUMIN (Allwood, 2005), both integrating McNeill's gesture description (McNeill05).

The following structure encodes the description of gesture phases, phrases (representing different semiotic types), the hand shape as well as its orientation, the gesture space, and the possible contact with bodies or objects. A last feature also describes the movement itself: trajectory, quality (fast, normal or slow) and amplitude (small, medium and large).

2.5 Application

We have experimented this modeling in the complete annotation of a multimodal corpus (see (Blache, 2010)). In this project, a complete TFS model has been first designed, covering all the different domains (prosody, syntax, gestures, discourse, etc.). From this model, the annotations have been created, leading to a 3-hours corpus of narrative dialogs, fully transcribed. The corpus is fully annotated for some domains (phonetics, prosody and syntax) and partly for others (gestures, discourse, disfluencies, specific phenomena). The result is one of the first large annotated multimodal corpus.

3 Graphs for Multimodal Annotation

Graphs are frequently used in the representation of complex information, which is the case with multimodality. As for linguistic annotation, one of the most popular representations is *Annotation Graphs* (Bird, 2001). They have been proposed in particular in the perspective of anchoring different kinds of information in the same reference,

making it possible to align them[2]. In AGs, nodes represent positions in the signal while edges bear linguistic information. Two edges connecting the same nodes are aligned: they specify different information on the same part of the input. Implicitly, this means that these edges bear different features of the same object.

Such a representation constitutes the basis of different approaches aiming at elaborating generic annotation formats, for example LAF (and its extension GrAF (Ide, 2007)). In this proposal, edge labels can be considered as nodes in order to build higher level information. One can consider the result as an *hypergraph*, in which nodes can be subgraphs.

We propose in this section a more generalized representation in which nodes are not positions in the signal, but represent directly objects (or set of objects). All nodes have here the same structure, being them nodes or hypernodes. The main interest of this proposal, on top of having an homogeneous representation, is the possibility to anchor information in different references (temporal, spatial or semantic).

3.1 Nodes

As seen above, multimodal annotation requires the representation of different kinds of information (speech signal, video input, word strings, images, etc.). The *objects*[3] that will be used in the description (or the annotation) of the input are of different nature: temporal or spatial, concrete or abstract, visual or acoustic, etc. A generic description requires first a unique way of locating (or indexing) all objects, whatever their domain. In this perspective, an index (in the HPSG sense) can be specified, relying on different information:

- LOCATION: objects can in most of the cases be localized in reference to a temporal or a spatial situation. For example, phonemes have a temporal reference into the speech signal, physical objects have spatial localization that can be absolute (spatial coordinates), or relative (with respect to other objects).
- REALIZATION: data can either refer to *concrete* or physical objects (phonemes, gestures, referential elements, etc.) as well as *abstract* ones (concepts, emotions, etc.).
- MEDIUM: specification of the different modalities: *acoustic, tactile* and *visual*.[4]
- ACCESSIBILITY: some data are directly accessible from the signal or the discourse, they have a physical existence or have already been mentioned. In this case, they are said to be "*given*" (e.g. gestures, sounds, physical objects). Some other kinds of data are deduced from the context, typically the abstract ones. They are considered as "*accessible*".

A generic structure node can be given, gathering the index and the some other object properties.

$$
\begin{bmatrix}
\text{ID} \\
\text{DOMAIN} \left\{ \textit{prosody, syntax, pragmatics, ...} \right\} \\
\text{INDEX} \begin{bmatrix}
\text{LOCATION} \left\{ \begin{array}{l} \text{TEMPORAL} \begin{bmatrix} \text{START } \textit{value} \\ \text{END } \textit{value} \end{bmatrix} \\ \text{SPATIAL } \textit{coord} \end{array} \right\} \\
\text{REALIZATION} \left\{ \textit{concrete, abstract} \right\} \\
\text{MEDIUM} \left\{ \textit{acoustic, tactile, visual} \right\} \\
\text{ACCESSIBILITY} \left\{ \textit{given, accessible} \right\}
\end{bmatrix} \\
\text{FEATURES } \textit{object\_type}
\end{bmatrix}_{node}
$$

This structure relies on the different information. Besides INDEX, some other features complete the description:

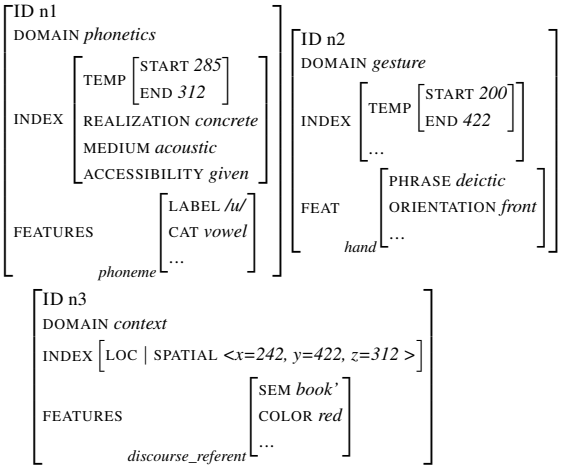
- ID: using an absolute ID is useful in the perspective of graph representation, in which nodes can encode any kind of information (atomic or complex, including subgraphs).
- DOMAIN: specification of the domain to which the information belongs. This feature is useful in the specification of generic interaction constraints between domains.
- FEATURES: nodes have to bear specific linguistic indications, describing object properties. This field encodes the type of information presented in the first section.

[2] Another important interest of AGs is that they can constitute the basis for an exchange format, when thinking on annotation tools interoperability (a proposal is currently elaborated under auspices of the MITRE program, see `http://www.mitre.org/`).

[3] We call *object* any annotation that participates to the description: phoneme, words, gestures, but also phrases, emotions, etc.

[4] See the W3C EMMA recommendation (*Extensible Multi-Modal Annotations*, `http://www.w3.org/2002/mmi/`.

The following examples illustrate the representation of atomic nodes from different domains: a phoneme (node *n1*) and a gesture (node *n2*), that are temporally anchored, and a physical object (node *n3*) which is spatially situated. This last object can be used as a referent, for example by a deictic gesture.

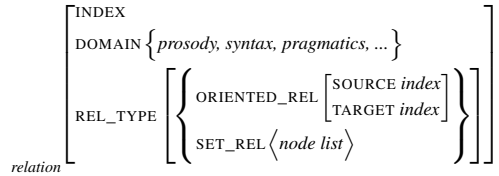


3.2 Relations

Linguistic information is usually defined in terms of relations between (sets of) objects, which can be atomic or complex. For example, a phrase is defined by syntactic relations (government, agreement, linearity, etc.) between its constituents. In some cases, these relations can concern objects from the same domain (e.g. syntax in the previous example). In other cases, different domains can be involved. For example, a long break (greater than 200ms) usually precedes a left corner of a new phrase.

The nature of the relation can also be different according to the kind of information to be encoded. Many relations are binary and oriented (precedence, dependency, etc.). Some others only consists in gathering different objects. A construction (in the sense of *Construction Grammars*, see (Fillmore96)) is precisely that: a set of object or properties that, put together, form a specific phenomenon. It is then useful in our representation to distinguish between *oriented relations* and *set relations*. Oriented relations (for example precedence) connect a source and a target, that can be eventually formed with set of objects. Set relations are used to gather a set of objects, without orientation or order (e.g. the constituency relation).

On top of this distinction, it is also necessary to give an index to the relations, in order to make their reference possible by other objects. As for nodes, an index is used, even though its form is simple and does not need a complex anchor. Finally, for the same reasons as for nodes, the specification of the domain is necessary. The following feature structure gives a first view of this organization:



Besides these information, a relation description has to be completed with other information:

- TYPE: different types of relations can be implemented in such representation, such as dependency, precedence, constituency, anaphore, etc.
- SCOPE: a relation can be specific to a construction or at the opposite valid whatever the context. For example, the precedence relation $[V \prec Clit_{[nom]}]$ is only valid in the context of interrogative constructions whereas the relation exluding the realization of a backchannel[5] after a connective is valid whatever the context. We distinguish then between *local* and *global* scopes.
- POLARITY: a relation can be negated, implementing the impossibility of a relation in a given context.
- CONSTRUCTION: in the case of a local relation, it is necessary to specify the construction to which it belongs.
- STRENGTH: some relation are mandatory, some other optional. As for constraints, we distinguish then between *hard* and *soft* relations, depending on their status.

Finally, a last property has to be precisely defined: the synchronization between two objects

[5] A backchannel is a reaction, verbal or gestual, of the adressee during a conversation.

coming from different domains (for example gestures and words). In some cases, both objects have to be strictly aligned, with same boundaries. For example, a syllable has to be strictly aligned with its set of phonemes: the left syllable boundary (resp. the right) has to be the same as that of the first syllable phoneme (resp. the last). In other cases, the synchronization must not be strict. For example, a deictic gesture is not necessarily strictly aligned with a referential pronoun. In this case, boundaries of both objects only have to be roughly in the same part of the signal.

We propose the definition of alignment operators adapted from (Allen, 1985) as follows:

$=$	*same*	boundaries have to be equal
$<_\Delta$	*before*	$b_1 <_\Delta b_2$ means b_1 value is lower than b_2, with $b_2 - b_1 < \Delta$
$>_\Delta$	*after*	$b_1 >_\Delta b_2$ means that the boundary b_1 follows b_2, with $b_1 - b_2 < \Delta$
$\approx_\Delta$	*almost*	boundaries are neighbors, without order relation, with $\mid b_1 - b_2 \mid \leq \Delta$

This set of operators allow to specify *alignment equations* between different objects. The advantage of this mechanism is that an equation system can describe complex cases of synchronization. For example, a construction can involve several objects from different domains. Some of these objects can be strictly aligned, some others not.

The final TFS representation is as follows:

$$\text{relation}\begin{bmatrix}\text{INDEX}\\ \text{DOMAIN}\{\textit{prosody, syntax, pragmatics, ...}\}\\ \text{REL\_TYPE}\begin{bmatrix}\{\text{ORIENTED\_REL}\begin{bmatrix}\text{SOURCE } \textit{index}\\ \text{TARGET } \textit{index}\end{bmatrix}\\ \{\text{SET\_REL}\langle \textit{node list}\rangle\}\end{bmatrix}\\ \text{TYPE}\{\textit{dependency, precedence, etc.}\}\\ \text{SCOPE}\{\textit{global, local}\}\\ \text{POLARITY}\{\textit{plus, minus}\}\\ \text{CONSTRUCTION } \textit{contruction\_type}\\ \text{STRENGTH}\{\textit{hard, soft}\}\\ \text{ALIGNMENT}\langle \textit{alignment\_equations}\rangle\end{bmatrix}$$

The following feature structure shows an example of a global relation indicating that a verbal nucleus usually comes with a minor raising of the intonation (only main features are indicated here). This information is represented by an implication relation, which is oriented from the syntactic category to the prosodic phenomenon. Alignment equations stipulate a strict synchronization between object.

$$\text{relation}\begin{bmatrix}\text{INDEX}\\ \text{REL\_TYPE} \mid \text{ORIENTED\_REL}\begin{bmatrix}\text{SOURCE } VN_1\\ \text{TARGET } mr_2\end{bmatrix}\\ \text{TYPE}\{\textit{implication}\}\\ \text{STRENGTH}\{\textit{soft}\}\\ \text{ALIGNMENT}\langle lb_1{=}lb_2;\ rb_1{=}rb_2\rangle\end{bmatrix}$$

4 Representation with Hypergraphs

Nodes and relations can be combined and form higher level nodes, representing constructions which are a set of objects (the constituents) plus a set of relations between them. Such nodes are in fact *hypernodes* and bear two kinds of information: the properties characterizing the object plus a set of relations between the constituents (representing a subgraph). In the syntactic domain, for example, they represent phrases, as follows:

$$\begin{bmatrix}\text{DOMAIN } \textit{syntax}\\ \text{INDEX} \mid \text{LOCATION} \mid \text{TEMPORAL}\begin{bmatrix}\text{START } 122\\ \text{END } 584\end{bmatrix}\\ \text{FEATURES}\begin{bmatrix}\text{CAT } VP\end{bmatrix}\\ \text{RELATIONS}\left\{\begin{bmatrix}\text{INDEX } r_1\\ \text{REL\_TYPE} \mid \text{SET\_REL}\langle V, NP, Adv\rangle\\ \text{TYPE } \textit{constituency}\\ \text{STRENGTH } \textit{hard}\end{bmatrix};\right.\\ \left.\begin{bmatrix}\text{INDEX } r_2\\ \text{REL\_TYPE} \mid \text{ORIENTED\_REL}\begin{bmatrix}\text{SOURCE } NP\\ \text{TARGET } V\end{bmatrix}\\ \text{TYPE } \textit{dependency}\\ \text{STRENGTH } \textit{hard}\end{bmatrix}\right\}\end{bmatrix}$$

In the same way, the interaction between different objects from different domains can involve several relations. For example, a deictic construction can be made of the conjunction of an anaphoric pronoun, a deictic gesture and a physical object (for example a book on a shelf). Such a construction can be described by the following structure:

$$\begin{bmatrix}\text{INDEX} \mid \text{LOCATION} \mid \text{TEMPORAL}\begin{bmatrix}\text{START } 841\\ \text{END } 1520\end{bmatrix}\\ \text{FEATURES}\begin{bmatrix}\text{SEM } book'\end{bmatrix}\\ \text{RELATIONS}\left\{\begin{bmatrix}\text{INDEX } r_3\\ \text{SET\_REL}\langle Pro_1, Dx\_gest_2, Ph\_object_3\rangle\\ \text{TYPE } \textit{constituency}\\ \text{ALIGNMENT}\langle lb_1 \approx_\Delta lb_2;\ rb_1 \approx_\Delta rb_2\rangle\end{bmatrix};\right.\\ \left.\begin{bmatrix}\text{INDEX } r_4\\ \text{ORIENTED\_REL}\begin{bmatrix}\text{SOURCE } Pro_1\\ \text{TARGET } Ph\_object_3\end{bmatrix}\\ \text{TYPE } \textit{reference}\end{bmatrix}\right\}\end{bmatrix}$$

This construction indicates some properties (limited here to the semantic value) and two re-

lations between the different objects: one constituency, indicating the different objects involved in the construction and their (fuzzy) alignment and a reference relation between the pronoun and a physical object (here, a book).

This structure represents an hypergraph: it is a graph connecting different nodes, each of them being to its turn described by another graph, as shown above. The main interest of such a representation is its flexibility: all kinds of information can be described, at any level. Graphs being less constrained than trees, and edges (or relations) being typed, we can gather different levels, different domains and different granularities. For example, an agreement relation can be specified thanks to the deictic construction, besides the constituency one, making it possible to instanciate the agreement value of the pronoun.

Note that hypergraphs are also investigated in other knowledge representation, their properties are well known (Hayes, 2004) and the implementation of specific hypergraphs as the one presented here could be done in RDF graphs for example as suggested in (Cassidy, 2010).

5 Constraints for Multimodal Grammars

In the same way as typed feature structures can implement constraints and constitute a description language on linguistic structures (cf. HPSG,), the same approach can be generalized to multimodal information. SOme recent works have been done in this direction (see (Alahverdzhieva, 2010; **?**)). The representation we propose can implement generic information about multimodal constructions. We illustrate in the following this aspect with two phenomena: *backchannels* and *dislocation*.

Several studies on conversational data (see for example (Bertrand09)) have described backchannels (that can be vocal or gestual) and their context. They have in particular underline some regularities on the left context:

- backchannels usually follow: major intonative phrases (IP), flat contours, end of conversational turn (i.e. saturated from a semantic, syntactic and pragmatic point of view)
- backchannels never appear after connectives

These constraints can be implemented by means of a feature structure (representing an hypernode) with a set of precedence relations. The different objects involved in the description of the phenomenon (IP, flat contour, conversational turn, connective) are indicated with an indexed ID, referring to their complete feature structure, not presented here.

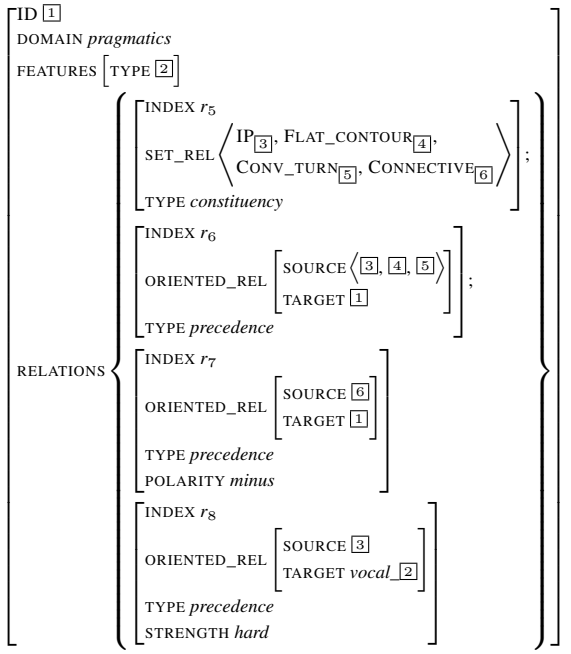


Figure 2: *Backchannel Constraint*

This structure (cf. Figure 2) represents a constraint that backchannels have to satisfy. The first relation specifies the constituents and their indexes, with which the different precedence constraints are represented. The relation *r6* indicates all kinds of object that should precede a backchannel. This constraint subsumes the most specific relation *r8* stipulating that a vocal backchannel is always preceded with an *IP* (this is a *hard* constraint). The relation *r7* excludes the possibility for a backchannel to be preceded with a connective.

The second example (cf. Figure 3) proposes a constraint system describing dislocated structures. We propose in this description to distinguish two syntactic constituents that form the two parts of the dislocation: the dislocated phrase (called *S1*) and the sentence from which the phrase has been

extracted (called *S2*). Usually (even if not always), *S2* contains a clitic referring to *S1*. We note in the following this clitic with the notation *S2//Clit*. For readability reasons, we only present in this structure the relations.

This structure describes the case of a left dislocation (with *S1* preceding *S2*, the constraint being hard). In such cases, *S1* is usually realized with a minor raising contour. The constraint *r13* implements the anaphoric relation between the clitic and the dislocated element. Finally, the relation *r14* indicates an agreement relation between the clitic and *S1* and in particular the fact that the case has to be the same for both objects.

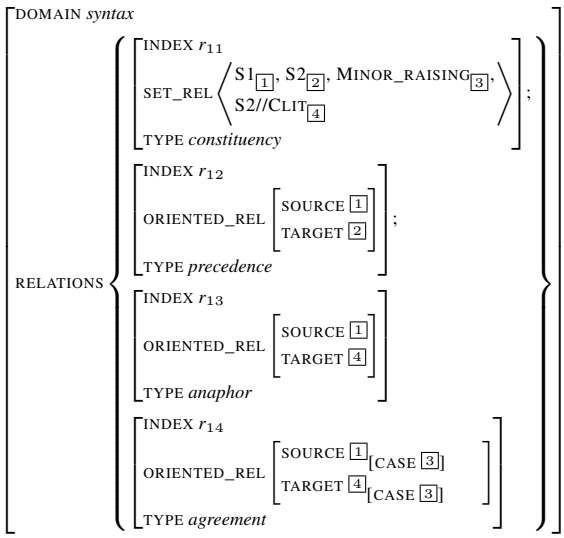


Figure 3: *Dislocation Constraint*

6 Conclusion

Linguistic annotation in general, and multimodality in particular, requires high level annotation schemes making it possible to represent in an homogeneous way information coming from the different domains and modalities involved in human communication.

The approach presented in this paper generalizes previous methods (in particular annotation graphs) thanks to two proposals: first in providing a way to index objects without strict order relation between nodes and second in specifying a precise and homogeneous representation of the objects and their relations. This approach has been developed into a formal scheme, *typed feature structures*, in which all the different domains can be represented, and making it possible to implement directly hypergraphs. TFS and hypergraphs are particularly well adapted for the specification of interaction constraints, describing interaction relations between modalities. Such constraints constitute the core of the definition of future multimodal grammars.

From a practical point of view, the proposal described in this paper is currently under experimentation within the OTIM project (see (Blache, 2010)). An XML scheme has been automatically generated starting from TFS formal scheme. The existing multimodal annotations, created with ad hoc annotation schemes, are to their turn automatically translated following this format. We obtain then, for the first time, a large annotated multimodal corpus, using an XML schema based on a formal specification.

References

Alahverdzhieva, K. and A. Lascarides (2010) "Analysing Language and Co-verbal Gesture and Constraint-based Grammars", in Proceedings of the *17th International Conference on Head-Driven Phase Structure Grammar*.

Allen F. and P. J. Hayes (1985) "A common-sense theory of time", in *9th International Joint Conference on Artificial Intelligence*.

Allwood J., L. Cerrato, L. Dybkjaer and al. (2005) The MUMIN Multimodal Coding Scheme, NorFA yearbook 2005

Bertrand R., M. Ader, P. Blache, G. Ferré, R. Espesser, S. Rauzy (2009) "Représentation, édition et exploitation de données multimodales : le cas des backchannels du corpus CID", in *Cahiers de linguistique française*, 33:2.

Blache P., R. Bertrand, and G. Ferré (2009) "Creating and Exploiting Multimodal Annotated Corpora: The ToMA Project". in Kipp, Martin, Paggio and Heylen (eds.) *Multimodal Corpora: From Models of Natural Interaction to Systems and Applications*, LNAI 5509, Springer.

Blache P. et al. (2010) "Multimodal Annotation of Conversational Data", in proceedings of *LAW-IV - The Linguistic Annotation Workshop*

Bird S., Day D., Garofolo J., Henderson J., Laprun C. & Liberman M. (2000) "ATLAS : A Flexible and Extensible Architecture for Linguistic Annotation", in procs of *LREC00*

Bird S., M. Liberman (2001) "A formal framework for linguistic annotation" *Speech Communication*, Elsevier

Boersma P. & D. Weenink (2009) *Praat*: doing phonetics by computer, http://www.praat.org/

Carletta, J., J. Kilgour, and T. O'Donnell (2003) "The NITE Object Model Library for Handling Structured Linguistic Annotation on Multimodal Data Sets" in procs of the *EACL Workshop on Language Technology and the Semantic Web*

Carpenter B. (1992) *The Logic of Typed Feature Structures*. Cambridge University Press.

Cassidy S. (2010) *An RDF Realisation of LAF in the DADA Annotation Server*. Proceedings of ISA-5, Hong Kong, January 2010.

Dipper S., M. Goetze and S. Skopeteas (eds.) (2007) *Information Structure in Cross-Linguistic Corpora: Annotation Guidelines for Phonology, Morphology, Syntax, Semantics and Information Structure*, Working Papers of the SFB 632, 7:07

Dybkjaer L., S. Berman, M. Kipp, M. Wegener Olsen, V. Pirrelli, N .Reithinger, C. Soria (2001) "Survey of Existing Tools, Standards and User Needs for Annotation of Natural Interaction and Multimodal Data", *ISLE Natural Interactivity and Multimodality Working Group Deliverable D11.1*

Fillmore C. & P. Kay (1996) *Construction Grammar*, Manuscript, University of California at Berkeley Department of linguistics.

Gruenstein A., J. Niekrasz, and M. Purver. (2008) "Meeting structure annotation: Annotations collected with a general purpose toolkit". In L. Dybkjaer and W. Minker, editors, *Recent Trends in Discourse and Dialogue*, Springer-Verlag.

Hayes J. and Gutierrez C. (2004) Bipartite graphs as intermediate model for RDF. Proceedings of ISWC 2004, 3rd International Semantic Web Conference (ISWC2004), Japan.

Ide N. and K. Suderman (2007) "GrAF: A Graph-based Format for Linguistic Annotations" in proceedings of the *Linguistic Annotation Workshop* (LAW-07)

Ide N. and Suderman K. (2009) Bridging the Gaps: Interoperability for GrAF, GATE, and UIMA. Proceedings of the Third Linguistic Annotation Workshop, held in conjunction with ACL 2009, Singapore.

Kipp M. (2001) "Anvil-a generic annotation tool for multimodal dialogue" in procs of 7th European Conference on Speech Communication and Technology

Kipp, M. (2003) *Gesture Generation by Immitation: From Human Behavior to Computer Character Animation*, PhD Thesis, Saarland University.

Lascarides, A. and M. Stone (2009) "A Formal Semantic Analysis of Gesture", in *Journal of Semantics*, 26(4).

McNeill, D. (2005) *Gesture and Thought*, The University of Chicago Press.

Pineda, L., and G. Garza (2000) "A Model for Multimodal Reference Resolution", in *Computational Linguistics*, Vol. 26 no. 2

Rodriguez K., Stefan, K. J., Dipper, S., Goetze, M., Poesio, M., Riccardi, G., Raymond, C., Wisniewska, J. (2007) "Standoff Coordination for Multi-Tool Annotation in a Dialogue Corpus", in procs of the *Linguistic Annotation Workshop at the ACL'07 (LAW-07)*

Wegener Knudsen M.and al. (2002) *Survey of Multimodal Coding Schemes and Best Practice*, ISLE

Wittenburg, P.; Brugman, H.; Russel, A.; Klassmann, A. and Sloetjes, H. (2006) "ELAN: a Professional Framework for Multimodality Research". In proceedings of LREC 2006

Composition of Semantic Relations: Model and Applications

Eduardo Blanco, **Hakki C. Cankaya** and **Dan Moldovan**
Human Language Technology Research Institute
The University of Texas at Dallas
{eduardo,candan,moldovan}@hlt.utdallas.edu

Abstract

This paper presents a framework for combining semantic relations extracted from text to reveal even more semantics that otherwise would be missed. A set of 26 relations is introduced, with their arguments defined on an ontology of sorts. A semantic parser is used to extract these relations from noun phrases and verb argument structures. The method was successfully used in two applications: rapid customization of semantic relations to arbitrary domains and recognizing entailments.

1 Introduction

Semantic representation of text facilitates inferences, reasoning, and greatly improves the performance of Question Answering, Information Extraction, Machine Translation and other NLP applications. Broadly speaking, semantic relations are unidirectional underlying connections between concepts. For example, the noun phrase *the car engine* encodes a PART-WHOLE relation: the engine is a part of the car.

Semantic relations are the building blocks for creating a semantic structure of a sentence. There is a growing interest in text semantics fueled by the new wave of semantic technologies and ontologies that aim at transforming unstructured text into structured knowledge. More and more enterprises and academic organizations have adopted the World Wide Web Consortium (W3C) Resource Description Framework (RDF) specification as a standard representation of text knowledge. This is based on semantic triples, which can be used to represent semantic relations.

The work reported in this paper aims at extracting as many semantic relations from text as possible. Semantic parsers (SP) extract semantic relations from text. Typically they detect relations between adjacent concepts or verb argument structures, leaving considerable semantics unrevealed. For example, given *John is a rich man*, a typical SP extracts *John is a man* and *man is rich*, but not *John is rich*. The third relation can be extracted by combining the two relations detected by the parser. The observation that combining elementary semantic relations yields more relations is the starting point and the motivation for this work.

2 Related Work

In Computational Linguistics, WordNet (Miller, 1995), FrameNet (Baker et al., 1998) and PropBank (Palmer et al., 2005) are probably the most used semantic resources. Like our approach and unlike PropBank, FrameNet annotates semantics between concepts regardless of their position in a parse tree. Unlike us, they use a predefined set of frames to be filled. PropBank adds semantic annotation on top of the Penn TreeBank and it contains only annotations between a verb and its arguments. Moreover, the semantics of a given label depends on the verb. For example, ARG2 is used for INSTRUMENT and VALUE.

Copious work has been done lately on semantic roles (Màrquez et al., 2008). Approaches to detect semantic relations usually focus on particular lexical and syntactic patterns or kind of arguments. There are both unsupervised (Turney, 2006) and supervised approaches. The SemEval-2007 Task 4 (Girju et al., 2007) focused on relations between nominals. Work has been done on detecting relations between noun phrases (Davidov and Rappoport, 2008; Moldovan et al., 2004), named entities (Hirano et al., 2007), and clauses (Szpakowicz et al., 1995). There have been pro-

Coling 2008: Poster Volume, pages 72–80,
Beijing, August 2010

posals to detect a particular relation, e.g., CAUSE (Chang and Choi, 2006), INTENT (Tatu, 2005) and PART-WHOLE (Girju et al., 2006).

Researchers have also worked on combining semantic relations. Harabagiu and Moldovan (1998) combine WordNet relations and Helbig (2005) transforms chains of relations into theoretical axioms. Some use logic as the underlying formalism (Lakoff, 1970; Sánchez Valencia, 1991), more ideas can be found in (Copestake et al., 2001).

3 Approach

In contrast to First Order Logic used in AI to represent text knowledge, we believe text semantics should be represented using a fixed set of relations. This facilitates a more standard representation and extraction automation which in turn allows reasoning. The fewer the relation types, the easier it is to reason and perform inferences. Thus, a compromise has to be made between having enough relation types to adequately represent text knowledge and yet keeping the number small for making the extraction and manipulation feasible.

The main contributions of this paper are: (i) an extended definition of a set of 26 semantic relations resulted after many iterations and pragmatic considerations; (ii) definition of a semantic calculus, a framework to manipulate and compose semantic relations (CSR); (iii) use of CSR to rapidly customize a set of semantic relations; and (iv) use of CSR to detect entailments. The adoption of CSR to other semantic projects does not require any modification of existing tools while being able to detect relations ignored by such tools.

4 Semantic Relations

Formally, a semantic relation is represented as R(x, y), where R is the relation type and x and y the first and second argument. R(x, y) should be read as *x is* R *of y*. The sentence *"John painted his truck"* yields AGENT(*John*, *painted*), THEME(*his truck*, *painted*) and POSSESSION(*truck*, *John*).

Extended definition Given a semantic relation R, DOMAIN(R) and RANGE(R) are defined as the set of sorts of concepts that can be part of the first and second argument. A semantic relation R(x, y) is defined by its: (i) relation type R; (ii) DOMAIN(R); and (iii) RANGE(R). Stating restrictions for DOMAIN and RANGE has several advantages: it (i) helps distinguishing between relations, e.g., $[\text{tall}]_{ql}$ and $[\text{John}]_{aco}$ can be linked through VALUE, but not POSSESSION; (ii) helps discarding potential relations that do not hold, e.g., abstract objects do not have INTENT; and (iii) helps combining semantic relations (Section 5).

Ontology of Sorts In order to define DOMAIN(R) and RANGE(R), we use a customized ontology of sorts (Figure 1) modified from (Helbig, 2005). The root corresponds to entities, which refers to *all things about which something can be said*.

Objects can be either concrete or abstract. The former occupy space, are touchable and tangible. The latter are intangible; they are somehow a product of human reasoning. Concrete objects are further divided into animate or inanimate. The former have life, vigor or spirit; the later are dull, without life. Abstract objects are divided into temporal or non temporal. The first corresponds to abstractions regarding points or periods of time (e.g. *July*, *last week*); the second to any other abstraction (e.g. *disease*, *justice*). Abstract objects can be sensually perceived, e.g., *pain*, *odor*.

Situations are anything that happens at a time and place. Simply put, if one can think of the time and location of an entity, it is a situation. Events (e.g. *mix*, *grow*) imply a change in the status of other entities, states (e.g. *standing next to the door*) do not. Situations can be expressed by verbs (e.g. *move*, *print*) or nouns (e.g. *party*, *hurricane*).

Descriptors complement entities by stating properties about their spatial or temporal context. They are composed of an optional non-content word signaling the local or temporal context and another entity. Local descriptors are further composed of a concrete object or situation, e.g., $[\textit{above}]_{prep}$ $[\textit{the roof}]_{co}$; temporal descriptors by a temporal abstract object or situation, e.g., $[\textit{during}]_{prep}$ $[\textit{the party}]_{ev}$. The non-content word signaling the local or temporal context is usually present, but not always, e.g., *"The* $[\textit{birthplace}]_{ev}$ *of his mother is* $[\textit{Ankara}]_{loc}$*"*.

Qualities represent characteristics than can be assigned to entities. They can be quantifiable like *tall* and *heavy*, or unquantifiable like *difficult* and *sleepy*. Quantities represent quantitative characteristics of concepts, e.g., *a few pounds*, *22 yards*.

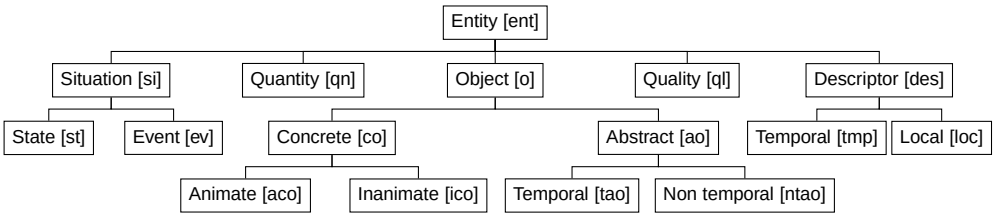


Figure 1: The ontology of sorts of concepts and their acronyms.

Cluster	Relation type	Abr.	Class.	Properties r	s	t	Domain × Range	Example
Reason	CAUSE	CAU	iv	-	-	√	[si] × [si]	CAU(*earthquake, tsunami*)
	JUSTIFICATION	JST	iv	-	-	√	[si ∪ ntao] × [si]	JST(*it is forbidden, don't smoke*)
	INFLUENCE	IFL	iv	-	-	√	[si] × [si]	IFL(*missing classes, poor grade*)
Goal	INTENT	INT	i	-	-	-	[si] × [aco]	INT(*teach, professor*)
	PURPOSE	PRP	v	-	-	√	[si ∪ ntao] × [si ∪ co ∪ ntao]	PRP(*storage, garage*)
Object modifiers	VALUE	VAL	v	-	-	-	[ql] × [o ∪ si]	VAL(*smart, kids*)
	SOURCE	SRC	ii	-	-	√	[loc ∪ ql ∪ ntao ∪ ico] × [o]	SRC(*Mexican, students*)
Syntactic subjects	AGENT	AGT	iii	-	-	-	[aco] × [si]	AGT(*John, bought*)
	EXPERIENCER	EXP	iii	-	-	-	[o] × [si]	EXP(*John, heard*)
	INSTRUMENT	INS	iii	-	-	-	[co ∪ ntao] × [si]	INS(*the hammer, broke*)
Direct objects	THEME	THM	iii	-	-	-	[o] × [ev]	THM(*a car, bought*)
	TOPIC	TPC	iii	-	-	-	[o ∪ si] × [ev]	TPC(*flowers, gave*)
	STIMULUS	STI	iii	-	-	-	[o] × [ev]	STI(*the train, heard*)
Association	ASSOCIATION	ASO	v	√	√	√	[ent] × [ent]	ASO(*fork, knife*)
	KINSHIP	KIN	ii	√	√	√	[aco] × [aco]	KIN(*John, his wife*)
None	IS-A	ISA	ii	-	-	√	[o] × [o]	ISA(*gas guzzler, car*)
	PART-WHOLE	PW	ii	-	-	*	[o] × [o] ∪ [l] × [l] ∪ [t] × [t]	PW(*engine, car*)
	MAKE	MAK	ii	-	-	-	[co ∪ ntao] × [co ∪ ntao]	MAK(*cars, BMW*)
	POSSESSION	POS	ii	-	-	√	[co] × [co]	POS(*Ford F-150, John*)
	MANNER	MNR	iii	-	-	-	[ql ∪ st ∪ ntao] × [si]	MNR(*quick, delivery*)
	RECIPIENT	RCP	iii	-	-	-	[co] × [ev]	RCP(*Mary, gave*)
	SYNONYMY	SYN	v	√	√	√	[ent] × [ent]	SYN(*a dozen, twelve*)
	AT-LOCATION	AT-L	v	√	-	*	[o ∪ si] × [loc]	AT-L(*party, John's house*)
	AT-TIME	AT-T	v	√	-	*	[o ∪ si] × [tmp]	AT-T(*party, last Saturday*)
	PROPERTY	PRO	v	-	-	-	[ntao] × [o ∪ si]	PRO(*height, John*)
	QUANTIFICATION	QNT	v	-	-	-	[qn] × [si ∪ o]	QNT(*a dozen, eggs*)

Table 1: The set of 26 relations clustered and classified with their properties (reflexive, symmetric, transitive) and examples. An asterisk indicates that the property holds under certain conditions.

4.1 Semantic Relation Types

This work focuses on the set of 26 semantic relations depicted in Table 1. We found this set specific enough to capture the most frequent semantics of text without bringing unnecessary overspecialization. The set is inspired by several previous proposals. Fillmore introduced the notion of *case frames* and proposed a set of nine roles: AGENT, EXPERIENCER, INSTRUMENT, OBJECT, SOURCE, GOAL, LOCATION, TIME and PATH (Fillmore, 1971). Fillmore's work was extended to FrameNet (Baker et al., 1998). PropBank (Palmer et al., 2005) annotates a set of 17 semantic roles in a per-verb basis.

We aim to encode relations not only between a verb and its arguments, but also between and within noun phrases and adjective phrases. Therefore, more relations are added to the set. It includes relations present in WordNet (Miller, 1995), such as IS-A, PART-WHOLE and CAUSE. Szpakowicz et al. (1995) proposed a set of nine relations and Turney (2006) a set of five. Rosario and Hearst (2004) proposed a set of 38 relations including standard case roles and a set of specific relations for medical domain. Helbig (2005) proposed a set of 89 relations, including ANTONYMY and several TEMPORAL relations, e.g. SUCCESSION, EXTENSION, END.

Our set clusters some of the previous proposals (e.g. we only consider AT-TIME) and discards relations proposed elsewhere when they did not occur frequently enough in our experiments. For example, even though ANTONYMY and ENTAILMENT are semantically grounded, they are very infrequent and we do not deal with them. Our pragmatic goal is to capture as many semantics as possible with as few relations as possible. How-

ever, we show (Section 7.1) that our set can be easily customized to a specific domain.

The 26 relations are clustered such that relations belonging to the same cluster are close in meaning. Working with clusters is useful because it allows us to: (i) map to other proposed relations, justifying the chosen set of relations; (ii) work with different levels of specificity; and (iii) reason with the relations in a per cluster basis.

The reason cluster includes relations between a concept having a direct impact on another. CAU(*x*, *y*) holds if *y* would not hold if *x* did not happen. JST(*x*, *y*) encodes a moral cause, motive or socially convened norm. If IFL(*x*, *y*), *x* affects the intensity of *y*, but it does not affect its occurrence.

The goal cluster includes INT and PRP. INT(*x*, *y*) encodes intended consequences, which are volitional. PRP(*x*, *y*) is a broader relation and can be defined for inanimate objects.

The object modifiers cluster encodes descriptions of objects and situations: SRC(*x*, *y*) holds if *x* expresses the origin of *y*. VAL(*x*, *y*) holds for any other attribute, e.g. *heavy*, *handsome*.

The syntactic subjects cluster includes relations linking a syntactic subject and a situation. The differences rely on the characteristics of the subject and the connection per se. AGT(*x*, *y*) encodes an intentional doer, *x* must be volitional. If EXP(*x*, *y*), *x* does not change the situation, it only experiences *y*; it does not participate intentionally in *y* either. If INS(*x*, *y*), *x* is used to perform *y*, *x* is a tool or device that facilitates *y*.

The direct objects cluster includes relations encoding syntactic direct objects. THM(*x*, *y*) holds if *x* is affected or directly involved by *y*. TPC(*x*, *y*) holds if *y* is a communication verb, like *talk* and *argue*. STI(*x*, *y*) holds if *y* is a perception verb and *x* a stimulus that makes *y* happen.

The association cluster includes ASO and KIN. ASO is a broad relation between any pair of entities; KIN encodes a relation between relatives.

The rest of the relations do not fall into any cluster. ISA, PW, SYN, AT-L and AT-T have been widely studied in the literature. MAK(*x*, *y*) holds if *y* makes or produces *x*; POS(*x*, *y*) holds if *y* owns *x*; MNR encodes the way a situation occurs. RCP captures the connection between an event and an object which is the receiver of the event. PRO describes links between a situation or object and its characteristics, e.g., *height*, *age*. Values to the characteristics are given through VAL. QNT(*x*, *y*) holds if *y* is quantitatively determined by *x*.

Relations can also be classified depending on the kind of concepts they describe and their *intra* or *inter* nature into: (i) Intra-Object; (ii) Inter-Objects; (iii) Intra-Situation; (iv) Inter-Situations; and (v) for Object and Situation description.

4.2 Detection of Semantic Relations

Relations are extracted by an in-house SP from a wide variety of syntactic realizations. For example, the compound nominal *steel knife* contains PW(*steel*, *knife*), whereas *carving knife* contains PRP(*carving*, *knife*); the genitive *Mary's toy* contains POS(*toy*, *Mary*), whereas *Mary's brother* contains KIN(*brother*, *Mary*), and *eyes of the baby* contains a PW(*eyes*, *baby*). Relations are also extracted from a verb and its arguments (NP verb, verb NP, verb PP, verb ADVP and verb S), adjective phrases and adjective clauses.

The SP first uses a combination of state-of-the-art text processing tools, namely, part-of-speech tagging, named entity recognition, syntactic parsing and word sense disambiguation. After a candidate syntactic pattern has been found, a series of machine learning classifiers are applied to decide if a relation holds. Four different algorithms are used: decision trees, Naive Bayes, SVM and Semantic Scattering combined in a hybrid approach. Some algorithms use a per-relation approach (i.e., decide whether or not a given relation holds) and others a per-pattern approach (i.e., which relation, if any, holds for a particular pattern). Additionally, human-coded rules are used for a few unambiguous cases. The SP participated in the SemEval 2007 Task 4 (Badulescu and Srikanth, 2007).

5 Composition of Semantic Relations

The goal of semantic calculus (SC) is to provide a formal framework for manipulating semantic relations. CSR is a part of this, its goal is to apply *inference axioms* over already identified relations in text in order to infer more relations.

Semantic Calculus: Operators and Properties The *composition operator* is represented by the

$(\mathrm{R}^{-1})^{-1} = \mathrm{R}$
$\mathrm{R}_i \circ \mathrm{R}_j = (\mathrm{R}_j{}^{-1} \circ \mathrm{R}_i{}^{-1})^{-1}$
R^{-1} inherits all the properties of R
$\perp^{-1} = \perp$
$\forall i$: $\perp \bowtie \mathrm{R}_i$
R is reflexive iff $\forall x$: $\mathrm{R}(x, x)$
R is symmetric iff $\mathrm{R}(x, y) = \mathrm{R}(y, x)$
R is transitive iff $\mathrm{R}(x, y) \circ \mathrm{R}(y, z) \rightarrow \mathrm{R}(x, z)$
$\mathrm{R}_i \triangleright \mathrm{R}_j \leftrightarrow \mathrm{R}_i{}^{-1} \triangleleft \mathrm{R}_j{}^{-1}$
$\mathrm{R}_i \bowtie \mathrm{R}_j \leftrightarrow \mathrm{R}_i{}^{-1} \bowtie \mathrm{R}_j{}^{-1}$
If R_i is symmetric and $\mathrm{R}_i \bowtie \mathrm{R}_j$, $\mathrm{R}_i{}^{-1} \bowtie \mathrm{R}_j$
If R_j is symmetric and $\mathrm{R}_i \bowtie \mathrm{R}_j$, $\mathrm{R}_i \bowtie \mathrm{R}_j{}^{-1}$

Table 2: Semantic calculus properties

symbol $\circ$. It combines two relations and yields a third one. Formally, we denote $\mathrm{R}_1 \circ \mathrm{R}_2 \rightarrow \mathrm{R}_3$.

The *inverse* of R is denoted R^{-1} and can be obtained by simply switching its arguments. Given $\mathrm{R}(x, y)$, $\mathrm{R}^{-1}(y, x)$ always holds. The easiest way to read $\mathrm{R}^{-1}(y, x)$ is *x is* R *of y*.

R_1 *left dominates* R_2, denoted by $\mathrm{R}_1 \triangleright \mathrm{R}_2$, iff the composition of R_1 and R_2 yields R_1, i.e., $\mathrm{R}_1 \triangleright \mathrm{R}_2$ iff $\mathrm{R}_1 \circ \mathrm{R}_2 \rightarrow \mathrm{R}_1$. R_1 *right dominates* R_2, denoted by $\mathrm{R}_1 \triangleleft \mathrm{R}_2$, iff the composition of R_2 and R_1 yields R_1, i.e., $\mathrm{R}_1 \triangleleft \mathrm{R}_2$ iff $\mathrm{R}_2 \circ \mathrm{R}_1 \rightarrow \mathrm{R}_1$. R_1 *completely dominates* R_2, denoted by $\mathrm{R}_1 \bowtie \mathrm{R}_2$, iff $\mathrm{R}_1 \triangleright \mathrm{R}_2$ and $\mathrm{R}_1 \triangleleft \mathrm{R}_2$, i.e., $\mathrm{R}_1 \bowtie \mathrm{R}_2$ iff $\mathrm{R}_1 \circ \mathrm{R}_2 \rightarrow \mathrm{R}_1$ and $\mathrm{R}_2 \circ \mathrm{R}_1 \rightarrow \mathrm{R}_1$.

An OTHER ($\perp$) relation holds between *x* and *y* if no relation from the given set holds. Formally, $\perp(x, y)$ iff $\neg\exists \mathrm{R}_i$ such that $\mathrm{R}_i(x, y)$.

Using the notation above, the properties depicted in Table 2 follow.

Necessary conditions for Combining Relations Axioms can be defined only for compatible relations as premises. R_1 and R_2 are *compatible* if it is possible, from a theoretical point of view, to apply the composition operator to them. Formally, $\mathrm{RANGE}(\mathrm{R}_1) \cap \mathrm{DOMAIN}(\mathrm{R}_2) \neq \emptyset$

If R_1 and R_2 are compatible but not equal a *restriction* occurs. Let us denote $\mathrm{RANGE}(\mathrm{R}_1) \cap \mathrm{DOMAIN}(\mathrm{R}_2) = I$. A *backward* restriction takes place if $\mathrm{RANGE}(\mathrm{R}_1) \neq I$ and a *forward* restriction if $\mathrm{DOMAIN}(\mathrm{R}_2) \neq I$. In the former case $\mathrm{RANGE}(\mathrm{R}_1)$ is reduced; in the later $\mathrm{DOMAIN}(\mathrm{R}_2)$ is reduced. A forward and backward restriction can be found with the same pair of relations.

It is important to note that two compatible relations may not be the premises for a valid axiom. For example, KIN and AT-L are compatible but do not yield any valid inference.

Another necessary condition for combining two relations $\mathrm{R}_1(x, y)$ and $\mathrm{R}_2(y, z)$ is that they have to have a common argument, *y*.

5.1 Unique axioms

An axiom is defined as a set of relations called premises and a conclusion. Given the premises it unequivocally yields a relation that holds as conclusion. The composition operator is the basic way of combining two relations to form an axiom.

In general, for n relations there are $\binom{n}{2} = \frac{n(n-1)}{2}$ different pairs. For each pair, taking into account the two relations and their inverses, there are $4 \times 4 = 16$ different possible combinations. Applying property $\mathrm{R}_i \circ \mathrm{R}_j = (\mathrm{R}_j{}^{-1} \circ \mathrm{R}_i{}^{-1})^{-1}$, only 10 combinations are unique: (i) 4 combine R_1, R_2 and their inverses; (ii) 3 combine R_1 and $\mathrm{R}_1{}^{-1}$; and (iii) 3 combine R_2 and $\mathrm{R}_2{}^{-1}$. The most interesting axioms fall into category (i), since the other two can be resolved by the transitivity property of a relation and its inverse.

For n relations there are $2n^2 + n$ potential axioms: $\binom{n}{2} \times 4 + 3n = 2 \times n(n-1) + 3n = 2n^2 + n$. For $n = 26$, there are 1300 potential axioms in (i), 820 of which are compatible.

The number can be further reduced. After manual examination of combinations of ASO and KIN with other relations, we conclude that they do not yield any valid inferences, invalidating 150 potential axioms. This is due to the broad meaning of these relations. QNT can be discarded as well, invalidating 45 more potential axioms.

Some axioms can be easily validated. Because synonymous concepts are interchangeable, SYN is easily combined with any other relation: $\mathrm{SYN}(x, y) \circ \mathrm{R}(y, z) \rightarrow \mathrm{R}(x, z)$ and $\mathrm{R}(x, y) \circ \mathrm{SYN}(y, z) \rightarrow \mathrm{R}(x, z)$. Because hyponyms inherit relations from their hypernyms, $\mathrm{ISA}(x, y) \circ \mathrm{R}(y, z) \rightarrow \mathrm{R}(x, z)$ and $\mathrm{R}(x, y) \circ \mathrm{ISA}^{-1}(y, z) \rightarrow \mathrm{R}(x, z)$ hold. These observations allow us to validate 138 of the 625 potential axioms left, still leaving 487.

As noted before, relations belonging to the same cluster tend to behave similarly. This is especially true for the reason and goal clusters due to their semantic motivation. Working with these two clusters instead of the relations brings the

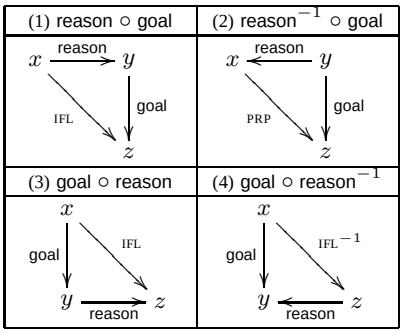


Table 3: The four axioms taking as premises reason and goal clusters. Diagonal arrows indicate inferred relations.

number of axioms to be examined down to 370.

Out of the 370 axioms left, we have extensively analyzed and defined the 35 involving AT-L, the 43 involving reason and the 58 involving goal. Because of space constraints, in this paper we only fully introduce the axioms for reason and goal (Section 6), as well as a variety of axioms useful to recognize textual entailments (Section 7.2).

6 Case Study: Reason and Goal

In this section, we present the four unique axioms for reason and goal relations (Table 3).

(1) **REA(*x, y*) ∘ GOA(*y, z*) → IFL(*x, z*)**: an event is influenced by the reason of its goal.

For example: Bill saves money because he is unemployed; he spends far less than he used to. Therefore, being unemployed can lead to spend far less.

P REA(*be unemployed, save money*)
GOA(*save money, spend far less*)
C IFL(*be unemployed, spend far less*)

(2) **REA^{-1}(*x, y*) ∘ GOA(*y, z*) → PRP(*x, z*)**: events have as their purpose the effects of their goals. This is a strong relation.

For example: Since they have a better view, they can see the mountain range. They cut the tree to have a better view. Therefore, they cut the tree to see the mountain range.

P REA^{-1}(*see the mountain range, better view*)
GOA(*better view, cut the tree*)
C PRP(*see the mountain range, cut the tree*)

Note that possible unintended effects of cutting the tree (e.g. homeowners' association complains) are caused by the event *cut the tree*, not by its effect *get a better view*.

(3) **GOA(*x, y*) ∘ REA(*y, z*) → IFL(*x, z*)**: the goal of an action influences its effects.

For example: John crossed the street carelessly to get there faster. He got run over by a propane truck. Therefore, John got run over by a propane truck influenced by (having the goal of) getting there faster.

P GOA(*get there faster, crossed carelessly*)
REA(*crossed carelessly, got run over*)
C IFL(*get there faster, got run over*)

(4) **GOA(*x, y*) ∘ REA^{-1}(*y, z*) → IFL^{-1}(*x, z*)**. Events influence the goals of its effects.

For example: Jane exercises to lose weight. She exercised because of the good weather. Therefore, good weather helps to lose weight.

P GOA(*lose weight, exercise*)
REA^{-1}(*exercise, good weather*)
C IFL^{-1}(*lose weight, good weather*)

The axioms have been evaluated using manually annotated data. PropBank CAU and PNC are used as reason and goal. Reason annotation is further collected from a corpus which adds causal annotation to the Penn TreeBank (Bethard et al., 2008). A total of 5 and 29 instances for axioms 3 and 4 were found. For all of them, the axioms yield a valid inference. For example, *Buick [approached]$_y$ American express about [a joint promotion]$_x$ because [its card holders generally have a good credit history]$_z$*. PropBank annotation states GOA(x, y) and REA^{-1}(y, z), axiom 4 makes the implicit relation IFL^{-1}(x, z) explicit.

7 Applications and Results

7.1 Customization of Semantic Relations

Problem There is no agreement on a set of relations that best represent text semantics. This is rightfully so since different applications and domains call for different relations. CSR can be used to rapidly customize a set of relations without having to train a new SP or modify any other tool. Given a text, the SP extracts 26 elementary semantic relations. Axioms within the framework of CSR yield n new relations, resulting in a richer semantic representation (Figure 2).

CSR axioms Two ways to get new relations are:

(i) Direct mapping. This is the easiest case and it is equivalent to rename a relation. For example, we can map POS to BELONG or IS-OWNER-OF.

Axiom	Rest. on y	Example
AGT(x, y) ∘ THM^{-1}(y, z) → ARRESTED(x, z)	*arrested* concept	[Police]$_x$ [apprehended]$_y$ 51 [football fans]$_z$.
THM(x, y) ∘ AT-L(y, z) → ARRESTED-AT(x, z)	*arrested* concept	Police [apprehended]$_y$ 51 [fans]$_x$ [near the Dome]$_z$.
AGT(x, y) ∘ AT-L(y, z) → BANKS-AT(x, z)	*banking* activity	[John]$_x$ [withdrew]$_y$ $20 [at the nearest Chase]$_z$.
POS(x, y) ∘ AT-L(y, z) → BANKS-AT(x, z)	*account* concept	[John]$_x$ got a [checkbook]$_y$ at [Chase]$_z$.

Table 4: Examples of semantic relation customization using CSR.

Pair	Text T		Hypothesis H	
113	Belknap married and lost his first two wives, Cora LeRoy and Carrie Tomlinson, and married Mrs. John Bower, his second wife's sister.		Belknap was married to Carrie Tomlinson.	
	$T1$	AGT(*Belknap, married*)	$H1$	AGT(*Belknap, was married*)
	$T2$	THM(*wives, married*)	$H2$	THM(*Carrie Tomlinson, was married*)
	$T3$	QNT(*first two, wives*)		
	$T4$	ISA(*Carrie Tomlinson, wives*)		
429	India's yearly pilgrimage to the Ganges river, worshiped by Hindus as the goddess Ganga, is the world's largest gathering of people, ...		Ganga is a Hindu goddess.	
	$T1$	AGT(*Hindus, worship*)	$H1$	ISA(*Ganga, goddess*)
	$T2$	THM(*Ganga, worship*)	$H2$	VAL(*Hindu, goddess*)
	$T3$	ISA(*Ganga, goddess*)		
445	[...] At present day YouTube represents the most popular site sharing on-line video.		YouTube is a video website.	
	$T1$	ISA(*YouTube, site*)	$H1$	ISA(*YouTube, website*)
	$T2$	EXP(*site, sharing*)	$H2$	VAL(*video, website*)
	$T3$	THM(*video, sharing*)		
716	The Czech and Slovak republics have been unable to agree a political basis for their future coexistence in one country.		The Czech and Slovak republics do not agree to coexist in one country.	
	$T1$	AGT(*The Czech and Slovak republics, have been unable to agree*)	$H1$	AGT(*The Czech and Slovak republics, do not agree*)
	$T2$	THM(*political basis, have been unable to agree*)	$H2$	PRP(*coexist in one country, do not agree*)
	$T3$	PRP(*their future coexistence in one country, political basis*)		
771	In 2003, Yunus brought the microcredit revolution to the streets of Bangladesh to support more than 50,000 beggars, whom the Grameen Bank respectfully calls Struggling Members.		Yunus supported more than 50,000 Struggling Members.	
	$T1$	AGT(*Yunus, brought*)	$H1$	AGT(*Yunus, supported*)
	$T2$	PRP(*support, brought*)		
	$T3$	RCP(*beggars, support*)	$H2$	RCP(*Struggling Members, support*)
	$T4$	QNT(*more than 50,000, beggars*)	$H3$	QNT(*more than 50,000, Struggling Members*)
	$T5$	SYN(*beggars, Struggling Members*)		

Table 5: RTE3 examples and their elementary semantic relations (i.e., the ones the SP detects). Only relevant semantic relations for entailment detection are shown for T.

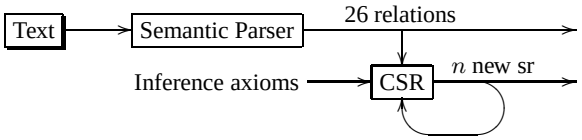


Figure 2: Flowchart for obtaining customized semantic relations using CSR.

(ii) Combinations of two elementary relations yield new specialized relations. In this case, restrictions on the arguments must be fulfilled.

Consider we need the new relation ARRESTED(x, y), which encodes the relation between two animate concrete objects x and y, where x arrested y. We can infer this relation by using the following axiom: AGENT(x, y) ∘ THEME^{-1}(y, z) → ARRESTED(x, z) provided that y is an *arrested* concept. A simple way of checking if a given concept is of a certain kind is to check WordNet. Collecting all the words belonging the the synset arrest.v.1, we get the following list of *arrested* concepts: *collar*, *nail*, *apprehend*, *pick up*, *nab* and *cop*. Using lexical chains the list could be further improved.

More examples of axioms for generating customized semantic relations are shown in Table 4.

Results Virtually any domain could be covered by applying customization over the set of 26 relations. The set has been successfully customized to a law enforcement domain. Ax-

ioms for a total of 37 new relations were defined and implemented. Among others, axioms to infer IS-EMPLOYER, IS-COWORKER, IS-PARAMOUR, IS-INTERPRETER, WAS-ASSASSIN, ATTENDS-SCHOOL-AT, JAILED-AT, COHABITS-WITH, AFFILIATED-TO, MARRIED-TO, RENTED-BY, KIDNAPPED-BY and the relations in Table 4 were defined. Note that a relation can be inferred by several axioms. This customization effort to add 37 new specialized relations took a person only a few days and without modifying the SP.

7.2 Textual Entailment

Problem An application of CSR is recognizing entailments. Given text T and hypothesis H, the task consists on determining whether or not H can be inferred by T (Giampiccolo et al., 2007).

CSR axioms Several examples of the RTE3 challenge can be solved by applying CSR (Table 5). The rest of this section depicts the axioms involved in detecting entailment for each pair.

Pair 113 is a simple one. A perfect match for H in T can be obtained by an axiom reading *all concepts inherit the semantic relations of their hypernyms*. Formally, ISA(x, y) $\circ$ THM(y, z) $\rightarrow$ THM(x, z), $T2$ and $T4$ are the premises and the conclusion matches $H2$. $T1$ matches $H1$.

Pair 429 can be solved by an axiom reading *agents are values for their themes*. Formally, AGT(x, y) $\circ$ THM^{-1}(y, z) $\rightarrow$ VAL(x, z); $T1$ and $T2$ yield VAL(*Hindu, Ganga*), which combined with $T3$ results in a match between T and H.

Pair 445 follows a similar pattern, but the way an EXP combines with its THM differs from the way an AGT does. The *theme is a value of the experiencer*, THM(x, y) $\circ$ EXP^{-1}(y, z) $\rightarrow$ VAL(x, z). Given $T2$ and $T3$, the axiom yields $T4$: VAL(*video, site*). Assuming that SYN(*site, website*), $T1$ and $T4$ match H.

Pair 716 also requires only one inference step. Using $T3$ and $T2$, an axiom reading *situations have as their purpose the purpose of its theme* infers $H2$, yielding a perfect match between T and H. Formally, PRP(x, y) $\circ$ THM(y, z) $\rightarrow$ PRP(x, z).

Pair 771 Using as premises $T1$ and $T2$, an axiom reading *an agent performs the purposes of its actions* infers $H1$. Using $T3$ and $T5$, and $T4$ and $T5$ as premises, an axiom reading *synonymous concepts are interchangeable* infers $H2$ and $H3$, resulting in a perfect match between T and H. Formally, AGT(x, y) $\circ$ PRP^{-1}(y, z) $\rightarrow$ AGT(x, z), RCP^{-1}(x, y) $\circ$ SYN(y, z) $\rightarrow$ RCP^{-1}(x, z) and QNT(x, y) $\circ$ SYN(y, z) $\rightarrow$ QNT(x, z).

Results We conducted two experiments to quantify the impact of CSR in detecting entailments.

First, 60 pairs were randomly selected from the RTE3 challenge and parsed with the SP. 14 of them (23%) could be solved by simply matching the elementary relations in T and H. After applying CSR, 21 more pairs (35%) were solved. Thus, adding CSR on top of the SP clearly improves entailment detection. Out of the 25 pairs not solved, 5 (8%) need coreference resolution and 20 (34%) require commonsense knowledge or fairly complicated reasoning methods (e.g. *a shipwreck is a ship that sank*).

CSR has also been added to a state of the art system for detecting textual entailment (Tatu and Moldovan, 2007). Prior to the addition, the system made 222 errors consisting of 46 false negatives (examples in Table 5) and 176 false positives. CSR was able to correctly solve 18 (39%) of the 46 false negatives.

8 Conclusions

Although the idea of chaining semantic relations has been proposed before, this paper provides a formal framework establishing necessary conditions for composition of semantic relations. The CSR presented here can be used to rapidly customize a set of relations to any arbitrary domain. In addition to the customization of an information extraction tool and recognizing textual entailments, CSR has the potential to contribute to other applications. For example, it can help improve a semantic parser, it can be used to acquire commonsense knowledge axioms and more.

When an axiom that results from combining two relations does not always hold, it may be possible to add constraints that limit the arguments of the premises to only some concepts.

This work stems from the need to automate the extraction of deep semantics from text and representing text as semantic triples. The paper demonstrates that CSR is able to extract more relations than a normal semantic parser would.

References

Badulescu, Adriana and Munirathnam Srikanth. 2007. LCC-SRN: LCC's SRN System for SemEval 2007 Task 4. In *Proceedings of the Fourth International Workshop on Semantic Evaluations*, pages 215–218.

Baker, Collin F., Charles J. Fillmore, and John B. Lowe. 1998. The Berkeley FrameNet Project. In *Proceedings of the 17th international conference on Computational Linguistics*, Montreal, Canada.

Bethard, Steven, William Corvey, Sara Klingenstein, and James H. Martin. 2008. Building a Corpus of Temporal-Causal Structure. In *Proceedings of the Sixth International Language Resources and Evaluation Conference*, Marrakech, Morocco.

Chang, Du S. and Key S. Choi. 2006. Incremental cue phrase learning and bootstrapping method for causality extraction using cue phrase and word pair probabilities. *Information Processing & Management*, 42(3):662–678.

Copestake, Ann, Alex Lascarides, and Dan Flickinger. 2001. An Algebra for Semantic Construction in Constraint-based Grammars. In *Proceedings of 39th Annual Meeting of the ACL*, pages 140–147.

Davidov, Dmitry and Ari Rappoport. 2008. Classification of Semantic Relationships between Nominals Using Pattern Clusters. In *Proceedings of ACL-08: HLT*, pages 227–235, Columbus, Ohio.

Fillmore, Charles J. 1971. Some Problems for Case Grammar. *Monograph Series on Languages and Linguistics*, 24:35–36.

Giampiccolo, Danilo, Bernardo Magnini, Ido Dagan, and Bill Dolan. 2007. The Third PASCAL Recognizing Textual Entailment Challenge. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 1–9.

Girju, Roxana, Adriana Badulescu, and Dan Moldovan. 2006. Automatic Discovery of Part-Whole Relations. *Computational Linguistics*, 32(1):83–135.

Girju, Roxana, Preslav Nakov, Vivi Nastase, Stan Szpakowicz, Peter Turney, and Deniz Yuret. 2007. SemEval-2007 Task 04: Classification of Semantic Relations between Nominals. In *Proceedings of the Fourth International Workshop on Semantic Evaluations*, pages 13–18, Prague, Czech Republic.

Harabagiu, Sanda and Dan Moldovan. 1998. Knowledge Processing on an Extended WordNet. In Fellbaum, Christiane, editor, *WordNet: An Electronic Lexical Database and Some of its Applications*, chapter 17, pages 684–714. The MIT Press.

Helbig, Hermann. 2005. *Knowledge Representation and the Semantics of Natural Language*. Springer.

Hirano, Toru, Yoshihiro Matsuo, and Genichiro Kikui. 2007. Detecting Semantic Relations between Named Entities in Text Using Contextual Features. In *Proceedings of the 45th Annual Meeting of the ACL, Demo and Poster Sessions*, pages 157–160.

Lakoff, George. 1970. Linguistics and Natural Logic. *Synthese*, 22(1):151–271.

Màrquez, Lluís, Xavier Carreras, Kenneth C. Litkowski, and Suzanne Stevenson. 2008. Semantic Role Labeling: An Introduction to the Special Issue. *Computational Linguistics*, 34(2):145–159.

Miller, George A. 1995. WordNet: A Lexical Database for English. *Communications of the ACM*, 38:39–41.

Moldovan, Dan, Adriana Badulescu, Marta Tatu, Daniel Antohe, and Roxana Girju. 2004. Models for the Semantic Classification of Noun Phrases. In *HLT-NAACL 2004: Workshop on Computational Lexical Semantics*, pages 60–67.

Palmer, Martha, Daniel Gildea, and Paul Kingsbury. 2005. The Proposition Bank: An Annotated Corpus of Semantic Roles. *Computational Linguistics*, 31(1):71–106.

Rosario, Barbara and Marti Hearst. 2004. Classifying Semantic Relations in Bioscience Texts. In *Proc. of the 42nd Meeting of the ACL*, pages 430–437.

Sánchez Valencia, Victor. 1991. *Studies on Natural Logic and Categorial Grammar*. Ph.D. thesis, University of Amsterdam.

Szpakowicz, Barker, Ken Barker, and Stan Szpakowicz. 1995. Interactive semantic analysis of Clause-Level Relationships. In *Proceedings of the Second Conference of the Pacific ACL*, pages 22–30.

Tatu, Marta and Dan Moldovan. 2007. COGEX at RTE 3. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 22–27, Prague, Czech Republic.

Tatu, Marta. 2005. Automatic Discovery of Intentions in Text and its Application to Question Answering. In *Proceedings of the ACL Student Research Workshop*, pages 31–36, Ann Arbor, Michigan.

Turney, Peter D. 2006. Expressing Implicit Semantic Relations without Supervision. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL*, pages 313–320, Sydney, Australia.

Improved Unsupervised Sentence Alignment for Symmetrical and Asymmetrical Parallel Corpora

Fabienne Braune **Alexander Fraser**
Institute for Natural Language Processing
Universität Stuttgart
{braunefe,fraser}@ims.uni-stuttgart.de

Abstract

We address the problem of unsupervised and language-pair independent alignment of symmetrical and asymmetrical parallel corpora. Asymmetrical parallel corpora contain a large proportion of 1-to-0/0-to-1 and 1-to-many/many-to-1 sentence correspondences. We have developed a novel approach which is fast and allows us to achieve high accuracy in terms of F_1 for the alignment of both asymmetrical and symmetrical parallel corpora. The source code of our aligner and the test sets are freely available.

1 Introduction

Sentence alignment is the problem of, given a parallel text, finding a bipartite graph matching minimal groups of sentences in one language to their translated counterparts. Because sentences do not always align 1-to-1, the sentence alignment task is non-trivial.

The achievement of high accuracy with minimal consumption of computational resources is a common requirement for sentence alignment approaches. However, in order to be applicable to parallel corpora in any language without requiring a separate training set, a method for sentence-alignment should also work in an unsupervised fashion and be language pair independent. By "unsupervised", we denote methods that infer the alignment model directly from the data set to be aligned. Language pair independence refers to approaches that require no specific knowledge about the languages of the parallel texts to align.

We have developed an approach to unsupervised and language-pair independent sentence alignment which allows us to achieve high accuracy in terms of F_1 for the alignment of both symmetrical and asymmetrical parallel corpora. Due to the incorporation of a novel two-pass search procedure with pruning, our approach is acceptably fast. Compared with Moore's bilingual sentence aligner (Moore, 2002), we obtain an average F_1 of 98.38 on symmetrical parallel documents, while Moore's aligner achieves 94.06. On asymmetrical documents, our approach achieves 97.67 F_1 while Moore's aligner obtains 88.70. On average, our sentence aligner is only about 4 times slower than Moore's aligner.

This paper is organized as follows: previous work is described in section 2. In section 3, we present our approach. Finally, in section 4, we conduct an extensive evaluation, including a brief insight into the impact of our aligner on the overall performance of an MT system.

2 Related Work

Among approaches that are unsupervised and language independent, (Brown et al., 1991) and (Gale and Church, 1993) use sentence-length statistics in order to model the relationship between groups of sentences that are translations of each other. As shown in (Chen, 1993) the accuracy of sentence-length based methods decreases drastically when aligning texts containing small deletions or free translations. In contrast, our approach augments a sentence-length based model with lexical statistics and hence constantly provides high quality alignments.

(Moore, 2002) proposes a multi-pass search

Coling 2008: Poster Volume, pages 81–89,
Beijing, August 2010

procedure where sentence-length based statistics are used in order to extract the training data for the IBM Model-1 translation tables. The acquired lexical statistics are then combined with the sentence-length based model in order to extract 1-to-1 correspondences with high accuracy[1]. Moore's approach constantly achieves high precision, is robust to sequences of inserted and deleted text, and is fast. However, the obtained recall is at most equal to the proportion of 1-to-1 correspondences contained in the parallel text to align. This point is especially problematic when aligning asymmetrical parallel corpora. In contrast, our approach allows to extract 1-to-many/many-to-1 correspondences. Hence, we achieve high accuracy in terms of precision and recall on both symmetrical and asymmetrical documents. Moreover, because we use, in the last pass of our multi-pass method, a novel two-stage search procedure, our aligner also requires acceptably low computational resources.

(Deng et al., 2006) have developed a multi-pass method similar to (Moore, 2002) but where the last pass is composed of two alignment procedures: a standard dynamic programming (DP) search that allows one to find many-to-many alignments containing a large amount of sentences in each language and a divisive clustering algorithm that optimally refines those alignments through iterative binary splitting. This alignment method allows one to find, in addition to 1-to-1 correspondences, high quality 1-to-many/many-to-1 alignments. However, 1-to-0 and 0-to-1 correspondences are not modeled in this approach[2]. This leads to poor performance on parallel texts containing that type of correspondence. Furthermore performing an exhaustive DP search in order to find large size many-to-many alignments involves high computational costs. In comparison to (Deng et al., 2006), our approach works in the opposite way. Our two-step search procedure first finds a model-optimal alignment composed of the smallest possible correspondences, namely 1-to-0/0-to-1 and 1-to-1, and then merges those correspondences into larger alignments. This allows the finding of 1-to-0/0-to-1 alignments as well as high quality 1-to-many/many-to-1 alignments, leading to high accuracy on parallel texts but also on corpora containing large blocs of inserted or deleted text. Furthermore, our approach keeps the computational costs of the alignment procedure low: our aligner is, on average, about 550 times faster than our implementation[3] of (Deng et al., 2006).

Many other approaches to sentence-alignment are either supervised or language dependent. The approaches by (Chen, 1993), (Ceausu et al., 2006) or (Fattah et al., 2007) need manually aligned pairs of sentences in order to train the used alignment models. The approaches by (Wu, 1994), (Haruno and Yamazaki, 1996), (Ma, 2006) and (Gautam and Sinha, 2007) require an externally supplied bilingual lexicon. Similarly, the approaches by (Simard and Plamondon, 1998) or (Melamed, 2000) are language pair dependent insofar as they are based on cognates.

3 Two-Step Clustering Approach

We present here our two-step clustering approach to sentence alignment[4] which is the main contribution of this paper. We begin by giving the main ideas of our approach using an introductory example (section 3.1). Then we show to which extent computational costs are reduced in comparison to a standard DP search (section 3.2) before presenting the theoretical background of our approach (section 3.3). We further discuss a novel pruning strategy used within our approach (section 3.4). This pruning technique is another important contribution of this paper. Next, we present the alignment model (section 3.5) which is a slightly modified version of the alignment model used in (Moore, 2002). Finally, we describe the overall

[1] The used search heuristic is a forward-backward computation with a pruned dynamic programming procedure as the forward pass.

[2] In (Deng et al., 2006), p. 5, the $p(a_k) = p(x, y)$ which determines the prior probability of having an alignment containing x source and y target sentences, is equal to 0 if $x < 1$ or $y < 1$. As $p(a_k)$ is a multiplicative factor of the model, the probability of having an insertion or a deletion is always equal to 0.

[3] In order to provide a precise comparison between our aligner and (Deng et al., 2006), we have implemented their model into our optimized framework.

[4] Note that our approach does not aim to find many-to-many alignments. None of the unsupervised sentence alignment approaches discussed in section 2 are able to correctly find that type of correspondence.

procedure required to align a parallel text with our method (section 3.6).

3.1 Sketch of Approach

Consider a parallel text composed of six source language sentences F_i and four target language sentences E_j. Further assume that the correct alignment between the given texts is composed of four correspondences: three 1-to-1 alignments between F1, E1; F2, E2 and F6, E4 as well as a 3-to-1 alignment between F3, F4, F5 and E3. Figure 1 illustrates this alignment.

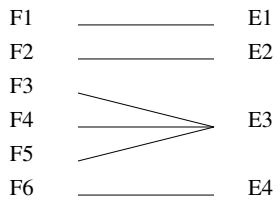


Figure 1: *Correct Alignment between F_i and E_j*

In the perspective of a statistical approach to sentence alignment, the alignment in figure 1 is found by computing the model-optimal alignment A^* for the bitext considered:

$$A^* = \underset{A}{\operatorname{argmax}} \prod_{a_k \in A} SCORE(a_k) \qquad (1)$$

where $SCORE(a_k)$ denotes the score attributed by the alignment model[5] to a minimal alignment a_k composing A^*. The optimization given in equation 1 relies on two commonly made assumptions: (c_1) a model-optimal alignment A^* can be decomposed into k minimal and independent alignments a_k; (c_2) each alignment a_k depends only on local portions of text in both languages.

The search for A^* is generally performed using a dynamic programming (DP) procedure over the space formed by the l source and m target sentences. The computation of A^* using a DP search relies on the assumption (c_3) that sentence alignment is a monotonic and continuous process. The DP procedure recursively computes the optimal score $D(l,m)^*$ for a sequence of alignments covering the whole parallel corpus. The optimal score $D(l,m)^*$ is given by the following recursion:

$$D(l,m)^* = \min_{0 \leq x,y \leq R \ , \ x=1 \vee y=1} D(l-x, m-y)^* - \log SCORE(a_k) \qquad (2)$$

where x denotes the number of sentences on the source language side of a_k and y the number of sentences on the target language side of a_k.

The constant R constitutes an upper bound to the number of sentences that are allowed on each side of a minimal alignment a_k. This constant has an important impact on the computational costs of the DP procedure insofar as it determines the number of minimal alignments that have to be compared and scored at each step of the recursion given in equation 2. As will be shown in section 3.2, the number of comparisons increases depending on R.

The solution we propose to the combinatorial growth of the number of performed operations consists of dividing the search for A^* into two steps. First, a model-optimal alignment A_1^*, in which the value of R is fixed to 1, is found. Second, the alignments a'_k composing A_1^* are merged into clusters m_r containing up to R sentences on either the source or target language side. The alignment composed of these clusters is A_R^*.

The search for the first alignment A_1^* is performed using a standard DP procedure as given in equation 2 but with $R = 1$. This first alignment is, hence, only composed of 0-to-1, 1-to-0 and 1-to-1 correspondences. Using our example, we show, in figure 2, the alignment A_1^* found in the first step of our approach. The neighbors of F4, that is F3 and F5, are aligned as 1-to-0 correspondences.

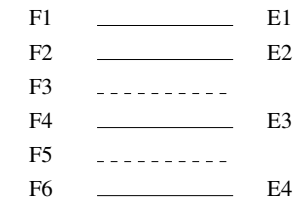


Figure 2: *A_1^* in our Approach (first step)*

The search for A_R^* is performed using a DP search over the alignments a'_k composing A_1^*. The score $D(A_R)^*$ obtained when all alignments $a'_k \in A_1^*$ have been optimally clustered can be written

[5]The alignment model will be presented in section 3.5.

recursively as:

$$D(A_R)^* = \min_{0 \leq r \leq R} D(A_R - r)^* - \log SCORE(m_r) \quad (3)$$

where $D(A_R - r)^*$ denotes the best score obtained for the prefix covering all minimal alignments in A_1^* except the last r minimal alignments considered for composing the last cluster m_r.

The application of the second step of our approach is illustrated in figure 3. The first alignment, between F1 and E1, cannot be merged to be part of a 1-to-many or many-to-1 cluster because the following alignment in A_1^* is also 1-to-1. So it must be retained as given in A_1^*. The five last alignments are, however, candidates for composing clusters. For instance, the alignment F2-E2 and F3-ϵ, where ϵ denotes the empty string, could be merged in order to compose the 2-to-1 cluster F2,F3-E2. However, in our example, the alignment model chooses to merge the alignments F3-ϵ, F4-E3 and F5-ϵ in order to compose the 3-to-1 cluster F3,F4,F5-E3.

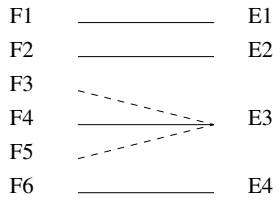


Figure 3: A_R^* *in our Approach (second step)*

3.2 Computational Gains

The aim of this section is to give an idea about why our method is faster than the standard DP approach. Let C denote the number of comparisons performed at each step of the recursion of the standard DP procedure, as given in equation 2. This amount is equivalent to the number of possible combinations of x source sentences with y target sentences. Hence, for an approach finding all types of correspondences except many-to-many, we have:

$$C = 2R + 1 \quad (4)$$

In terms of lookups in the word-correspondence tables of a model including lexical statistics, the number of operations C_l performed at each step of the recursion is given by:

$$C_l = R' * w^2 \quad (5)$$

where R' denotes the number of scored sentences[6]. w denotes the average length of each sentence in terms of words. The total number of lookups performed in order to align a parallel text containing l source and m target sentences using a standard DP procedure is hence given by:

$$L = R' * w^2 * l * m \quad (6)$$

In the perspective of our two-step search procedure, the computational costs of the search for the initial alignment A_1^* is given by:

$$L'_1 = w^2 * l * m \quad (7)$$

For the second step of our approach, because A_R^* is a cluster of A_1^*, the dynamic programming procedure used to find this alignment is no longer over the $l * m$ space formed by the source and target sentences but instead over the space formed by the minimal alignments a'_k in A_1^*. The average number of those alignments is approximately $\frac{l+m}{2}$.[7] The number of lookups performed at each step of our DP procedure is given by:

$$L'_2 = R' * w^2 * \frac{l+m}{2} \quad (8)$$

where R' and w are defined as in equation 6. The total number of lookups for our clustering approach is hence given by:

$$L'_{1+2} = (w^2 * l * m) + (R' * w^2 * \frac{l+m}{2}) \quad (9)$$

In order to compare the costs of our approach and a standard DP search over the $l * m$ space formed by the source and target sentences, we re-write equation 6 as:

$$L = (w^2 * l * m) + ((R' - 1) * w^2 * l * m) \quad (10)$$

The comparison of equation 9 with equation 10 shows that the computational gains obtained using our two-step approach reside in the reduction of the search space from $l * m$ to $\frac{l+m}{2}$.[8]

[6] In a framework where no caching of scores is performed, we have $R' = R^2 + R + 1$ compared sentences while score-caching allows one to reduce R' to R.

[7] Note that this amount tends to $l + m$ when A_1^* contains a large number of 0-to-1/1-to-0 correspondences.

[8] It should be noted that through efficient pruning, the search space of the standard (DP) procedure can be further reduced, see section 3.4.

3.3 Theoretical Background

We now present the theoretical foundation of our approach. First, we rewrite equation 1 in a more detailed fashion as:

$$A_R^* = \underset{A}{\operatorname{argmax}} \prod_{a_k(x_k,y_k)\in A_R} P(a_k(x_k,y_k), s_i^q, t_j^r) \tag{11}$$

with $0 \leq x_k, y_k \leq R$, where R denotes the maximal amounts x and y of source and target language sentences composing a minimal alignment $a_k(x_k, y_k)$. The distribution $P(a_k(x_k, y_k), s_i^q, t_j^r)$ specifies the alignment model presented in section 3.5.

As seen in section 3.1, the formulation of the alignment problem as given in equation 11 and the use of a DP search in order to solve this equation rely on the assumptions (c_1) to (c_3). Following these assumptions, a model-optimal alignment A_1^* can be defined as an ordered set of minimal alignments $a'_k(x_k, y_k)$, with $0 \leq x_k, y_k \leq 1$, where the aligned portions of text are sequential. In other words, if the $k-th$ alignment $a'_k(x_k, y_k)$ contains the sequences s_i^q and t_j^r of source and target language sentences, then the next alignment $a'_{k+1}(x_{k+1}, y_{k+1})$ is composed of the sequences s_{q+1}^u and t_{r+1}^v. Hence, each alignment composing A_R, with $R > 1$, can be obtained through sequential merging of a series of alignments $a'_k(x_k, y_k) \in A_1^*$.[9] Accordingly, the sequences of sentences s_1^u and t_1^v are obtained by merging s_1^q and t_1^r with s_{q+1}^u and t_{r+1}^v. It can then be assumed that (c_4) the ordered set of minimal alignments composing A_R^* under equation 11 is equivalent to the set of clusters obtained by sequentially merging the minimal alignments composing A_1^*. Following assumption (c_4), the optimization over $a_k(x_k, y_k) \in A_R$ is equivalent to an optimization over the merged alignments $m_r(x_r, y_r) \in A_R$. Hence, equation 11 is equivalent to:

$$A_R^* = \underset{A_R}{\operatorname{argmax}} \prod_{m_r(x_r,y_r)\in A_R} P(m_r(x_r,y_r), s_i^u, t_j^v) \tag{12}$$

where each $m_r(x_r, y_r)$ is obtained by merging r minimal alignments $a'_k(x_k, y_k) \in A_1^*$.

The computation of A_R^* is done in two steps. First, a model-optimal alignment A_1^* is found using a standard DP procedure as defined in equation 2 but with $R = 1$ and where $SCORE(a_k)$ is given by the alignment model $-\log P(a_k, s_{l-x+1}^l, t_{m-y+1}^m)$. In the second step, the search procedure used to find the optimal clusters is defined as in equation 3 but where $SCORE(m_r)$ is given by the alignment model $-\log P(m_r, s_i^u, t_j^v)$.

3.4 Search Space Pruning

In order to further reduce the costs of finding A_1^*, we initially pruned the search space in the same fashion as (Moore, 2002). We explored a narrow band around the main diagonal of the bitext to align. Each time the approximated alignment came close to the boundaries of the band, the search was reiterated with a larger band size. However, the computational costs for alignments that were not along the diagonal quickly increased with this pruning strategy. A high loss of efficiency was hence observed when aligning asymmetrical documents with this technique. Incidentally, Moore reports, in his experiments, that for the alignment of a parallel text containing 300 deleted sentences, the computational costs of his pruned DP procedure is 40 times higher than for a corpus containing no deletions.

In order to overcome this problem, we developed a pruning strategy that allows us to avoid the loss of efficiency occurring when aligning asymmetrical documents. Instead of exploring a narrow band around the main diagonal of the text to align, we use sentence-length statistics in order to compute an approximate path through the considered bitext. Our search procedure then explores the groups of sentences that are around this path. If the approximated alignment comes close to the boundaries of the band, the search is re-iterated.

The path initially provided using a sentence-length model[10] and then iteratively refined is closer to the correct alignment than the main diagonal of the bitext to align. Hence, the approximated alignment does not come close to the band

[9] Alignments of type 1-to-0/0-to-1 and 1-to-1 are assumed to be clusters where a minimal alignment $a'_k(x_k, y_k) \in A_1^*$ has been merged with the empty alignment $e_0(0,0)(\epsilon, \epsilon)$.

[10] The used model is the sentence-length based component of (Moore, 2002), which is able to find 1-to-0/0-to-1 correspondences.

as often as when searching around the main diagonal. This results in relatively high computational gains, especially for asymmetrical parallel texts (see section 4).

3.5 Moore's Alignment Model

The model we use is basically the same as in (Moore, 2002) but minor modifications have been made in order to integrate this model in our two-step clustering approach. The three component distributions of the model are given by[11]:

$$P(a_k, s_i^q, t_j^r) = P(a_k)P(s_i^q|a_k)P(t_j^r|a_k, s_i^q) \quad (13)$$

The first component, $P(a_k)$, specifies the generation of a minimal alignment a_k. The second component, $P(s_i^q|a_k)$, specifies the generation of a sequence s_i^q of source language sentences in a minimal alignment a_k. The last component, i.e. $P(t_j^r|a_k, s_i^q)$, specifies the generation of a sequence of target language sentences depending on a sequence of generated source sentences.

Our first modification to Moore's model concerns the component distribution $P(a_k)$. In the second pass of our two-step approach, which is the computation of the model-optimal clustered alignment A_R^*, we estimate $P(a_k)$ by computing the relative frequency of sequences of alignments a_k' in the initial alignment A_1^* that are candidates for composing a cluster m_r of specific size.[12] A second minimal modification to Moore's model concerns the lexical constituent of $P(t_j^r|a_k, s_i^q)$, which we denote here by $P(f_b|e_n, a_k)$. In contrast with Moore, we use the best alignment (Viterbi alignment) of each target word f_b with all source words e_n, according to IBM Model-1:

$$P(f_b|e_n, a_k) = \frac{\arg\max_{n=1}^{l_e} P_t(f_b|e_n)}{l_e + 1} \quad (14)$$

where l_e denotes the number of words in the source sentence(s) of a_k. Our experimental results have shown that this variant performed slightly better than Moore's summing over all alignments.

[11] In order to simplify the presentation of the model, we use the short notation a_k for denoting $a_k(x_k, y_k)$

[12] For the computation of A_1^*, the distribution $P(a_k)$ is defined as in Moore's work.

3.6 Alignment Procedure

In order to align a parallel text (s_1^l, t_1^m) we use a multi-pass procedure similar to (Moore, 2002) but where the last pass is replaced by our two-step clustering approach. In the first pass, an approximate alignment is computed using sentence-length based statistics and the one-to-one correspondences with likelihood higher than a given threshold are selected for the training of the IBM Model-1 translation tables[13]. Furthermore, each found alignment is cached in order to be used as the initial diagonal determining the search space for the next pass. In the second pass, the corpus is re-aligned according to our two-step approach: (i) a model-optimal[14] alignment containing at most one sentence on each side of the minimal alignments $a_k(x_k, y_k)$ is found; (ii) those alignments are model-optimally merged in order to obtain an alignment containing up to R sentences on each side of the clusters $m_r(x_r, y_r)$. In our experiments, a maximum number of 4 sentences is allowed on each side of a cluster.

4 Experiments

We evaluate our approach (CA) using three baselines against which we compare alignment quality and computational costs.[15] The first (Mo) is the method by (Moore, 2002). As a second baseline (Std), we have implemented an aligner that finds the same type of correspondences as our approach but performs a standard DP search instead of our two-pass clustering procedure and implements Moore's pruning strategy. Our third baseline (Std P.) is similar to (Std) but integrates our pruning technique.[16] We also evaluate the impact

[13] Words with frequency < 3 in the corpus have been dropped.

[14] This is optimal according to the alignment model which will be presented in section 3.5.

[15] We do not evaluate sentence-length based methods in our experiments because these methods obtain an F_1 which is generally about 10% lower than for our approach on symmetrical documents. For asymmetrical documents the performance is even worse. For example, when using Gale&Church F_1 sinks to 13.8 on documents which are not aligned at paragraph level and contain small deletions.

[16] We do not include (Deng et al., 2006) in our experiments because our implementation of this aligner is 550 times slower than our proposed method and the inability to find 1-to-0/0-to-1 correspondences makes it inappropriate for asymmetrical documents.

S	1-1	1-N/N-1	0-1/1-0	Oth.	Tot.
1	88.2%	10.9 %	0.005%	0.85%	3,877
2	91.9%	7.5%	0.007%	0.53%	2,646
3	91.6%	2.7%	4.3%	1.4%	23,715
4	44.8%	6.2%	49%	0.01%	2,606

Table 1: *Test Set for Evaluation with $2 \leq N \leq 4$*

of our aligner on the overall performance of an MT system.

Evaluation. We evaluate the alignment accuracy of our approach using four test sets annotated at sentence-level. The two first are composed of hand aligned documents from the Europarl corpus for the language-pairs German-to-English and French-to-English. The third is composed of an asymmetric document from the German-to-English part of the Europarl corpus. Our fourth test set is a version of the BAF corpus (Simard, 1998), where we corrected the tokenization. BAF is an interesting heterogeneous French-to-English test set composed of 11 texts belonging to four different genres. The types of correspondences composing our test sets are given in table 1. The metrics used are precision, recall and F_1[17]. Only alignments that correspond exactly to reference alignments count as correct. The computational costs required for each approach are measured in seconds. The time required to train IBM Model-1 is not included in our calculations[18].

Summary of Results. Regarding alignment accuracy, the results in table 2 show that (CA) obtains, on average, an F_1 that is 4.30 better than for (Mo) on symmetrical documents. The results in table 3 show that, on asymmetrical texts, (CA) achieves an F_1 which is 8.97 better than (Mo). The accuracy obtained using (CA), (Std) and (Std P.) is approximately the same. We have further compared the accuracy of (CA) with (Std) for finding 1-to-many/many-to-1 alignments. The obtained results show that (CA) achieves an F_1 that is 5.0 better than (Std).

Regarding computational costs, the time required by (CA) is on average 4 times larger than for (Mo) when aligning symmetrical documents. On asymmetrical documents, (Mo) is, however, only 1.5 times faster than (CA). Compared to (Std), (CA) is approximately 6 times faster on symmetrical and 80 times faster on asymmetrical documents. The time of (Std P.) is 3 times higher than for (CA) on symmetrical documents and 22 times higher on asymmetrical documents. This shows that, first, our pruning technique is more efficient than Moore's and, second, that the main increase in speed is due to the two step clustering approach.

Discussion. On the two first test sets, (Mo) achieves high precision while the obtained recall is limited by the number of correspondences that are not 1-to-1 (see table 1). Regarding (Std), (Std P.) and (CA), all aligners achieve high precision as well as high recall, leading to an F_1 which is over 98% for both documents. The computational costs of (CA) for the alignment of symmetrical documents are, on average, 4 times higher than (Mo), 6 times lower than (Std) and 3.5 times lower than (Std P.). On our third test set (Mo) achieves, with an F_1 of 88.70, relatively poor recall while the other aligners reach precision and recall values that are over 98%. Regarding the computational costs, (CA) is only 1.5 times slower than (Mo) on asymmetrical documents while it is 80 times faster than (Std) and about 22 times faster than (Std P.). On our fourth test set all evaluated aligners perform approximately the same than on Europarl. While (Mo) obtains, with 94.46, an F_1 which is the same as for Europarl, (CA) performs, with an F_1 of 97.67, about 1% worse than on Europarl. A slightly larger decrease of 1.6% is observed for (Std) which obtains 96.81 F_1. Note, however, that (CA), (Std) and (Std P.) still perform about 3% better than (Mo). Regarding computational costs, (CA) is 4 times slower than (Mo) and 40 times faster than (Std). The high difference in speed between our approach and (Std) is due to the fact that the BAF corpus contains texts of variable symmetry while (Std) shows a great speed decrease when aligning asymmetrical documents. Finally, we have compared the accuracy of (Std) and (CA) for the finding of 1-to-many/many-to-1 alignments containing at least 3 sentences on the "many"

[17] We measure precision, recall and F_1 on the 1-to-N/N-to-1 alignments, $N >= 1$, which means that we view insertions and deletions as "negative" decisions, like Moore.

[18] The reason for this decision is that our optimized framework trains the Model-1 translation tables far faster than Moore's bilingual sentence aligner.

Appr.	Lang.	Prec.	Rec.	F1	Speed
Mo	D-E	98.75	87.88	92.99	935s
Mo	F-E	98.97	91.56	95.12	1,661s
Std	D-E	98.42	98.57	98.49	24,152s
Std	F-E	98.45	98.83	98.64	35,041s
Std P.	D-E	98.37	98.49	98.43	13,387s
Std P.	F-E	98.41	98.78	98.60	21,848s
CA	D-E	98.25	98.70	98.47	3,461s
CA	F-E	98.00	98.60	98.30	6,978s

Table 2: *Performance on Europarl*

Appr.	Prec.	Rec.	F1	Speed
Mo	97.90	81.08	88.70	552s
Std	97.66	97.74	97.70	71,475s
Std P.	97.74	97.81	97.77	17,502s
CA	97.38	97.97	97.67	800s

Table 3: *Performance on asym. documents*

Appr.	Prec.	Rec.	F1	Speed
Mo	96.58	92.43	94.46	563s
Std	96.82	96.80	96.81	84,988s
CA	97.05	97.63	97.34	2,137s

Table 4: *Performance on BAF*

side. This experiment has shown that (Std) finds a larger amount of those alignments while making numerous wrong conjectures. On the other hand, (CA) finds less 1-to-many/many-to-1 correspondences but makes only few incorrect hypotheses. Hence, F_1 is about 5% better for (CA).

MT evaluation We also measured the impact of 1-to-N/N-to-1 alignments (which are not extracted by Moore) on MT. We used standard settings of the Moses toolkit, and the Europarl devtest2006 set as our test set. We ran MERT separately for each system. System (s1) was trained just on the 1-to-1 alignments extracted from the Europarl v3 corpus by our system while system (s2) was trained with all correspondences found. (s1) obtains a BLEU score of 0.2670 while (s2) obtains a BLEU score of 0.2703. Application of the pairwise bootstrap test (Koehn, 2004) shows that (s2) is significantly better than (s1).

5 Conclusion

We have addressed the problem of unsupervised and language-pair independent alignment of symmetrical and asymmetrical parallel corpora. We have developed a novel approach which is fast and allows us to achieve high accuracy in terms of F_1 for the alignment of bilingual corpora. Our method achieved high accuracy on symmetrical and asymmetrical parallel corpora, and we have shown that the 1-to-N/N-to-1 alignments extracted by our approach are useful. The source code of the aligner and the test sets are available at http://sourceforge.net/projects/gargantua .

6 Acknowledgements

The first author was partially supported by the Hasler Stiftung[19]. Support for both authors was provided by Deutsche Forschungsgemeinschaft grants Models of Morphosyntax for Statistical Machine Translation and SFB 732.

References

Brown, Peter F., Jennifer C. Lai, and Robert L. Mercer. 1991. Aligning sentences in parallel corpora. In *In Proceedings of 29th Annual Meeting of the Association for Computational Linguistics*, pages 169–176.

Ceausu, Alexandru, Dan Stefanescu, and Dan Tufis. 2006. Acquis communautaire sentence alignment using support vector machines. In *LREC 2006: Fifth International Conference on Language Resources and Evaluation*.

Chen, Stanley F. 1993. Aligning sentences in bilingual corpora using lexical information. In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*, pages 9–16.

Deng, Yoggang, Shankar Kumar, and William Byrne. 2006. Segmentation and alignment of parallel text for statistical machine translation. *Natural Language Engineering*, 12:1–26.

Fattah, Mohamed Abdel, David B. Bracewell, Fuji Ren, and Shingo Kuroiwa. 2007. Sentence alignment using p-nnt and gmm. *Computer Speech and Language*, (21):594–608.

Gale, William A. and Kenneth Ward Church. 1993. A program for aligning sentences in bilingual corpora. *Computational Linguistics*, 19(1):75–102.

Gautam, Mrityunjay and R. M. K. Sinha. 2007. A program for aligning sentences in bilingual corpora. *Proceedings of the International Conference*

[19] http://www.haslerstiftung.ch/.

on Computing: Theory and Applications, ICCTA '07, (1):480–484.

Haruno, M. and T. Yamazaki. 1996. High-performance bilingual text alignment using statistical and dictionary information. In *Proceedings of ACL '96*, pages 131–138.

Koehn, Philipp. 2004. Statistical significance tests for machine translation evaluation. In Lin, Dekang and Dekai Wu, editors, *Proceedings of EMNLP 2004*, pages 388–395, Barcelona, Spain, July. Association for Computational Linguistics.

Ma, Xiaoyi. 2006. Champollion: A robust parallel text sentence aligner. In *LREC 2006: Fifth International Conference on Language Resources and Evaluation*.

Melamed, I. Dan. 2000. Models of translational equivalence among words. *Computational Linguistics*, 26:221–249.

Moore, Robert. 2002. Fast and accurate sentence alignment of bilingual corpora. In *In Proceedings of 5th Conference of the Association for Machine Translation in the Americas*, pages 135–244.

Simard, Michel and Pierre Plamondon. 1998. Bilingual sentence alignment: Balancing robustness and accuracy. *Machine Translation*, 13(1):59–80.

Simard, Michel. 1998. The baf: A corpus of english-french bitext. In *Proceedings of LREC 98*, Granada, Spain.

Wu, Dekai. 1994. Aligning a parallel English-Chinese corpus statistically with lexical criteria. In *In Proceedings of the 32nd Annual Conference of the Association for Computational Linguistics, 80–87, Las*, pages 80–87.

Automatic Acquisition of Lexical Formality

Julian Brooke, Tong Wang, and Graeme Hirst
Department of Computer Science
University of Toronto
{jbrooke,tong,gh}@cs.toronto.edu

Abstract

There has been relatively little work focused on determining the formality level of individual lexical items. This study applies information from large mixed-genre corpora, demonstrating that significant improvement is possible over simple word-length metrics, particularly when multiple sources of information, i.e. word length, word counts, and word association, are integrated. Our best hybrid system reaches 86% accuracy on an English near-synonym formality identification task, and near perfect accuracy when comparing words with extreme formality differences. We also test our word association method in Chinese, a language where word length is not an appropriate metric for formality.

1 Introduction

The derivation of lexical resources for use in computational applications has been focused primarily on the denotational relationships among words, e.g. the synonym and hyponym relationships encapsulated in WordNet (Fellbaum, 1998). Largely missing from popular lexical resources such as WordNet and the General Inquirer (Stone et al., 1966) is stylistic information; there are, for instance, no resources which provide comprehensive information about the formality level of words, which relates to the appropriateness of a word in a given context. Consider, for example, the problem of choice among near-synonyms: there are only minor denotational differences among synonyms such as *get*, *acquire*, *obtain*, and *snag*, but it is difficult to construct a situation where any choice would be equally suitable. The key difference between these words is their formality, with *acquire* the most formal and *snag* the most informal.

In this work, we conceive of formality as a continuous property. This approach is inspired by resources such as *Choose The Right Word* (Hayakawa, 1994), in which differences between synonyms are generally described in relative rather than absolute terms, as well as linguistic literature in which the quantification of stylistic differences among genres is framed in terms of dimensions rather than discrete properties (Biber, 1995). We begin by defining the *formality score* for a word as a real number value in the range 1 to -1, with 1 representing an extremely formal word, and -1 an extremely informal word. A formality lexicon, then, gives a FS score to every word within its coverage.

The core of our approach to the problem of classifying lexical formality is the automated creation of formality lexicons from large corpora. In this paper, we focus on the somewhat low-level task of identifying the relative formality of word pairs; we believe, however, that a better understanding of lexical formality is relevant to a number of problems in computational linguistics, including sub-fields such as text generation, error correction of (ESL) writing, machine translation, text classification, text simplification, word-sense disambiguation, and sentiment analysis. One conclusion of our research is that formality variation is omnipresent in natural corpora, but it does not follow that the identification of these differences on the lexical level is a trivial one; nevertheless,

Coling 2008: Poster Volume, pages 90–98,
Beijing, August 2010

we are able to make significant progress using the methods presented here, in particular the application of latent semantic analysis to blog corpora.

2 Related Work

As far as we are aware, there are only a few lines of research explicitly focused on the question of linguistic formality. In linguistics proper, the study of register and genre usually involves a number of dimensions or clines, sometimes explicitly identified as formality (Leckie-Tarry, 1995; Carter, 1998), or decomposed into notions such as informational versus interpersonal content (Biber, 1995). Heyligen and Dewaele (1998) provide a part-of-speech based quantification of textual contextuality (which they argue is fundamental to the notion of formality); their metric has been used, for instance, in a computational investigation of the formality of online encyclopedias (Emigh and Herring, 2005). In this kind of quantification, however, there is little, if any, focus on individual elements of the lexicon. In computational linguistics, formality has received attention in the context of text generation (Hovy, 1990); of particular note relevant to our research is the work of Inkpen and Hirst (2006), who derive boolean formality tags from *Choose the Right Word* (Hayakawa, 1994). Like us, their focus was improved word choice, though the approach was much broader, also including dimensions such as polarity. An intriguing example of formality relevant to text classification is the use of informal language (slang) to help distinguish true news from satire (Burfoot and Baldwin, 2009).

Our approach to this task is inspired and informed by automatic lexical acquisition research within the field of sentiment analysis (Turney and Littman, 2003; Esuli and Sebastiani, 2006; Taboada and Voll, 2006; Rao and Ravichandra, 2009). Turney and Littman (2003) apply latent semantic analysis (LSA) (Landauer and Dumais, 1997) and pointwise mutual information (PMI) to derive semantic orientation ratings for words using large corpora; like us, they found that LSA was a powerful technique for deriving this lexical information. The lexical database SentiWordNet (Esuli and Sebastiani, 2006) provides 0–1 rankings for positive, negative, and neutral polarity, derived automatically using relationships between words in WordNet (Fellbaum, 1998). Unfortunately, WordNet synsets tend to cut across the formal/informal distinction, and so the resource is not obviously useful for our task.

The work presented here builds directly on a pilot study (Brooke et al., 2010), the focus of which was the construction of formality score (FS) lexicons. In that work, we employed less sophisticated forms of some of the methods used here in a relatively small dataset (the Brown Corpus), providing a proof of concept, but with poor coverage, and with no attempt to combine the methods to maximize performance. However, the small dataset allowed us to do a thorough test of certain options associated with our task. In particular we found that using a similarity metric based on LSA gave good performance across our test sets, especially when the term-document matrix was binary (unweighted), the k-value used for LSA was small, and the method used to derive a formality score was cosine similarity to our seed terms. A metric using total word counts in corpora with divergent formality also showed promise, with both methods performing above our word-length baseline for words within their coverage. PMI, by comparison, proved less effective, and we do not pursue it further here.

3 Data and Resources

3.1 Word Lists

All the word lists discussed here are publicly available.[1] We begin with two, one formal and one informal, that we use both as seeds for our lexicon construction methods and as test sets for evaluation (our gold standard). We assume that all slang terms are by their very nature informal and so our 138 informal seeds were taken primarily from an online slang dictionary[2] (e.g. *wuss*, *grubby*) and also include some contractions and interjections (e.g. *cuz*, *yikes*). The 105 formal seeds were selected from a list of discourse markers (e.g. *moreover*, *hence*) and adverbs from a sentiment lexicon (e.g. *preposterously*, *inscrutably*); these sources were chosen to avoid words with

[1] http://www.cs.toronto.edu/~jbrooke/FormalityLists.zip

[2] http://onlineslangdictionary.com/

overt topic, and to ensure that there was some balance of sentiment across formal and informal seed sets. Part of speech, however, is not balanced across our seed sets.

Another test set we use to evaluate our methods is a collection of 399 pairs of near-synonyms from *Choose the Right Word* (CTRW), a manual for assisting writers with synonym word choice; each pair was either explicitly or implicitly compared for formality in the book. Implicit comparison included statements such as *this is the most formal of these words*; in those cases, and more generally, we avoided words appearing in more than one comparison (there are no duplicate words in our CTRW set), as well as multiword expressions and words whose formality is strongly ambiguous (i.e. word-sense dependent). An example of this last phenomenon is the word *cool*, which is used colloquially in the sense of *good* but more formally as in the sense of *cold*. Partly as a result of this polysemy, which is clearly more common among informal words, our pairs are biased toward the formal end of the spectrum; although there are some informal comparisons, e.g. *belly-ache/whine*, *wisecrack/joke*, more typical pairs include *determine/ascertain* and *hefty/ponderous*. Despite this imbalance, one obvious advantage of using near-synonyms in our evaluation is that factors other than linguistic formality (e.g. topic, opinion) are less likely to influence performance. In general, the CTRW allows for a more objective, fine-grained evaluation of our methods, and is oriented towards our primary interest, near-synonym word choice.

To test the performance of our unsupervised method beyond English, one of the authors (a native speaker of Mandarin Chinese) created two sets of Chinese two-character words, one formal, one informal, based on but not limited to the words in the English sets. The Chinese seeds include 49 formal seeds and 43 informal seeds.

3.2 Corpora

Our corpora fall generally into three categories: formal (written) copora, informal (spoken) corpora, and mixed corpora. The Brown Corpus (Francis and Kučera, 1982), our development corpus, is used here both as a formal and mixed corpus. Although extremely small by modern corpus standards (only 1 million words), the Brown Corpus has the advantage of being compiled explicitly to represent a range of American English, though it is all of the published, written variety. The Switchboard (SW) Corpus is a collection of American telephone conversations (Godfrey et al., 1992), which contains roughly 2400 conversations with over 2.6 million word tokens; we use it as an informal counterpart to the Brown Corpus. Like the Brown Corpus, The British National Corpus (Burnard, 2000) is a manually-constructed mixed-genre corpus; it is, however, much larger (roughly 100 million words). It contains a written portion (90%), which we use as a formal corpus, and a spontaneous spoken portion (4.3%), which we use as an informal corpus. Our other mixed corpora are two blog collections available to us: the first, which we call our development blog corpus (DevBlog) contains a total of over 900,000 English blogs, with 216 million tokens.[3] The second is the 'first tier' English blogs included in the publicly available ICSWM 2009 Spinn3r Dataset (Burton et al., 2009), a total of about 1.3 billion word tokens in 7.5 million documents. For our investigations in Chinese, we use the Chinese portion of the ICSWM blogs, approximately 25.4 million character tokens in 86,000 documents.

4 Methods

4.1 Simple Formality Measures

The simplest kind of formality measure is based on word length, which is often used directly as an indicator of formality for applications such as genre classification (Karlgren and Cutting, 1994). Here, we use logarithmic scaling to derive a FS score based on word length. Given a maximum word length L[4] and a word w of length l, the formality score function, $FS(w)$, is given by:

$$FS(w) = -1 + 2\frac{\log l}{\log L}$$

[3] These blogs were gathered by the University of Toronto Blogscope project (www.blogscope.net) over a week in May 2008.

[4] We use an upper bound of 28 characters, which is the length of *antidisestablishmentarianism*, the prototypical longest word in English; this value of L provides an appropriate formality/informality threshold, between 5- and 6-letter words

For hyphenated terms, the length of each component is averaged. Though this metric works relatively well for English, we note that it is problematic in a language with significant word agglutination (e.g. German) or without an alphabet (e.g. Chinese, see below).

Another straightforward method is the assumption that Latinate prefixes and suffixes are indicators of formality in English (Kessler et al., 1997), i.e. informal words will not have Latinate affixes such as *-ation* and *intra-*. Here, we simply assign words that appear to have such a prefix or suffix an FS of 1, and all other words an FS of -1.

Our frequency methods derive FS from word counts in corpora. Our first, naive approach assumes a single corpus, where either formal words are common and informal words are rare, or vice versa. To smooth out the Zipfian distribution, we use the frequency rank of words as exponentials; for a corpus with R frequency ranks, the FS for a word of rank r under the *formal is rare* assumption is given by:

$$FS(w) = -1 + 2\frac{e^{(r-1)}}{e^{(R-1)}}$$

Under the *informal is rare* assumption:

$$FS(w) = 1 - 2\frac{e^{(r-1)}}{e^{(R-1)}}$$

We have previously shown that these methods are not particularly effective on their own (Brooke et al., 2010), but we note that they provide useful information for a hybrid system.

A more sophisticated method is to use two corpora that are known to vary with respect to formality and use the relative appearance of words in each corpus as the metric. If word appears n times in a (relatively) formal corpus and m times in an informal corpus (and one of m, n is not zero), we derive:

$$FS(w) = -1 + 2\frac{n}{m \times N + n}$$

Here, N is the ratio of the size (in tokens) of the informal corpus (IC) to the formal corpus (FC). We need the constant N so that an imbalance in the size of the corpora does not result in an equivalently skewed distribution of FS.

4.2 Latent Semantic Analysis

Next, we turn to LSA, a technique for extracting information from a large corpus of texts by (drastically) reducing the dimensionality of a term–document matrix, i.e. a matrix where the row vectors correspond to the appearance or (weighted) frequency of words in a set of texts. In essence, LSA simplifies the variation of words across a collection of texts, exploiting document–document correlation to produce information about the k most important dimensions of variation ($k <$ total number of documents), which are generally thought to represent semantic concepts, i.e. topic. The mathematical basis for this transformation is singular value decomposition[5]; for the details of the matrix transformations, we refer the reader to the discussion of Turney and Littman (2003). The factor k, the number of columns in the compacted matrix, is an important variable in any application of LSA, one is generally determined by trial and error (Turney and Littman, 2003).

LSA is computationally intensive; in order to apply it to extremely large blog corpora, we need to filter the documents and terms before building our term–document matrix. We adopt the following strategy: to limit the number of documents in our term–document matrix, we first remove documents less than 100 tokens in length, with the rationale that these documents provide less co-occurrence information. Second, we remove documents that either do not contain any target words (i.e. one of our seeds or CTRW test words), or contain only target words which are among the most common 20 in the corpus; these documents are less likely to provide us with useful information, and the very common target terms will be well represented regardless. We further shrink the set of terms by removing all hapax legomena; a single appearance in a corpus is not enough to provide reliable co-occurrence information, and roughly half the words in our blog corpora appear only once. Finally, we remove symbols and all words which are not entirely lower

[5] We use the implementation included in Matlab; we take the rows of the decomposed U matrix weighted by the singular values in Σ for our word vectors. Using no weights or Σ^{-1} generally resulted in worse performance, particularly with the CTRW sets.

case; we are not interested, for instance, in numbers, acronyms, and proper nouns. We can estimate the effect this filtering has on performance by testing it both ways in a development corpus.

Once a k-dimensional vector for each relevant word is derived using LSA, a standard method is to use the cosine of the angle between a word vector and the vectors of seed words to identify how similar the distribution of the word is to the distribution of the seeds. To begin, each formal seed is assigned a FS value of 1, each informal seed a FS value of -1, and then a raw seed similarity score (FS') is calculated for each word w:

$$FS'(w) = \sum_{s \in S, s \neq w} W_s \times FS(s) \times \cos(\theta(w,s))$$

S is the set of all seeds. Note that seed terms are excluded from their own FS calculation, this is equivalent to *leave-one-out* cross-validation. W_s is a weight that depends on whether s is a formal or informal seed, W_i (for informal seeds) is calculated as:

$$W_i = \frac{\sum_{f \in F} FS(f)}{|\sum_{i \in I} FS(i)| + \sum_{f \in F} FS(f)}$$

and W_f (for formal seeds) is:

$$W_f = \frac{|\sum_{i \in I} FS(i)|}{|\sum_{i \in I} FS(i)| + \sum_{f \in F} FS(f)}$$

Here, I is the set of all informal seeds, and F is the set of all formal seeds. These weights have the effect of countering any imbalance in the seed set, as formal and informal seeds ultimately have the same (potential) influence on each word, regardless of their count. This weighting is necessary for the iterative extension of this method discussed in the next section.

We calculate the final FS score as follows:

$$FS(w) = \frac{FS'(w) - FS'(r)}{N_w}$$

The word r is a reference term, a common function word that has no formality.[6] This has the effect of countering any (moderate) bias that might exist in the corpus; in the Brown Corpus, for instance, function words have positive formality before this step, simply because formal words occurred more often in the corpus. N_w is a normalization factor, either

$$N_w = \max_{w_i \in I'} |FS'(w_i) - FS'(r)|$$

for all $w_i \in I'$ or

$$N_w = \max_{w_f \in F'} |FS'(w_f) - FS'(r)|$$

for all $w_f \in F'$. I' contains all words w such that $FS'(w) - FS'(r) < 0$, and F' contains all words w such that $FS'(w) - FS'(r) > 0$. This ensures that the resulting lexicon has terms exactly in the range 1 to -1, with the reference word r at the midpoint.

We also tested the LSA method in Chinese. The only major relevant difference between Chinese and English is word segmentation: Chinese does not have spaces between words. To sidestep this problem, we simply included all character bigrams found in our corpus. The drawback of this approach in the inclusion of a huge number of nonsense 'words' (1.3 million terms in just 86,000 documents), however we are at least certain to identify all instances of our seeds.

4.3 Hybrid Methods

There are a number of ways to leverage the information we derive from our basic methods. One intriguing option is to use the basic FS measures as the starting point for an iterative process using the LSA cosine similarity. Under this paradigm, all words in the starting FS lexicon are potential seed words; we choose a cutoff value for inclusion in the seed word set (e.g. words which have at least .5 or $-.5$ FS), and then carry out the cosine calculations, as above, to derive new FS values (a new FS lexicon). We can repeat this process as many times as required, with the idea that the connections between various words (as reflected in their LSA-derived vectors) will cause the system to converge towards the true FS values.

A simple hybrid method that combines the two word count models uses the ratio of word counts in two corpora to define the center of the FS spectrum, but single corpus methods to define the extremes. Formally, if m and n (word counts for the

[6] The particular choice of this word is relatively unimportant; common function words all have essentially the same LSA vectors because they appear at least once in nearly every document of any size. For English, we chose r = *and*, and for Chinese, r = *yinwei* (*because*); there does not seem to be an obvious two-character, formality-neutral equivalent to *and* in Chinese.

informal corpus *IC* and formal corpus *FC*, respectively) are both non-zero, then FS is given by:

$$FS(w) = -0.5 + \frac{n}{m \times N + n}$$

However, if n is zero, FS is given by:

$$FS(w) = -1 + 0.5\frac{e^{\sqrt{r_{IC}-1}}}{e^{\sqrt{R_{IC}-1}}}$$

where r_{IC} is the frequency rank of the word in IC, and R_{IC} is the total number of ranks in IC. If m is zero, FS is given by:

$$FS(w) = 1 - 0.5\frac{e^{\sqrt{r_{FC}-1}}}{e^{\sqrt{R_{FC}-1}}}$$

where i is the rank of the word in IC, and R_{IC} is the total number of frequency ranks in IC). This function is undefined in the case where m and n are both zero. Intuitively, this is a kind of backoff, relying on the idea that words of extreme formality are rare even in a corpus of corresponding formality, whereas words in the *core vocabulary* (Carter, 1998), which are only moderately formal, will appear in all kinds of corpora, and thus are amenable to the ratio method.

Finally, we explore a number of ways to combine lexicons directly. The motivation for this is that the lexicons have different strengths and weaknesses, representing partially independent information. An obvious method is an averaging or other linear combination of the scores, but we also investigate vote-based methods (requiring agreement among n dictionaries). Beyond these simple options, we test support vector machines and naive Bayes classification using the WEKA software suite (Witten and Frank, 2005), applying 10-fold cross-validation using default WEKA settings for each classifier. The features here are task dependent (see Section 5); for the pairwise task, we use the difference between the FS value of the words in each lexicon, rather than their individual scores. Finally, we can use the weights from the SVM model of the CTRW (pairwise) task to interpolate an optimal formality lexicon.

5 Evaluation

We evaluate our methods using the gold standard judgments from the seed sets and CTRW word pairs. To differentiate the two, we continue to use the term *seed* for the former; in this context, however, these 'seed sets' are being viewed as a test set (recall that our LSA method is equivalent to *leave-one-out* cross-validation).

We derive the following measures: first, the coverage (Cov.) is the percentage of words in the set that are covered under the method. The class-based accuracy (C-Acc.) of our seed sets is the percentage of covered words which are correctly classified as formal (FS > 0) or informal (FS < 0). The pair-based accuracy (P-Acc.) is the result of exhaustively pairing words in the two seed sets and testing their relative formality; that is, for all $w_i \in I$ and $w_f \in F$, the percentage of w_i/w_f pairs where $FS(w_i) < FS(w_f)$. For the CTRW pairs there are only two metrics, the coverage and the pair-based accuracy; since the CTRW pairs represent relative formality of varying degrees, it is not possible to calculate a class-based accuracy.

The first section of Table 1 provides the results for the basic methods in various corpora. The word length (1) and morphology-based (2) methods provide good coverage, but poor accuracy, while the word count ratio methods (3–4) are fairly accurate, but suffer from low coverage. The LSA results in Table 1 are the best for each corpus across the k values we tested. When both coverage and accuracy are considered, there is a clear benefit associated with increasing the amount of data, though the difference between the Dev-Blog and ICWSM suggests diminishing returns. The performance of the filtered Dev-Blog is actually slightly better than the unfiltered versions (though there is a drop in coverage), suggesting that filtering is a good strategy.

In our previous work (Brooke et al., 2010), we noted that CTRW set performance in the Brown dropped for $k > 3$, while performance on the seed set was mostly steady as k increased. Figure 1 shows the pairwise performance of each test set for various corpora across various k. The results here are similar; all three corpora reach a CTRW maximum at a relatively low k values (though higher than Brown Corpus); however the seed set performance in each corpus continues to improve (though marginally) as k increases, while CTRW performance drops. An explanation for this is that

Table 1: Seed coverage, class-based accuracy, pairwise accuracy, CTRW coverage, and pairwise accuracy for various FS lexicons and hybrid methods (%).

Method	Seed set			CTRW set	
	Cov.	**C-Acc.**	**P-Acc.**	**Cov.**	**P-Acc.**
Simple					
(1) Word length	100	86.4	91.8	100	63.7
(2) Latinate affix	100	74.5	46.3	100	32.6
(3) Word count ratio, Brown and Switchboard	38.0	81.5	85.7	36.0	78.2
(4) Word count ratio, BNC Written vs. Spoken	60.9	89.2	97.3	38.8	74.3
(5) LSA (k=3), Brown	51.0	87.1	94.2	59.6	73.9
(6) LSA (k=10), BNC	94.7	83.0	98.3	96.5	69.4
(7) LSA (k=20), Dev-Blog	100	91.4	96.8	99.0	80.5
(8) LSA (k=20), Dev-Blog, filtered	99.0	92.1	97.0	97.7	80.5
(9) LSA (k=20), ICWSM, filtered	100	93.0	98.4	99.7	81.9
Hybrid					
(10) BNC ratio with backoff (4)	97.1	78.8	75.7	97.0	78.8
(11) Combined ratio with backoff (3 + 4)	97.1	79.2	79.9	97.5	79.9
(12) BNC weighted average (10,6), ratio 2:1	97.1	83.5	90.0	97.0	83.2
(13) Blog weighted average (9,7), ratio 4:1	100	93.8	98.5	99.7	83.4
(14) Voting, 3 agree (1, 6, 7, 9, 11)	92.6	99.1	99.9	87.0	91.6
(15) Voting, 2 agree (1, 11, 13)	86.8	99.1	100	81.5	96.9
(16) Voting, 2 agree (1, 12, 13)	87.7	98.6	100	82.7	97.3
(17) SVM classifier (1, 2, 6, 7, 9, 11)	100	97.9	99.9	100	84.2
(18) Naive Bayes classifier (1, 2, 6, 7, 9, 11)	100	97.5	99.8	100	83.9
(19) SVM (Seed, class) weighted (1, 2, 6, 7, 9, 11)	100	98.4	99.8	100	80.5
(20) SVM (CTRW) weighted (1, 6, 7, 9, 11)	100	93.0	99.0	100	86.0
(21) Average (1, 6, 7, 9, 11)	100	95.9	99.5	100	84.5

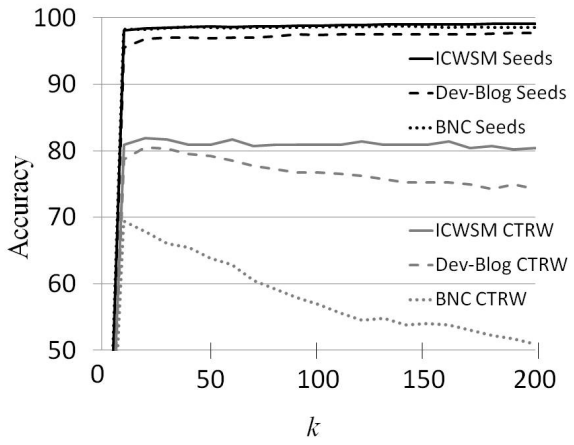


Figure 1: Seed and CTRW pairwise accuracy, LSA method for large corpora k, $10 \leq k \leq 200$.

the seed terms represent extreme examples of formality; thus there are numerous semantic dimensions to distinguish them. However, the CTRW set includes near-synonyms, many with only relatively subtle differences in formality; for these pairs, it is important to focus on the core dimensions relevant to formality, which are among the first discovered in a factor analysis of mixed-register texts (Biber, 1995).

With regards to hybrid methods, we first briefly summarize our testing with the iterative model, which included extensive experiments using basic lexicons and the LSA vectors derived from the Brown Corpus, and some targeted testing with the blog corpora (iteration on these corpora is extraordinarily time-consuming). In general, we found only that there were only small, inconsistent benefits to be gained from the iterative ap-

proach. More generally, the intuition behind the iterative method, i.e. that performance would increase with an drastic increase in the number of seeds, was found to be flawed: in other testing, we found that we could randomly remove most of the seeds without negatively affecting performance. Even at relatively high k values, it seems that a few seeds are enough to calibrate the model.

The ratio (with backoff) hybrid built from the BNC (10) provides CTRW performance that is comparable the best LSA models, though performance in the seed sets is somewhat poor; supplementing with word counts from the Brown Corpus and Switchboard Corpus provides a small improvement (11). The weighed hybrid dictionaries in (12,13) demonstrate that it is possible to effectively combine lexicons built using two different methods on the same corpus (12) or the same method on different corpora (13); the former, in particular, provides an impressive boost to CTRW accuracy, indicating that word count and word association methods are partially independent.

The remainder of Table 1 shows the best results using voting, averaging, and weighting. The voting results (14–16) indicate that it is possible to sacrifice some coverage for very high accuracy in both sets, including a near-perfect score in the seed sets and significant gains in CTRW performance. In general, the best accuracy without a significant loss of coverage came from 2 of 3 voting (15–16), using dictionaries that represented our three basic sources of information (word length, word count, and word association). The machine learning hybrids (17–18) also demonstrate a marked improvement over any single lexicon, though it is important to note that each accuracy score here reflects a different task-specific model. Hybrid FS lexicons built with the weights learned by the SVM models (19–20) provide superior performance on the task corresponding to the model used, though the simple averaging of the best dictionaries (21) also provides good performance across all evaluation metrics.

Finally, the LSA results for Chinese are modest but promising, given the relatively small scale of our experiments: we saw a pairwise accuracy of 82.2%, with 79.3% class-based accuracy ($k = 10$). We believe that the main reason for the generally lower performance in Chinese (as compared to English) is the modest size of the corpus, though our simplistic character bigram term extraction technique may also play a role. As mentioned, smaller seed sets do not seem to be an issue. Interestingly, the class-based accuracy is 10.8% lower if no reference word is used to calibrate the divide between formal and informal, suggesting a rather biased corpus (towards informality); in English, by comparison, the reference-word normalization had a slightly negative effect on the LSA results, though the effect mostly disappeared after hybridization. The obvious next step is to integrate a Chinese word segmenter, and use a larger corpus. We could also try word count methods, though finding appropriate (balanced) resouces similar to the BNC might be a challenge; (mixed) blog corpora, on the other hand, are easily collected.

6 Conclusion

In this work, we have experimented with a number of different methods and source corpora for determining the formality level of lexical items, with the implicit goal of distinguishing the formality of near-synonym pairs. Our methods show marked improvement over simple word-length metrics; when multiple sources of information, i.e. word length, word counts, and word association, are integrated, we are able to reach over 85% performance on the near-synonym task, and close to 100% accuracy when comparing words with extreme formality differences; our voting methods show that even higher precision is possible. We have also demonstrated that our LSA word association method can be applied to a language where word length is not an appropriate metric of formality, though the results here are preliminary. Other potential future work includes addressing a wider range of phenomena, for instance assigning formality scores to morphological elements, syntactic cues, and multi-word expressions, and demonstrating that a formality lexicon can be usefully applied to other NLP tasks.

Acknowledgements

This work was supported by Natural Sciences and Engineering Research Council of Canada. Thanks to Paul Cook for his ICWSM corpus API.

References

Biber, Douglas. 1995. *Dimensions of Register Variation: A cross-linguistic comparison*. Cambridge University Press.

Brooke, Julian, Tong Wang, and Graeme Hirst. 2010. Inducing lexicons of formality from corpora. In *Proceedings of the Language Resources and Evaluation Conference (LREC '10), Workshop on Methods for the automatic acquisition of Language Resources and their evaluation methods*.

Burfoot, Clint and Timothy Baldwin. 2009. Automatic satire detection: Are you having a laugh? In *Proceedings of the Joint Conference of the 47th Annual Meeting of the Association for Computationsl Linguistics and the 4th International Joint Conference on Nautral Language Processing of the Asian Federation of Natural Language Processing (ACL-IJCNLP '09), Short Papers*, Singapore.

Burnard, Lou. 2000. User reference guide for British National Corpus. Technical report, Oxford University.

Burton, Kevin, Akshay Java, and Ian Soboroff. 2009. The ICWSM 2009 Spinn3r Dataset. In *Proceedings of the Third Annual Conference on Weblogs and Social Media (ICWSM 2009)*, San Jose, CA.

Carter, Ronald. 1998. *Vocabulary: applied linguistic perspectives*. Routledge, London.

Emigh, William and Susan C. Herring. 2005. Collaborative authoring on the web: A genre analysis of online encyclopedias. In *Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS '05)*.

Esuli, Andrea and Fabrizio Sebastiani. 2006. SentiWordNet: A publicly available lexical resource for opinion mining. In *Proceedings of the 5th Internation Conference on Language Resources and Evaluation(LREC)*, Genova, Italy.

Fellbaum, Christiane, editor. 1998. *WordNet: An Electronic Lexical Database*. The MIT Press.

Francis, Nelson and Henry Kučera. 1982. *Frequency Analysis of English Usage: Lexicon and Grammar*. Houghton Mifflin, Boston.

Godfrey, J.J., E.C. Holliman, and J. McDaniel. 1992. Switchboard: telephone speech corpus for research and development. *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1:517–520.

Hayakawa, S.I., editor. 1994. *Choose the Right Word*. HarperCollins Publishers, second edition. Revised by Eugene Ehrlich.

Heylighen, Francis and Jean-Marc Dewaele. 2002. Variation in the contextuality of language: An empirical measure. *Foundations of Science*, 7(3):293–340.

Hovy, Eduard H. 1990. Pragmatics and natural language generation. *Artificial Intelligence*, 43:153–197.

Inkpen, Diana and Graeme Hirst. 2006. Building and using a lexical knowledge base of near-synonym differences. *Computational Linguistics*, 32(2):223–262.

Karlgren, Jussi and Douglas Cutting. 1994. Recognizing text genres with simple metrics using discriminant analysis. In *Proceedings of the 15th Conference on Computational Linguistics*, pages 1071–1075.

Kessler, Brett, Geoffrey Nunberg, and Hinrich Schütze. 1997. Automatic detection of text genre. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, pages 32–38.

Landauer, Thomas K. and Susan Dumais. 1997. A solution to Plato's problem: The latent semantic analysis theory of the acquisition, induction, and representation of knowledge. *Psychological Review*, 104:211–240.

Leckie-Tarry, Helen. 1995. *Language Context: a functional linguistic theory of register*. Pinter.

Rao, Delip and Deepak Ravichandra. 2009. Semi-supervised polarity lexicon induction. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Lingusitics*, Athens, Greece.

Stone, Philip J., Dexter C. Dunphy, Marshall S. Smith, and Daniel M. Ogilivie. 1966. *The General Inquirer: A Computer Approach to Content Analysis*. MIT Press.

Taboada, Maite and Kimberly Voll. 2006. Methods for creating semantic orientation dictionaries. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC)*, Genova, Italy.

Turney, Peter and Michael Littman. 2003. Measuring praise and criticism: Inference of semantic orientation from association. *ACM Transactions on Information Systems*, 21:315–346.

Witten, Ian H. and Eibe Frank. 2005. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, San Francisco.

Toward Qualitative Evaluation of Textual Entailment Systems

Elena Cabrio
FBK-Irst, University of Trento
cabrio@fbk.eu

Bernardo Magnini
FBK-Irst
magnini@fbk.eu

Abstract

This paper presents a methodology for a quantitative and qualitative evaluation of Textual Entailment systems. We take advantage of the decomposition of Text Hypothesis pairs into *monothematic pairs*, i.e. pairs where only one linguistic phenomenon at a time is responsible for entailment judgment, and propose to run TE systems over such datasets. We show that several behaviours of a system can be explained in terms of the correlation between the accuracy on monothematic pairs and the accuracy on the corresponding original pairs.

1 Introduction

Since 2005, Recognizing Textual Entailment (RTE) has been proposed as a task whose aim is to capture major semantic inference needs across applications in Computational Linguistics (Dagan et al., 2009). Systems are asked to automatically judge whether the meaning of a portion of text, referred as Text (*T*), entails the meaning of another text, referred as Hypothesis (*H*). This evaluation provides useful cues for researchers and developers aiming at the integration of TE components in larger applications (see, for instance, the use of a TE engine in the QALL-ME project system[1], the use in relation extraction (Romano et al., 2006), and in reading comprehension systems (Nielsen et al., 2009)).

Although the RTE evaluations showed progresses in TE technologies, we think that there is still large room for improving qualitative analysis of both the RTE datasets and the system results. In particular, we intend to focus this paper on the following aspects:

1. There is relatively poor analysis of the linguistic phenomena that are relevant for the RTE datasets, and very little is known about the distribution of such phenomena, and about the ability of participating systems to correctly detect and judge them in *T,H* pairs. Experiments like the ablation tests attempted in the last RTE-5 campaign on lexical and lexical-syntactic resources go in this direction, although the degree of comprehension is still far from being optimal.

2. We are interested in the correlations among the capability of a system to address single linguistic phenomena in a pair and the ability to correctly judge the pair itself. Despite the strong intuition about such correlation (i.e. the more the phenomena for which a system is trained, the better the final judgment), no empirical evidences support it.

3. Although the ability to detect and manage single phenomena seems to be a crucial feature of high performing systems, very little is known about how systems manage to combine such results in a global score for a pair. The mechanism underlying such composition may shed light on meaning composition related to TE tasks.

4. Finally, we are interested in the relation between the above mentioned items over the different kinds of pairs represented in RTE

[1] http://qallme.fbk.eu/

Coling 2008: Poster Volume, pages 99–107,
Beijing, August 2010

datasets, specifically *entailment*, *contradiction* and *unknown* pairs. In this case the intuition is that some phenomena are more relevant for a certain judgment rather than for another.

To address the issues above, we propose an evaluation methodology aiming at providing a number of quantitative and qualitative indicators about a TE system. The method is based on the decomposition of *T,H* pairs into *monothematic pairs*, each representing one single linguistic phenomenon relevant for entailment judgment. Evaluation is carried out both on the original *T,H* pair and on the monothematic pairs originated from it. We define a correlation index between the accuracy of the system on the original *T,H* pairs and the accuracy on the corresponding monothematic pairs. We investigate the use of such correlations on different subsets of the evaluation dataset (i.e. positive vs negative pairs) and we try to induce regular patterns of evaluation.

The method we propose has been tested on a sample of 60 pairs, each decomposed in the corresponding monothematic pairs, and using two systems that obtained similar performances in RTE-5. We show that the main features and differences of these systems come to light when evaluated using qualitative criteria. Futhermore, we compare such systems with two different baseline systems, the first one performing Word Overlap, while the second one is an ideal system that knows *a priori* the probability of a linguistic phenomenon to be associated with a certain entailment judgement.

The paper is structured as follows. Section 2 explains the procedure for the creation of monothematic pairs starting from RTE pairs. Section 3 presents the evaluation methodology we propose, while Section 4 describes our pilot study. Section 5 concludes the paper and proposes future developments.

2 Decomposing RTE pairs

Our proposal on qualitative evaluation takes advantage of previous work on specialized entailment engines and monothematic datasets. A *monothematic pair* is defined (Magnini and Cabrio, 2009) as a *T,H* pair in which a certain phenomenon relevant to the entailment relation is highlighted and isolated. The main idea is to create the monothematic pairs basing on the phenomena that are actually present in the original RTE pairs, so that the actual distribution of the linguistic phenomena involved in the entailment relation emerges.

For the decomposition procedure, we refer to the methodology described in (Bentivogli et al., 2010), consisting of a number of steps carried out manually. The starting point is a *[T,H]* pair taken from one of the RTE data sets, that should be decomposed in a number of monothematic pairs $[T, H_i]$, where T is the original Text and H_i are the Hypotheses created for each linguistic phenomenon relevant for judging the entailment relation in *[T,H]*. In details, the procedure for the creation of monothematic pairs is composed of the following steps:

1. Individuate the phenomena contributing to the entailment decision in *[T,H]*.

2. For each linguistic phenomenon *i*:

   (a) Detect a general entailment rule r_i for *i*, and instantiate it using the part of T expressing *i* as the left hand side (LHS) of the rule, and information from H on *i* as the right side (RHS).

   (b) substitute the portion of T that matches the LHS of r_i with the RHS of r_i.

   (c) consider the result of the previous step as H_i, and compose the monothematic pair $[T, H_i]$. Mark the pair with phenomenon *i*.

3. Assign an entailment judgment to each monothematic pair.

Relevant linguistic phenomena are grouped using both fine-grained categories and broader categories, defined referring to widely accepted classifications in the literature (e.g. (Garoufi, 2007)) and to the inference types typically addressed in RTE systems: *lexical*, *syntactic*, *lexical-syntactic*, *discourse* and *reasoning*. Each macro category includes fine-grained phenomena (Table 2 lists the phenomena detected in RTE-5 datasets).

Text snippet (pair 125)			Phenomena	Judg.
T		**Mexico's new president, Felipe Calderon**, seems to be doing all the right things in cracking down on Mexico's drug traffickers. [...]		
H		Felipe Calderon is the outgoing President of Mexico.	lexical:semantic-opposition syntactic:argument-realization, syntactic:apposition	C
	H1	Mexico's **outgoing** president, Felipe Calderon, seems to be doing all all the right things in cracking down on Mexico's drug traffickers. [...]	lexical:semantic-opposition	C
	H2	**The new president of Mexico**, Felipe Calderon, seems to be doing all the right things in cracking down on Mexico's drug traffickers. . [...]	syntactic:argument-realization	E
	H3	**Felipe Calderon is** Mexico's **new president**.	syntactic:apposition	E

Table 1: Application of the decomposition methodology to an original RTE pair

Table 1 shows an example of the decomposition of a RTE pair (marked as *contradiction*) into monothematic pairs. At step 1 of the methodology both the phenomena that preserve the entailment and those that break the entailment rules causing a contradiction in the pair are detected, i.e. argument realization, apposition and semantic opposition (column *phenomena* in the table). While the monothematic pairs created basing on the first two phenomena preserve the entailment, the semantic opposition generates a contradiction (column *judgment*). As an example, let's apply step by step the procedure to the phenomenon of semantic opposition. At step 2a of the methodology the general rule:

Pattern: x $\Leftarrow$ / $\Rightarrow$ y
Constraint: *semantic opposition(y,x)*

is instantiated (*new*$\Leftarrow$ / $\Rightarrow$*outgoing*), and at step 2b the substitution in T is carried out (*Mexico's outgoing president, Felipe Calderon [...]*). At step 2c a negative monothematic pair T, H_1 is composed (column *text snippet* in the table) and marked as *semantic opposition* (macro-category *lexical*), and the pair is judged as *contradiction*.

3 Evaluation methodology

Aim of the evaluation methodology we propose is to provide quantitative and qualitative indicators about the behaviours of actual TE systems.

3.1 General Method

The basic assumption of the evaluation methodology is that the more a system is able to correctly solve the linguistic phenomena underlying the entailment relation separately, the more the system should be able to correctly judge more complex pairs, in which different phenomena are present and interact in a complex way. Such assumption is motivated by the notion of meaning compositionality, according to which the meaning of a complex expression *e* in a language *L* is determined by the structure of *e* in *L* and the meaning of the constituents of *e* in *L* (Frege, 1892). In a parallel way, we assume that it is possible to understand the entailment relation of a *T,H* pair (i.e. to correctly judge the *entailment/contradiction* relation) only if all the phenomena contributing to such relation are solved.

According to such assumption, we expect that the higher the accuracy of a system on the monothematic pairs and the compositional strategy, the better its performances on the original RTE pairs. Furthermore, the precision a system gains on single phenomena should be maintained over the general dataset, thanks to suitable mechanisms of meaning combination.

Given a dataset composed of original RTE pairs $[T, H]$, a dataset composed of all the monothematic pairs derived from it $[T, H]_{mono}$, and a TE system *S*, the evaluation methodology we propose consists of the following steps:

1. Run *S* both on $[T, H]$ and on $[T, H]_{mono}$, to obtain the accuracies of *S* both on the RTE original and on monothematic pairs;

2. Extract data concerning the behaviour of *S* on each phenomenon or on categories of phenomena, and calculate separate accuracies. This way it is possible to evaluate how much a system is able to correctly deal with single or with categories of phenomena;

3. Calculate the correlation between the ability of the system to correctly judge the monothematic pairs of $[T, H]_{mono}$ with respect to the

ability to correctly judge the original ones in $[T, H]$. Such correlation is expressed through a *Correlation Index (CI)*, as defined in Section 3.2;

4. In order to check if the same *CI* is maintained over both entailment and contradiction pairs (i.e. to verify if the system has peculiar strategies to correctly assign both judgments, and if the high similarity of monothematic pairs does not bias its behaviour), we calculate a *Deviation Index (DI)* as the difference between the *CIs* on entailment and on contradiction pairs, as explained in more details in Section 3.3.

3.2 Correlation Index (CI)

As introduced before, we assume that the accuracy obtained on $[T, H]_{mono}$ should positively correlate with the accuracy obtained on $[T, H]$. We define a *Correlation Index* as the ratio between the accuracy of the system on the original RTE dataset and the accuracy obtained on the monothematic dataset, as follows:

$$CI = \frac{acc[T, H]}{acc[T, H]_{mono}} \quad (1)$$

We expect the correlation index of an optimal ideal system (or the human goldstandard) to be equal to 1, i.e. 100% accuracy on the monothematic dataset should correspond to 100% accuracy on the original RTE dataset. For this reason, we consider $CI = 1$ as the ideal correlation, and we calculate the difference between such ideal CI and the correlation obtained for a system *S*.

Given such expectations, CI_S can assume three different configurations with respect to the upperbound (i.e. the ideal correlation):

- $CI_S \cong 1$ (ideal correlation): When CI_S approaches to 1, the system shows high correlation with the ideal behaviour assumed by the compositionality principle. As a consequence, we can predict that improving single modules will correspondingly affect the global performance.

- $CI_S < 1$ (missing correlation): The system is not able to exploit the ability in solving single phenomena to correctly judge the original RTE pairs. This may be due to the fact that the system does not adopt suitable combination mechanisms and loses the potentiality shown by its performances on monothematic pairs.

- $CI_S > 1$ (over correlation): The system does not exploit the ability to solve single linguistic components to solve the whole pairs, and has different mechanisms to evaluate the entailment. Probably, such a system is not intended to be modularized.

Beside this "global" correlation index calculated on the complete RTE data and on all the monothematic pairs created from it, the *CI* can also be calculated *i)* on categories of phenomena, to verify which phenomena a system is more able to solve both when isolated and when interacting with other phenomena, e.g. :

$$CI_{lex} = \frac{acc[T, H]_{lex}}{acc[T, H]_{mono-lex}} \quad (2)$$

including in $[T, H]_{lex}$ all the pairs in which at least one lexical phenomenon is present and contribute to the entailment/contradiction judgments, and in $[T, H]_{mono-lex}$ all the monothematic pairs in which a lexical phenomenon is isolated; or *ii)* on kind of judgment (*entailment*, *contradiction*, *unknown*), allowing deeper qualitative analysis of the performances of a system.

3.3 Deviation Index (DI)

We explained that a low *CI* (i.e. < 1) of a system reflects the inability to correctly exploit the potentially promising results obtained on monothematic pairs to correctly judge RTE pairs. Actually, it could also be the case that the system does not perform a correct combination because even the results got on the monothematic pairs were due to chance (e.g. a word overlap system performs well on monothematic pairs because of the high similarity between *T* and *H*, and not because it has linguistic strategies).

We detect such cases by decomposing the evaluation datasets, separating positive (i.e. *entailment*) from negative (i.e. *contradiction*, *unknown*) examples both in $[T, H]$ and in $[T, H]_{mono}$, and

independently run *S* on the new datasets. Then, we have more fine grained evaluation patterns through which we can analyze the system behaviour.

In the ideal case, we expect to have good correlation between the accuracy obtained on the monothematic pairs and the accuracy obtained on the original ones ($0 < CI_{pos} \leq 1$ and $0 < CI_{neg} \leq 1$). On the contrary, we expect that systems either without a clear composition strategy or without strong components on specific linguistic phenomena (e.g. a word overlap system), would show a significant difference of correlation on the different datasets. More specifically, situations of *inverse correlation* on the entailment and contradiction pairs (e.g. over correlation on contradiction pairs and missing correlation on entailment pairs) may reveal that the system itself is affected by the nature of the dataset (i.e. its behaviour is biased by the high similarity of $[T, H]_{mono}$), and weaknesses in the ability of solving phenomena that more frequently contribute to the assignment of a contradiction (or an entailment) judgment come to light.

We formalize such intuition defining a *Deviation Index (DI)* as the difference between the correlation indexes, respectively, on entailment and contradiction/unknown pairs, as follows:

$$|DI| = CI_{pos} - CI_{neg} \quad (3)$$

For instance, an high Deviation Index due to a missing correlation on positive entailment pairs and an over correlation for negative pairs, is interpreted as an evidence that the system has low accuracy on $[T, H]_{mono}$ - *T* and *H* are very similar and the system has no strategies to understand that the phenomenon that is present has to be judged as contradictory -, and a higher accuracy on $[T, H]$, probably due to chance. In the ideal case $DI_S \cong 0$, since we assumed the ideal *CIs* on both positive and negative examples to be as close as possible to 1 (see Section 3.2).

4 Experiments and discussion

This Section describes the experimental setup of our pilot study, carried out using two systems that took part in RTE-5 i.e EDITS and VENSES. We show the results obtained and the qualitative analysis performed basing on the proposed evaluation methodology. Their respective *CIs* and *DIs* are compared with two baselines: a word overlap system, and a system biased by the knowledge of the probability that a linguistic phenomenon contributes to the assignment of a certain entailment judgment.

4.1 Dataset

The evaluation method has been tested on a dataset composed of 60 pairs from RTE-5 test set ($[T, H]_{RTE5-sample}$, composed of 30 *entailment*, and 30 *contradiction* randomly extracted examples), and a dataset composed of all the monothematic pairs derived by the first one following the procedure described in Section 2. The second dataset $[T, H]_{RTE5-mono}$ is composed of 167 pairs (135 *entailment*, 32 *contradiction* examples, considering 35 different linguistic phenomena)[2]. On average, 2.78 monothematic pairs have been created from the original pairs. In this pilot study we decided to limit our analysis to entailment and contradiction pairs since, as observed in (Bentivogli et al., 2010), in most of the unknown pairs no linguistic phenomena relating T to H could be detected.

4.2 TE systems

EDITS The EDITS system (Edit Distance Textual Entailment Suite)[3] (Negri et al., 2009) assumes that the distance between *T* and *H* is a characteristic that separates the positive pairs, for which entailment holds, from the negative pairs, for which entailment does not hold (two way task). It is based on edit distance algorithms, and computes the *[T,H]* distance as the overall cost of the edit operations (i.e. *insertion*, *deletion* and *substitution*) required to transform *T* into *H*. For our experiments we applied the model that produced EDITS best run at RTE-5 (acc. on test set: 60.2%). The main features are: Tree Edit Distance algorithm on the parsed trees of *T* and *H*, Wikipedia lexical entailment rules, and PSO optimized operation costs (Mehdad et al., 2009).

[2] http://hlt.fbk.eu/en/Technology/TE-_Specialized_Data

[3] Available as open source at http://edits.fbk.eu/

VENSES The other system used in our experiments is VENSES[4] (Delmonte et al., 2009), that obtained performances similar to EDITS at RTE-5 (acc. on test set: 61.5%). It applies a linguistically-based approach for semantic inference, and is composed of two main components: *i)* a grammatically-driven subsystem validates the well-formedness of the predicate-argument structure and works on the output of a deep parser producing augmented head-dependency structures; and *ii)* a subsystem detects allowed logical and lexical inferences basing on different kind of structural transformations intended to produce a semantically valid meaning correspondence. Also in this case, we applied the best configuration of the system used in RTE-5.

Baseline system 1: Word Overlap algorithm The first baseline applies a Word Overlap (WO) algorithm on tokenized text. The threshold to separate positive from negative pairs has been learnt on the whole RTE-5 training dataset.

Baseline system 2: Linguistic biased system The second baseline is produced by a more sophisticated but biased system. It exploits the probability of linguistic phenomena to contribute more to the assignment of a certain judgment than to another. Such probabilities are learnt on the $[T, H]_{RTE5-mono}$ goldstandard: given the list of the phenomena with their frequency in monothematic positive and negative pairs (columns 1,2,3 of Table 2), we calculate the probability *P* of phenomenon *i* to appear in a positive (or in a negative) pair as follows:

$$P(i|[T,H]_{positive}) = \frac{\#(i|[T,H]_{RTE5-positive-mono})}{\#(i|[T,H]_{RTE5-mono})} \quad (4)$$

For instance, if the phenomenon *apposition* appears in 11 monothematic positive pairs and in 6 negative pairs, it has a probability of 64.7% to appear in positive examples and 35.3% to appear in negative ones. Such knowledge is then stored in the system, and is used in the classification phase, assigning the most probable judgment associated to a certain phenomenon.

[4]http://project.cgm.unive.it/venses_en.html

When applied to $[T, H]_{RTE5-sample}$, this system uses a simple combination strategy: if phenomena associated with different judgments are present in a pair, and one phenomenon is associated with a contradiction judgment with a probability $> 50\%$, the pair is marked as *contradiction*, otherwise it is marked as *entailment*.

4.3 Results

Following the methodology described in Section 3, at step 1 we run EDITS and VENSES on $[T, H]_{RTE5-sample}$, and on $[T, H]_{RTE5-mono}$ (Table 3 reports the accuracies obtained).

At step 2, we calculate the accuracy of EDITS and VENSES on each single linguistic phenomenon, and on categories of phenomena. Table 2 shows the distribution of the phenomena on the dataset, reflected in the number of positive and negative monothematic pairs created for each phenomenon. As can be seen, some phenomena appear more frequently than others (e.g. *coreference*, *general inference*). Furthermore, some linguistic phenomena allow only the creation of positive or negative examples, while others can contribute to the assignment of both judgments. Due to the small datasets we used, some phenomena appear rarely; the accuracy on them cannot be considered completely reliable.

Nevertheless, from these data the main features of the systems can be identified. For instance, EDITS obtains the highest accuracy on positive monothematic pairs, while it seems it has no peculiar strategies to deal with phenomena causing contradiction (e.g. *semantic opposition*, and *quantity mismatching*). On the contrary, VENSES shows an opposite behaviour, obtaining the best results on the negative cases.

At step 3 of the proposed evaluation methodology, we calculate the correlation index between the ability of the system to correctly judge the monothematic pairs of $[T, H]_{RTE5-mono}$ with respect to the ability to correctly judge the original ones in $[T, H]_{RTE5-sample}$.

Table 3 compares EDITS and VENSES *CI* with the two baseline systems described before. As can be noticed, even if EDITS *CI* outperforms the WO system, they show a similar behaviour (high accuracy on monothematic pairs, and much lower

phenomena	# $[T, H]$ $_{RTE5-mono}$		EDITS % acc.		VENSES % acc.	
	pos.	neg.	pos.	neg.	pos.	neg.
lex:identity	1	3	100	0	100	33.3
lex:format	2	-	100	-	100	-
lex:acronymy	3	-	100	-	33.3	-
lex:demonymy	1	-	100	-	100	-
lex:synonymy	11	-	90.9	-	90.9	-
lex:semantic-opp.	-	3	-	0	-	100
lex:hypernymy	3	-	100	-	66.6	-
lex:geo-knowledge	1	-	100	-	100	-
TOT lexical	22	6	95.4	0	77.2	66.6
lexsynt:transp-head	2	-	100	-	50	-
lexsynt:verb-nom.	8	-	87.5	-	25	-
lexsynt:causative	1	-	100	-	100	-
lexsynt:paraphrase	3	-	100	-	66.6	-
TOT lex-syntactic	14	-	92.8	-	42.8	-
synt:negation	-	1	-	0	-	0
synt:modifier	3	1	100	0	33.3	100
synt:arg-realization	5	-	100	-	40	-
synt:apposition	11	6	100	33.3	54.5	83.3
synt:list	1	-	100	-	100	-
synt:coordination	3	-	100	-	33.3	-
synt:actpass-altern.	4	2	100	0	25	50
TOT syntactic	28	9	96.4	22.2	42.8	77.7
disc:coreference	20	-	95	-	50	-
disc:apposition	3	-	100	-	0	-
disc:anaphora-zero	5	-	80	-	20	-
disc:ellipsis	4	-	100	-	25	-
disc:statements	1	-	100	-	0	-
TOT discourse	33	-	93.9	-	36.3	-
reas:apposition	2	1	100	0	50	100
reas:modifier	3	-	66.6	-	100	-
reas:genitive	1	-	100	-	100	-
reas:relative-clause	1	-	100	-	0	-
reas:elliptic-expr.	1	-	100	-	0	-
reas:meronymy	1	1	100	0	100	0
reas:metonymy	3	-	100	-	33.3	-
reas:representat.	1	-	100	-	0	-
reas:quantity	-	5	-	0	-	80
reas:spatial	1	-	100	-	0	-
reas:gen-inference	24	10	87.5	50	37.5	90
TOT reasoning	38	17	89.4	35.2	42.1	82.3
TOT (all phenom)	135	32	93.3	25	45.9	81.2

Table 2: Systems' accuracy on phenomena

on the RTE sample). According to our definition, their *CIs* ($0 < CI < 1$) show a good ability of the systems to deal with linguistic phenomena when isolated, but a scarce ability in combining them to assign the final judgment. EDITS *CI* is not far from the *CI* of the linguistic biased baseline system, even if we were expecting a higher *CI* for the latter system. The reason is that beside the linguistic phenomena that allow only the creation of negative monothematic pairs, all the phenomena that allow both judgments have a higher probability to contribute to the creation of positive monothematic pairs.

Comparing the *CI* of the four analyzed systems with the ideal correlation ($CI_S \cong 1$, see Section 3.2), VENSES is the closest one ($\Delta = 0.15$), even if it shows a light over correlation (probably due to the nature of the dataset). The second closest

	acc. % $_{RTE5-sample}$	acc. % $_{RTE5-mono}$	*CI*
EDITS	58.3	80.8	0.72
VENSES	60	52.6	1.15
Word Overlap	38.3	77.24	0.49
ling baseline	68.3	86.8	0.79

Table 3: Evaluation on RTE pairs and on monothematic pairs

		categories of linguistic phenomena				
	RTE5 data	lex.	lex-synt.	synt.	disc.	reas.
EDITS	$_{sample}$	47.8	64.3	51.7	75	62.5
	$_{mono}$	75	92.8	78.3	93.9	72.7
	CI	**0.63**	**0.69**	**0.66**	**0.79**	**0. 85**
VENSES	$_{sample}$	47.2	42.8	62	46.4	67.5
	$_{mono}$	75	42.8	51.3	33	54.5
	CI	**0.62**	**1**	**1.2**	**1.4**	**1.23**
WO baseline	$_{sample}$	36.3	57.1	34.4	50	35
	$_{mono}$	78.5	71.4	72.9	96.9	69
	CI	**0.46**	**0.79**	**0.47**	**0.51**	**0.5**
ling-biased baseline	$_{sample}$	82.6	92.8	58.6	82.1	70
	$_{mono}$	96.4	100	75.6	96.9	80
	CI	**0.85**	**0.92**	**0.77**	**0.84**	**0.87**

Table 4: Evaluation on categories of phenomena

one is the linguistic biased system ($\Delta = 0.21$), showing that the knowledge of the most probable judgment assigned to a certain phenomenon can be a useful information.

Table 4 reports an evaluation of the four systems on categories of linguistic phenomena.

To check if the same *CI* is maintained over both entailment and contradiction pairs, we calculate a *Deviation Index* as the difference between the *CIs* on entailment and on contradiction pairs (step 4 of our methodology). As described in Section 3, we created four datasets dividing both $[T, H]_{RTE5-sample}$ and $[T, H]_{RTE5-mono}$ into positive (i.e. *entailment*) and negative (i.e. *contradiction*) examples. We run EDITS and VENSES on the datasets and we calculate the *CI* on positive and on negative examples separately. If we obtained missing correlation between the accuracy on the monothematic pairs and the accuracy on RTE original ones, it would mean that the potentiality that the systems show on monothematic pairs is not exploited to correctly judge more complex pairs, therefore compositional mechanisms should be improved.

Table 5 shows that the *DIs* of the linguistic biased system and of VENSES are close to the ideal case ($DI_S \cong 0$), indicating a good capacity to correctly differentiate entailment from contradiction cases. EDITS results demonstrate that the

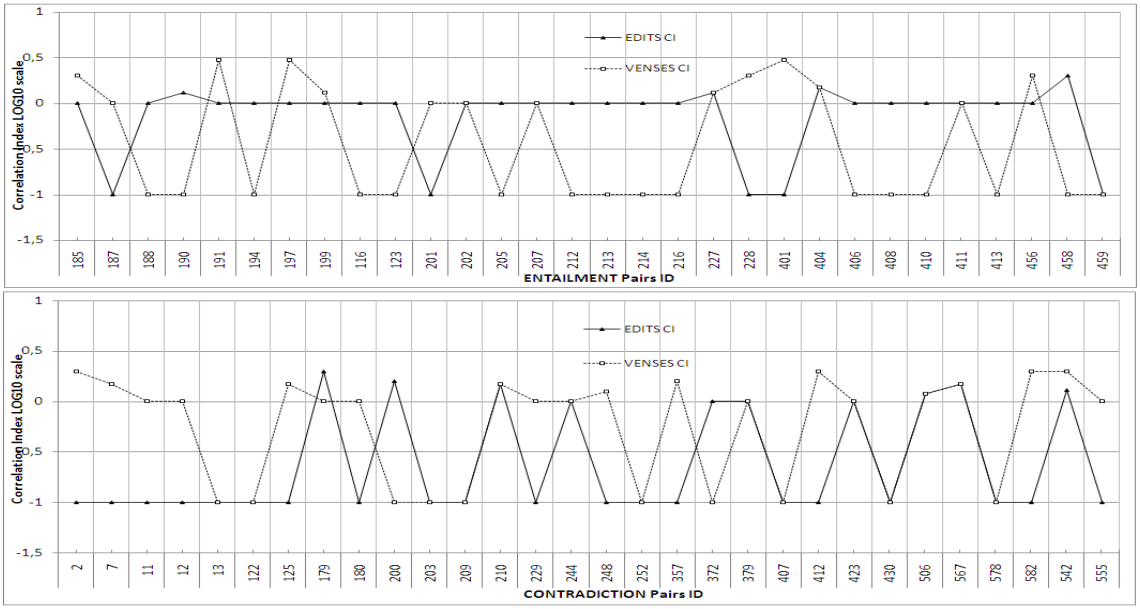


Figure 1: Correlation Index on entailment and contradiction pairs for EDITS and VENSES

		% acc. $_{RTE5\ sample}$	% acc. $_{RTE5\ mono}$	*CI*	*DI*
EDITS	E	83.3	94.7	0.88	0.5
	C	33.3	24	1.38	
VENSES	E	50	47.01	1.08	0.16
	C	70	75.7	0.92	
WO	E	50	88	0.56	0.24
baseline	C	26.6	33	0.80	
ling-biased	E	96.6	98.5	0.98	0.03
baseline	C	40	39.4	1.01	

Table 5: Evaluation on entail. and contr. pairs

shallow approach implemented by the system has no strategies to correctly judge negative examples (similarly to the WO system), therefore should be mainly improved with this respect.

We also calculated the CI for every pair of the dataset, putting into relation each original pair with all the monothematic pairs derived from it. Figure 1 shows EDITS and VENSES's *CI* on each pair of our sample.[5] Even if the systems obtained similar performances in the challenge, the second system seems to behave in an opposite way with respect to EDITS, showing higher *CI* for negative cases than for the positive ones.

[5]The ideal case CI=1 corresponds to 0 on the logarithmic scale.

5 Conclusion and Future work

We have proposed a methodology for the evaluation of TE systems based on the analysis of the system behaviour on monothematic pairs with respect to the behaviour on corresponding original pairs. Through the definition of two indicators, a Correlation Index and a Deviation Index, we infer evaluation patterns which indicate strength and weaknesses of the system. As a pilot study, we have compared two systems that took part in RTE-5. We discovered that, although the two systems have similar accuracy on RTE-5 datasets, they show significant differences in their respective abilities to manage different linguistic phenomena and to properly combine them. We hope that the analysis provided by our methodology may bring interesting elements both to TE system developers and for deep discussion on the nature of TE itself.

As future work, we plan to refine the evaluation methodology introducing the possibility to assign different relevance to the phenomena.

6 Acknowledgements

Thanks to Professor Rodolfo Delmonte and to Sara Tonelli for running the VENSES system on our data sets.

References

Bentivogli, Luisa, Bernardo Magnini, Ido Dagan, Hoa Trang Dang, and Danilo Giampiccolo. 2009. The Fifth PASCAL Recognizing Textual Entailment Challenge. *Proceedings of the TAC 2009 Workshop on Textual Entailment*. Gaithersburg, Maryland. 17 November.

Bentivogli, Luisa, Elena Cabrio, Ido Dagan, Danilo Giampiccolo, Medea Lo Leggio, and Bernardo Magnini. 2010. Building Textual Entailment Specialized Data Sets: a Methodology for Isolating Linguistic Phenomena Relevant to Inference. *Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC)* . Valletta, Malta. 19-21 May.

Dagan, Ido, Bill Dolan, Bernardo Magnini, and Dan Roth. 2009. Recognizing textual entailment: Rational, evaluation and approaches. *Natural Language Engineering (JNLE)*, Volume 15, Special Issue 04, October 2009, pp i-xvii. Cambridge University Press.

Delmonte, Rodolfo, Sara Tonelli, Rocco Tripodi. 2009. Semantic Processing for Text Entailment with VENSES. *Proceedings of the TAC 2009 Workshop on Textual Entailment*. To appear. Gaithersburg, Maryland. 17 November.

Garoufi, Konstantina. 2007. Towards a Better Understanding of Applied Textual Entailment. *Master Thesis*. Saarland University. Saarbrücken, Germany.

Gottlob, Frege. 1892. *Über Sinn und Bedeutung*. Zeitschrift für Philosophie und philosophische Kritik. 100.25-50.

Magnini, Bernardo, and Elena Cabrio. 2009. Combining Specialized Entailment Engines. *Proceedings of the 4th Language & Technology Conference (LTC '09)*. Poznan, Poland. 6-8 November.

Mehdad, Yashar, Matteo Negri, Elena Cabrio, Milen Kouylekov, and Bernardo Magnini. 2009. Using Lexical Resources in a Distance-Based Approach to RTE. *Proceedings of the TAC 2009 Workshop on Textual Entailment*. Gaithersburg, Maryland. 17 November.

Negri, Matteo, Milen Kouylekov, Bernardo Magnini, Yashar Mehdad, and Elena Cabrio. 2009. Towards Extensible Textual Entailment Engines: The EDITS Package. *AI*IA 2009: Emergent Perspectives in Artificial Intelligence, Lecture Notes in Computer Science*. Volume 5883. ISBN 978-3-642-10290-5. Springer-Verlag Berlin Heidelberg, p. 314.

Nielsen, Rodney D., Wayne Ward, and James H. Martin. 2009. Recognizing entailment in intelligent tutoring systems. In Ido Dagan, Bill Dolan, Bernardo Magnini and Dan Roth (Eds.) *The Journal of Natural Language Engineering, (JNLE)*. , 15, pp 479-501. Copyright Cambridge University Press, Cambridge, United Kingdom.

Romano, Lorenza, Milen Ognianov Kouylekov, Idan Szpektor, Ido Kalman Dagan, and Alberto Lavelli, 2006. Investigating a Generic Paraphrase-Based Approach for Relation Extraction. *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2006)*. Trento, Italy. 3-7 April.

Benchmarking of Statistical Dependency Parsers for French

Marie Candito⋆, Joakim Nivre◇, Pascal Denis⋆ and Enrique Henestroza Anguiano⋆
⋆ Alpage (Université Paris 7/INRIA)
◇ Uppsala University, Department of Linguistics and Philology
marie.candito@linguist.jussieu.fr {pascal.denis, henestro}@inria.fr joakim.nivre@lingfil.uu.se

Abstract

We compare the performance of three statistical parsing architectures on the problem of deriving typed dependency structures for French. The architectures are based on PCFGs with latent variables, graph-based dependency parsing and transition-based dependency parsing, respectively. We also study the influence of three types of lexical information: lemmas, morphological features, and word clusters. The results show that all three systems achieve competitive performance, with a best labeled attachment score over 88%. All three parsers benefit from the use of automatically derived lemmas, while morphological features seem to be less important. Word clusters have a positive effect primarily on the latent variable parser.

1 Introduction

In this paper, we compare three statistical parsers that produce typed dependencies for French. A syntactic analysis in terms of typed grammatical relations, whether encoded as functional annotations in syntagmatic trees or in labeled dependency trees, appears to be useful for many NLP tasks including question answering, information extraction, and lexical acquisition tasks like collocation extraction.

This usefulness holds particularly for French, a language for which bare syntagmatic trees are often syntactically underspecified because of a rather free order of post-verbal complements/adjuncts and the possibility of subject inversion. Thus, the annotation scheme of the French Treebank (Abeillé and Barrier, 2004) makes use of flat syntagmatic trees without VP nodes, with no structural distinction between complements, adjuncts or post-verbal subjects, but with additional functional annotations on dependents of verbs.

Parsing is commonly enhanced by using more abstract lexical information, in the form of morphological features (Tsarfaty, 2006), lemmas (Seddah et al., 2010), or various forms of clusters (see (Candito and Seddah, 2010) for references). In this paper, we explore the integration of morphological features, lemmas, and linear context clusters.

Typed dependencies can be derived using many different parsing architectures. As far as statistical approaches are concerned, the dominant paradigm for English has been to use constituency-based parsers, the output of which can be converted to typed dependencies using well-proven conversion procedures, as in the Stanford parser (Klein and Manning, 2003). In recent years, it has also become popular to use statistical dependency parsers, which are trained directly on labeled dependency trees and output such trees directly, such as MSTParser (McDonald, 2006) and MaltParser (Nivre et al., 2006). Dependency parsing has been applied to a fairly broad range of languages, especially in the CoNLL shared tasks in 2006 and 2007 (Buchholz and Marsi, 2006; Nivre et al., 2007).

We present a comparison of three statistical parsing architectures that output typed dependencies for French: one constituency-based architecture featuring the Berkeley parser (Petrov et al., 2006), and two dependency-based systems using radically different parsing methods, MSTParser (McDonald et al., 2006) and MaltParser (Nivre et al., 2006). These three systems are compared both in terms of parsing accuracy and parsing times, in realistic settings that only use predicted information. By using freely available software packages that implement language-independent approaches

Coling 2008: Poster Volume, pages 108–116,
Beijing, August 2010

and applying them to a language different from English, we also hope to shed some light on the capacity of different methods to cope with the challenges posed by different languages.

Comparative evaluation of constituency-based and dependency-based parsers with respect to labeled accuracy is rare, despite the fact that parser evaluation on typed dependencies has been advocated for a long time (Lin, 1995; Carroll et al., 1998). Early work on statistical dependency parsing often compared constituency-based and dependency-based methods with respect to their *unlabeled* accuracy (Yamada and Matsumoto, 2003), but comparison of different approaches with respect to *labeled* accuracy is more recent.

Cer et al. (2010) present a thorough analysis of the best trade-off between speed and accuracy in deriving Stanford typed dependencies for English (de Marneffe et al., 2006), comparing a number of constituency-based and dependency-based parsers on data from the Wall Street Journal. They conclude that the highest accuracy is obtained using constituency-based parsers, although some of the dependency-based parsers are more efficient.

For German, the 2008 ACL workshop on parsing German (Kübler, 2008) featured a shared task with two different tracks, one for constituency-based parsing and one for dependency-based parsing. Both tracks had their own evaluation metrics, but the accuracy with which parsers identified subjects, direct objects and indirect objects was compared across the two tracks, and the results in this case showed an advantage for dependency-based parsing.

In this paper, we contribute results for a third language, French, by benchmarking both constituency-based and dependency-based methods for deriving typed dependencies. In addition, we investigate the usefulness of morphological features, lemmas and word clusters for each of the different parsing architectures. The rest of the paper is structured as follows. Section 2 describes the French Treebank, and Section 3 describes the three parsing systems. Section 4 presents the experimental evaluation, and Section 5 contains a comparative error analysis of the three systems. Section 6 concludes with suggestions for future research.

2 Treebanks

For training and testing the statistical parsers, we use treebanks that are automatically converted from the French Treebank (Abeillé and Barrier, 2004) (hereafter FTB), a constituency-based treebank made up of $12,531$ sentences from the *Le Monde* newspaper. Each sentence is annotated with a constituent structure and words bear the following features: gender, number, mood, tense, person, definiteness, wh-feature, and clitic case. Nodes representing dependents of a verb are labeled with one of 8 grammatical functions.[1]

We use two treebanks automatically obtained from FTB, both described in Candito et al. (2010). FTB-UC is a modified version of the original constituency-based treebank, where the rich morphological annotation has been mapped to a simple tagset of 28 part-of-speech tags, and where compounds with regular syntax are broken down into phrases containing several simple words while remaining sequences annotated as compounds in FTB are merged into a single token. Function labels are appended to syntactic category symbols and are either used or ignored, depending on the task.

FTB-UC-DEP is a dependency treebank derived from FTB-UC using the classic technique of head propagation rules, first proposed for English by Magerman (1995). Function labels that are present in the original treebank serve to label the corresponding dependencies. The remaining unlabeled dependencies are labeled using heuristics (for dependents of non-verbal heads). With this conversion technique, output dependency trees are necessarily projective, and extracted dependencies are necessarily local to a phrase, which means that the automatically converted trees can be regarded as pseudo-projective approximations to the correct dependency trees (Kahane et al., 1998). Candito et al. (2010) evaluated the converted trees for 120 sentences, and report a 98% labeled attachment score when comparing the automatically converted dependency trees to the manually corrected ones.

[1] These are `SUJ` (subject), `OBJ` (object), `A-OBJ`/`DE-OBJ` (indirect object with preposition *à* / *de*), `P-OBJ` (indirect object with another preposition / locatives), `MOD` (modifier), `ATS`/`ATO` (subject/object predicative complement).

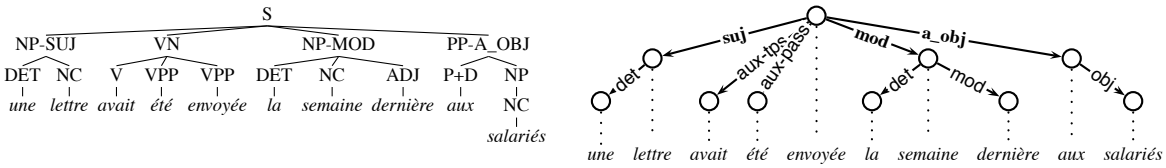


Figure 1: An example of constituency tree of the FTB-UC (left), and the corresponding dependency tree (right) for *A letter had been sent the week before to the employees.*

Figure 1 shows two parallel trees from FTB-UC and FTB-UC-DEP. In all reported experiments in this paper, we use the usual split of FTB-UC: first 10% as test set, next 10% as dev set, and the remaining sentences as training set.

3 Parsers

Although all three parsers compared are statistical, they are based on fairly different parsing methodologies. The Berkeley parser (Petrov et al., 2006) is a latent-variable PCFG parser, MSTParser (McDonald et al., 2006) is a graph-based dependency parser, and MaltParser (Nivre et al., 2006) is a transition-based dependency parser.

The choice to include two different dependency parsers but only one constituency-based parser is motivated by the study of Seddah et al. (2009), where a number of constituency-based statistical parsers were evaluated on French, including Dan Bikel's implementation of the Collins parser (Bikel, 2002) and the Charniak parser (Charniak, 2000). The evaluation showed that the Berkeley parser had significantly better performance for French than the other parsers, whether measured using a parseval-style labeled bracketing F-score or a CoNLL-style unlabeled attachment score. Contrary to most of the other parsers in that study, the Berkeley parser has the advantage of a strict separation of parsing model and linguistic constraints: linguistic information is encoded in the treebank only, except for a language-dependent suffix list used for handling unknown words.

In this study, we compare the Berkeley parser to MSTParser and MaltParser, which have the same separation of parsing model and linguistic representation, but which are trained directly on labeled dependency trees. The two dependency parsers use radically different parsing approaches but have achieved very similar performance for a wide range of languages (McDonald and Nivre, 2007). We describe below the three architectures in more detail.[2]

3.1 The Berkeley Parser

The Berkeley parser is a freely available implementation of the statistical training and parsing algorithms described in (Petrov et al., 2006) and (Petrov and Klein, 2007). It exploits the fact that PCFG learning can be improved by splitting symbols according to structural and/or lexical properties (Klein and Manning, 2003). Following Matsuzaki et al. (2005), the Berkeley learning algorithm uses EM to estimate probabilities on symbols that are automatically augmented with latent annotations, a process that can be viewed as symbol splitting. Petrov et al. (2006) proposed to score the splits in order to retain only the most beneficial ones, and keep the grammar size manageable: the splits that induce the smallest losses in the likelihood of the treebank are merged back. The algorithm starts with a very general treebank-induced binarized PCFG, with order h horizontal markovisation. created, where at each level a symbol appears without track of its original siblings. Then the Berkeley algorithm performs split/merge/smooth cycles that iteratively refine the binarized grammar: it adds two latent annotations on each symbol, learns probabilities for the refined grammar, merges back 50% of the splits, and smoothes the final probabilities to prevent overfitting. All our experiments are run using BerkeleyParser 1.0,[3] modified for handling

[2]For replicability, models, preprocessing tools and experimental settings are available at `http://alpage.inria.fr/statgram/frdep.html`.

[3]`http://www.eecs.berkeley.edu/\~petrov/berkeleyParser`

French unknown words by Crabbé and Candito (2008), with otherwise default settings (order 0 horizontal markovisation, order 1 vertical markovisation, 5 split/merge cycles).

The Berkeley parser could in principle be trained on functionally annotated phrase-structure trees (as shown in the left half of figure 1), but Crabbé and Candito (2008) have shown that this leads to very low performance, because the splitting of symbols according to grammatical functions renders the data too sparse. Therefore, the Berkeley parser was trained on FTB-UC without functional annotation. Labeled dependency trees were then derived from the phrase-structure trees output by the parser in two steps: *(1)* function labels are assigned to phrase structure nodes that have functional annotation in the FTB scheme; and *(2)* dependency trees are produced using the same procedure used to produce the pseudo-gold dependency treebank from the FTB (cf. Section 2).

The functional labeling relies on the Maximum Entropy labeler described in Candito et al. (2010), which encodes the problem of functional labeling as a multiclass classification problem. Specifically, each class is of the eight grammatical functions used in FTB, and each head-dependent pair is treated as an independent event. The feature set used in the labeler attempt to capture bilexical dependencies between the head and the dependent (using stemmed word forms, parts of speech, etc.) as well as more global sentence properties like mood, voice and inversion.

3.2 MSTParser

MSTParser is a freely available implementation of the parsing models described in McDonald (2006). These models are often described as *graph-based* because they reduce the problem of parsing a sentence to the problem of finding a directed maximum spanning tree in a dense graph representation of the sentence. Graph-based parsers typically use global training algorithms, where the goal is to learn to score correct trees higher than incorrect trees. At parsing time a global search is run to find the highest scoring dependency tree. However, unrestricted global inference for graph-based dependency parsing is NP-hard, and graph-based parsers like MSTParser therefore limit the scope of their features to a small number of adjacent arcs (usually two) and/or resort to approximate inference (McDonald and Pereira, 2006). For our experiments, we use MSTParser 0.4.3b[4] with 1-best projective decoding, using the algorithm of Eisner (1996), and second order features. The labeling of dependencies is performed as a separate sequence classification step, following McDonald et al. (2006).

To provide part-of-speech tags to MSTParser, we use the MElt tagger (Denis and Sagot, 2009), a Maximum Entropy Markov Model tagger enriched with information from a large-scale dictionary.[5] The tagger was trained on the training set to provide POS tags for the dev and test sets, and we used 10-way jackknifing to generate tags for the training set.

3.3 MaltParser

MaltParser[6] is a freely available implementation of the parsing models described in (Nivre, 2006) and (Nivre, 2008). These models are often characterized as *transition-based*, because they reduce the problem of parsing a sentence to the problem of finding an optimal path through an abstract transition system, or state machine. This is sometimes equated with shift-reduce parsing, but in fact includes a much broader range of transition systems (Nivre, 2008). Transition-based parsers learn models that predict the next state given the current state of the system, including features over the history of parsing decisions and the input sentence. At parsing time, the parser starts in an initial state and greedily moves to subsequent states – based on the predictions of the model – until a terminal state is reached. The greedy, deterministic parsing strategy results in highly efficient parsing, with run-times often linear in sentence length, and also facilitates the use of arbitrary non-local features, since the partially built dependency tree is fixed in any given state. However, greedy inference can also lead to error propagation if early predictions place the parser in incorrect states. For the experiments in this paper, we use MaltParser

[4] `http://mstparser.sourceforge.net`

[5] Denis and Sagot (2009) report a tagging accuracy of 97.7% (90.1% on unknown words) on the FTB-UC test set.

[6] `http://www.maltparser.org`

1.3.1 with the arc-eager algorithm (Nivre, 2008) and use linear classifiers from the LIBLINEAR package (Fan et al., 2008) to predict the next state transitions. As for MST, we used the MElt tagger to provide input part-of-speech tags to the parser.

4 Experiments

This section presents the parsing experiments that were carried out in order to assess the state of the art in labeled dependency parsing for French and at the same time investigate the impact of different types of lexical information on parsing accuracy. We present the features given to the parsers, discuss how they were extracted/computed and integrated within each parsing architecture, and then summarize the performance scores for the different parsers and feature configurations.

4.1 Experimental Space

Our experiments focus on three types of lexical features that are used either in addition to or as substitutes for word forms: morphological features, lemmas, and word clusters. In the case of MaltParser and MSTParser, these features are used in conjunction with POS tags. Motivations for these features are rooted in the fact that French has a rather rich inflectional morphology.

The intuition behind using morphological features like tense, mood, gender, number, and person is that some of these are likely to provide *additional cues* for syntactic attachment or function type. This is especially true given that the 29 tags used by the MElt tagger are rather coarse-grained.

The use of lemmas and word clusters, on the other hand, is motivated by *data sparseness* considerations: these provide various degrees of generalization over word forms. As suggested by Koo et al. (2008), the use of word clusters may also reduce the need for annotated data.

All our features are automatically produced: no features except word forms originate from the treebank. Our aim was to assess the performance currently available for French in a realistic setting.

Lemmas Lemmatized forms are extracted using Le*fff* (Sagot, 2010), a large-coverage morphosyntactic lexicon for French, and a set of heuristics for unknown words. More specifically, Le*fff* is queried for each $(word, pos)$, where pos is the tag predicted by the MElt tagger. If the pair is found, we use the longest lemma associated with it in Le*fff*. Otherwise, we rely on a set of simple stemming heuristics using the form and the predicted tag to produce the lemma. We use the form itself for all other remaining cases.[7]

Morphological Features Morphological features were extracted in a way similar to lemmas, again by querying Le*fff* and relying on heuristics for out-of-dictionary words. Here are the main morphological attributes that were extracted from the lexicon: mood and tense for verbs; person for verbs and pronouns; number and gender for nouns, past participles, adjectives and pronouns; whether an adverb is negative; whether an adjective, pronoun or determiner is cardinal, ordinal, definite, possessive or relative. Our goal was to predict all attributes found in FTB that are recoverable from the word form alone.

Word Form Clusters Koo et al. (2008) have proposed to use unsupervised word clusters as features in MSTParser, for parsing English and Czech. Candito and Crabbé (2009) showed that, for parsing French with the Berkeley parser, using the same kind of clusters as substitutes for word forms improves performance. We now extend their work by comparing the impact of such clusters on two additional parsers.

We use the word clusters computed by Candito and Crabbé (2009) using Percy Liang's implementation[8] of the Brown unsupervised clustering algorithm (Brown et al., 1992). It is a bottom-up hierarchical clustering algorithm that uses a bigram language model over clusters. The resulting cluster ids are bit-strings, and various levels of granularity can be obtained by retaining only the first x bits. Candito and Crabbé (2009) used the *L'Est Républicain* corpus, a 125 million word journalistic corpus.[9] To reduce lexi-

[7] Candito and Seddah (2010) report the following coverage for the Le*fff*: around 96% of the tokens, and 80.1% of the token types are present in the Le*fff* (leaving out punctuation and numeric tokens, and ignoring case differences).

[8] `http://www.eecs.berkeley.edu/~pliang/software`

[9] `http://www.cnrtl.fr/corpus/estrepublicain`

cal data sparseness caused by inflection, they ran a lexicon-based stemming process on the corpus that removes inflection marks without adding or removing lexical ambiguity. The Brown algorithm was then used to compute 1000 clusters of stemmed forms, limited to forms that appeared at least 20 times.

We tested the use of clusters with different values for two parameters: **nbbits** = the cluster prefix length in bits, to test varying granularities, and **minocc** = the minimum number of occurrences in the *L'Est Républicain* corpus for a form to be replaced by a cluster or for a cluster feature to be used for that form.

4.2 Parser-Specific Configurations

Since the three parsers are based on different machine learning algorithms and parsing algorithms (with different memory requirements and parsing times), we cannot integrate the different features described above in exactly the same way. For the Berkeley parser we use the setup of Candito and Seddah (2010), where additional information is encoded within symbols that are used as substitutes for word forms. For MaltParser and MSTParser, which are based on discriminative models that permit the inclusion of interdependent features, additional information may be used either in addition to or as substitutes for word forms. Below we summarize the configurations that have been explored for each parser:

- **Berkeley:**
  1. **Morphological features:** N/A.
  2. **Lemmas:** Concatenated with POS tags and substituted for word forms.
  3. **Clusters:** Concatenated with morphological suffixes and substituted for word forms; grid search for optimal values of **nbbits** and **minocc**.
- **MaltParser** and **MSTParser:**
  1. **Morphological features:** Added as features.
  2. **Lemmas:** Substituted for word forms or added as features.
  3. **Clusters:** Substituted for word forms or added as features; grid search for optimal values of **nbbits** and **minocc**.

4.3 Results

Table 1 summarizes the experimental results. For each parser we give results on the development set for the baseline (no additional features), the best configuration for each individual feature type, and the best configuration for any allowed combination of the three features types. For the final test set, we only evaluate the baseline and the best combination of features. Scores on the test set were compared using a χ^2-test to assess statistical significance: unless specified, all differences therein were significant at $p \leq 0.01$.

The MSTParser system achieves the best labeled accuracy on both the development set and the test set. When adding lemmas, the best configuration is to use them as substitutes for word forms, which slightly improves the UAS results. For the clusters, their use as substitutes for word forms tends to degrade results, whereas using them as features alone has almost no impact. This means that we could not replicate the positive effect[10] reported by Koo et al. (2008) for English and Czech. However, the best combined configuration is obtained using lemmas instead of words, a reduced set of morphological features,[11] and clusters as features, with minocc=$50,000$ and nbbits=10.

MaltParser has the second best labeled accuracy on both the development set and the test set, although the difference with Berkeley is not significant on the latter. MaltParser has the lowest unlabeled accuracy of all three parsers on both datasets. As opposed to MSTParser, all three feature types work best for MaltParser when used in addition to word forms, although the improvement is statistically significant only for lemmas and clusters. Again, the best model uses all three types of features, with cluster features minocc=600 and nbbits=7. MaltParser shows the smallest discrepancy from unlabeled to labeled scores. This might be because it is the only architecture where labeling is directly done as part of parsing.

[10] Note that the two experiments cannot be directly compared. Koo et al. (2008) use their own implementation of an MST parser, which includes extra second-order features (e.g. grand-parent features on top of sibling features).

[11] As MSTParser training is memory-intensive, we removed the features containing information already encoded part-of-speech tags.

	Development Set										Test Set			
	Baseline		**Morpho**		**Lemma**		**Cluster**		**Best**		**Baseline**		**Best**	
Parser	LAS	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS	UAS
Berkeley	85.1	89.3	–	–	85.9	90.0	86.5	90.8	86.5	**90.8**	85.6	89.6	86.8	**91.0**
MSTParser	**87.2**	**90.0**	87.2	90.2	87.2	90.1	87.2	90.1	**87.5**	90.3	87.6	90.3	**88.2**	90.9
MaltParser	86.2	89.0	86.3	89.0	86.6	89.2	86.5	89.2	86.9	89.4	86.7	89.3	87.3	89.7

Table 1: Experimental results for the three parsing systems. LAS=labeled accuracy, UAS=unlabeled accuracy, for sentences of any length, ignoring punctuation tokens. Morpho/Lemma/Cluster=best configuration when using morphological features only (resp. lemmas only, clusters only), Best=best configuration using any combination of these.

For Berkeley, the lemmas improve the results over the baseline, and its performance reaches that of MSTParser for unlabeled accuracy (although the difference between the two parsers is not significant on the test set). The best setting is obtained with clusters instead of word forms, using the full bit strings. It also gives the best unlabeled accuracy of all three systems on both the development set and the test set. For the more important labeled accuracy, the point-wise labeler used is not effective enough.

Overall, MSTParser has the highest labeled accuracy and Berkeley the highest unlabeled accuracy. However, results for all three systems on the test set are roughly within one percentage point for both labeled and unlabeled accuracy, which means that we do not find the same discrepancy between constituency-based and dependency-based parser that was reported for English by Cer et al. (2010).

Table 2 gives parsing times for the best configuration of each parsing architecture. MaltParser runs approximately 9 times faster than the Berkeley system, and 10 times faster than MSTParser. The difference in efficiency is mainly due to the fact that MaltParser uses a linear-time parsing algorithm, while the other two parsers have cubic time complexity. Given the rather small difference in labeled accuracy, MaltParser seems to be a good choice for processing very large corpora.

	Bky	**Malt**	**MST**
Tagging	_	0:27	0:27
Parsing	12:19	0:58 (0:18)	14:12 (12:44)
Func. Lab.	0:23	_	_
Dep. Conv.	0:4	_	_
Total	12:46	1:25	14:39

Table 2: Parsing times (min:sec) for the dev set, for the three architectures, on an imac 2.66GHz. The figures within brackets show the pure parsing time without the model loading time, when available.

5 Error Analysis

We provide a brief analysis of the errors made by the best performing models for Berkeley, MSTParser and MaltParser on the development set, focusing on labeled and unlabeled attachment for nouns, prepositions and verbs. For nouns, Berkeley has the best unlabeled attachment, followed by MSTParser and then MaltParser, while for labeled attachment Berkeley and MSTParser are on a par with MaltParser a bit behind. For prepositions, MSTParser is by far the best for both labeled and unlabeled attachment, with Berkeley and MaltParser performing equally well on unlabeled attachment and MaltParser performing better than Berkeley on labeled attachment.[12] For verbs, Berkeley has the best performance on both labeled and unlabeled attachment, with MSTParser and MaltParser performing about equally well. Although Berkeley has the best unlabeled attachment overall, it also has the worst labeled attachment, and we found that this is largely due to the functional role labeler having trouble assigning the correct label when the dependent is a preposition or a clitic.

For errors in attachment as a function of word distance, we find that precision and recall on dependencies of length > 2 tend to degrade faster for MaltParser than for MSTParser and Berkeley,

[12] In the dev set, for MSTParser, 29% of the tokens that do not receive the correct governor are prepositions (883 out of 3051 errors), while these represent 34% for Berkeley (992 out of 2914), and 30% for MaltParser (1016 out of 3340).

with Berkeley being the most robust for dependencies of length > 6. In addition, Berkeley is best at finding the correct root of sentences, while MaltParser often predicts more than one root for a given sentence. The behavior of MSTParser and MaltParser in this respect is consistent with the results of McDonald and Nivre (2007).

6 Conclusion

We have evaluated three statistical parsing architectures for deriving typed dependencies for French. The best result obtained is a labeled attachment score of 88.2%, which is roughly on a par with the best performance reported by Cer et al. (2010) for parsing English to Stanford dependencies. Note two important differences between their results and ours: First, the Stanford dependencies are in a way deeper than the surface dependencies tested in our work. Secondly, we find that for French there is no consistent trend favoring either constituency-based or dependency-based methods, since they achieve comparable results both for labeled and unlabeled dependencies.

Indeed, the differences between parsing architectures are generally small. The best performance is achieved using MSTParser, enhanced with predicted part-of-speech tags, lemmas, morphological features, and unsupervised clusters of word forms. MaltParser achieves slightly lower labeled accuracy, but is probably the best option if speed is crucial. The Berkeley parser has high accuracy for unlabeled dependencies, but the current labeling method does not achieve a comparably high labeled accuracy.

Examining the use of lexical features, we find that predicted lemmas are useful in all three architectures, while morphological features have a marginal effect on the two dependency parsers (they are not used by the Berkeley parser). Unsupervised word clusters, finally, give a significant improvement for the Berkeley parser, but have a rather small effect for the dependency parsers.

Other results for statistical dependency parsing of French include the pilot study of Candito et al. (2010), and the work ofSchluter and van Genabith (2009), which resulted in an LFG statistical French parser. However, the latter's results are obtained on a modified subset of the FTB, and are expressed in terms of F-score on LFG f-structure features, which are not comparable to our attachment scores. There also exist a number of grammar-based parsers, evaluated on gold test sets annotated with chunks and dependencies (Paroubek et al., 2005; de la Clergerie et al., 2008). Their annotation scheme is different from that of the FTB, but we plan to evaluate the statistical parsers on the same data in order to compare the performance of grammar-based and statistical approaches.

Acknowledgments

The first, third and fourth authors' work was supported by ANR Sequoia (ANR-08-EMER-013). We are grateful to our anonymous reviewers for their comments.

References

Abeillé, A. and N. Barrier. 2004. Enriching a french treebank. In *LREC'04*.

Bikel, D. M. 2002. Design of a multi-lingual, parallel-processing statistical parsing engine. In *HLT-02*.

Brown, P., V. Della Pietra, P. Desouza, J. Lai, and R. Mercer. 1992. Class-based n-gram models of natural language. *Computational linguistics*, 18(4).

Buchholz, S. and E. Marsi. 2006. CoNLL-X shared task on multilingual dependency parsing. In *CoNLL 2006*.

Candito, M. and B. Crabbé. 2009. Improving generative statistical parsing with semi-supervised word clustering. In *IWPT'09*.

Candito, M. and D. Seddah. 2010. Parsing word clusters. In *NAACL/HLT Workshop SPMRL 2010*.

Candito, M., B. Crabbé, and P. Denis. 2010. Statistical french dependency parsing : treebank conversion and first results. In *LREC 2010*.

Carroll, J., E. Briscoe, and A. Sanfilippo. 1998. Parser evaluation: A survey and a new proposal. In *LREC 1998*.

Cer, D., M.-C. de Marneffe, D. Jurafsky, and C. Manning. 2010. Parsing to stanford dependencies: Trade-offs between speed and accuracy. In *LREC 2010*.

Charniak, E. 2000. A maximum entropy inspired parser. In *NAACL 2000*.

Crabbé, B. and M. Candito. 2008. Expériences d'analyse syntaxique statistique du français. In *TALN 2008*.

de la Clergerie, E. V., C. Ayache, G. de Chalendar, G. Francopoulo, C. Gardent, and P. Paroubek. 2008. Large scale production of syntactic annotations for french. In *First International Workshop on Automated Syntactic Annotations for Interoperable Language Resources*.

de Marneffe, M.-C., B. MacCartney, and C. D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *LREC 2006*.

Denis, P. and B. Sagot. 2009. Coupling an annotated corpus and a morphosyntactic lexicon for state-of-the-art pos tagging with less human effort. In *PACLIC 2009*.

Eisner, J. 1996. Three new probabilistic models for dependency parsing: An exploration. In *COLING 1996*.

Fan, R.-E., K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. 2008. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9.

Kahane, S., A. Nasr, and O. Rambow. 1998. Pseudo-projectivity: A polynomially parsable non-projective dependency grammar. In *ACL/COLING 1998*.

Klein, D. and C. D. Manning. 2003. Accurate unlexicalized parsing. In *ACL 2003*.

Koo, T., X. Carreras, and M. Collins. 2008. Simple semi-supervised dependency parsing. In *ACL-08:HLT*.

Kübler, S. 2008. The PaGe 2008 shared task on parsing german. In *ACL-08 Workshop on Parsing German*.

Lin, D. 1995. A dependency-based method for evaluating broad-coverage parsers. In *IJCAI-95*.

Magerman, D. M. 1995. Statistical decision-tree models for parsing. In *ACL 1995*.

Matsuzaki, T., Y. Miyao, and J. Tsujii. 2005. Probabilistic cfg with latent annotations. In *ACL 2005*.

McDonald, R. and J. Nivre. 2007. Characterizing the errors of data-driven dependency parsing models. In *EMNLP-CoNLL 2007*.

McDonald, R. and F. Pereira. 2006. Online learning of approximate dependency parsing algorithms. In *EACL 2006*.

McDonald, R., K. Lerman, and F. Pereira. 2006. Multilingual dependency analysis with a two-stage discriminative parser. In *CoNLL 2006*.

McDonald, R. 2006. *Discriminative Learning and Spanning Tree Algorithms for Dependency Parsing*. Ph.D. thesis, University of Pennsylvania.

Nivre, J., Johan Hall, and Jens Nilsson. 2006. Maltparser: A data-driven parser-generator for dependency parsing. In *LREC 2006*.

Nivre, J., J. Hall, S. Kübler, R. McDonald, J. Nilsson, S. Riedel, and D. Yuret. 2007. The CoNLL 2007 shared task on dependency parsing. In *CoNLL Shared Task of EMNLP-CoNLL 2007*.

Nivre, J. 2006. *Inductive Dependency Parsing*. Springer.

Nivre, J. 2008. Algorithms for deterministic incremental dependency parsing. *Computational Linguistics*, 34.

Paroubek, P., L.-G. Pouillot, I. Robba, and A. Vilnat. 2005. Easy : Campagne d'évaluation des analyseurs syntaxiques. In *TALN 2005, EASy workshop : campagne d'évaluation des analyseurs syntaxiques*.

Petrov, S. and D. Klein. 2007. Improved inference for unlexicalized parsing. In *NAACL-07: HLT*.

Petrov, S., L. Barrett, R. Thibaux, and D. Klein. 2006. Learning accurate, compact, and interpretable tree annotation. In *ACL 2006*.

Sagot, B. 2010. The Le*fff*, a freely available and large-coverage morphological and syntactic lexicon for french. In *LREC 2010*.

Schluter, N. and J. van Genabith. 2009. Dependency parsing resources for french: Converting acquired lfg f-structure. In *NODALIDA 2009*.

Seddah, D., M. Candito, and B. Crabbé. 2009. Cross parser evaluation and tagset variation: a french treebank study. In *IWPT 2009*.

Seddah, D., G. Chrupała, O. Cetinoglu, J. van Genabith, and M. Candito. 2010. Lemmatization and statistical lexicalized parsing of morphologically-rich languages. In *NAACL/HLT Workshop SPMRL 2010*.

Tsarfaty, R. 2006. Integrated morphological and syntactic disambiguation for modern hebrew. In *COLING/ACL 2006 Student Research Workshop*.

Yamada, H. and Y. Matsumoto. 2003. Statistical dependency analysis with support vector machines. In *IWPT 2003*.

Tree Topological Features for Unlexicalized Parsing

Samuel W. K. Chan[†] **Lawrence Y. L. Cheung**[#] **Mickey W. C. Chong**[†]

[†]Dept. of Decision Sciences
Chinese University of Hong Kong

[#]Dept. of Linguistics & Modern Languages
Chinese University of Hong Kong

{swkchan, yllcheung, mickey_chong}@cuhk.edu.hk

Abstract

As unlexicalized parsing lacks word token information, it is important to investigate novel parsing features to improve the accuracy. This paper studies a set of tree topological (TT) features. They quantitatively describe the tree shape dominated by each non-terminal node. The features are useful in capturing linguistic notions such as grammatical weight and syntactic branching, which are factors important to syntactic processing but overlooked in the parsing literature. By using an ensemble classifier-based model, TT features can significantly improve the parsing accuracy of our unlexicalized parser. Further, the ease of estimating TT feature values makes them easy to be incorporated into virtually any mainstream parsers.

1 Introduction

Many state-of-the-art parsers work with lexicalized parsing models that utilize the information and statistics of word tokens (Magerman, 1995; Collins, 1999, 2003; Charniak, 2000). The performance of lexicalized models is susceptible to vocabulary variation as lexical statistics is often corpus-specific (Ratnaparkhi, 1999; Gildea, 2001). As parsers are typically evaluated using the Penn Treebank (Marcus *et al*., 1993), which is based on financial news, the problems of lexicalized parsing could easily be overlooked. Unlexicalized models, on the other hand, are less sensitive to lexical variation and are more portable across domains. Though the performance of unlexicalized models was believed not to exceed that of lexicalized models (Klein & Manning, 2003), Petrov & Klein (2007) show that unlexicalized parsers can match lexicalized parsers in performance using the grammar rule splitting technique. Given the practical advantages and the latest development, unlexicalized parsing deserves further scrutiny.

A profitable direction of research on unlexicalized parsing is to investigate novel parsing features. This paper examines a set of what we call *tree topological* (TT) features, including phrase span, phrase height, tree skewness, etc. This study is motivated by the fact that conventional parsers rarely consider the shape of subtrees dominated by these nodes and rely primarily on matching tags. As a result, an NP with a complicated structure is treated the same as an NP that dominates only one word. However, our study shows that TT features are useful predictors of phrase boundaries, a critical ambiguity resolution issue. TT features have two more advantages. First, TT features capture linguistic properties, such as branching and grammatical "heaviness", across different syntactic structures. Second, they are easily computable without the need for extra language resources.

The organization of the paper is as follows. Section 2 reviews the features commonly used in parsing. Section 3 provides the details of TT features in the unlexicalized parser. The parser is evaluated in Section 4. In Section 5, we discuss the effectiveness and advantages of TT features in parsing and possible enhancement. This is followed by a conclusion in Section 6.

2 Related Work

2.1 Parsing Features

This section reviews major types of information in parsing.

Coling 2008: Poster Volume, pages 117–125,
Beijing, August 2010

Tags: The dominant types of information that drive parsing and chunking algorithms are POS/syntactic tags, context-free grammar (CFG) rules, and their statistical properties. Matching tags against CFG rules to form phrases is central to all basic parsing algorithms such as Cocke-Kasami-Younger (CKY) algorithm, and the Earley algorithm, and the chart parsing.

Word Token-based: Machine learning and statistical modelling emerged in the 90s as an ideal computational approach to feature-rich parsing. Classifiers can typically capitalize on a large set of features in decision making. Magerman (1995), Ratnaparkh (1999) and Charniak (2000) used classifiers to model dependencies between word pairs. They popularized the use word tokens as attributes in lexicalized parsing. Collins (1999, 2003) also integrated information like head word and distance from head into the statistical model to enhance probabilistic chart parsing. Since then, word tokens, head words and their statistical derivatives have become standard features in many parsers. Word token information is also fundamental to dependency parsing (Kübler *et al*., 2009) because dependency grammar is rooted in the idea that the head and the dependent word are related by different dependency relations.

Semantic-based: Some efforts have also been made to consider semantic features, such as sense tags, in parsing. Words are first tagged with semantic classes, often using WordNet-based resources. The lexical semantic class can be instructive to the selection of the correct parse from a set of candidate structures. It has been reported that the lexical semantics of words is effective in resolving structural ambiguity, especially PP-attachment (Black *et al*., 1992; Stetina & Nagao, 1997; Agirre *et al*., 2008). Nevertheless, the use of semantic features has still been relatively rare. They incur overheads in acquiring semantic language resources, such as sense-tagged corpora and WordNet databases. Semantic-based parsing also requires accurate sense-tagging.

Since substantial gain from tag features is unlikely in the near future and deriving semantic features is often a tremendous task, there is a pressing need to seek for new features, particularly in unlexicalized parsing.

2.2 Linguistic-motivated Features

In this section, a review of the linguistic motivation behind the TT features is provided.

Grammatical Weight: Apart from syntactic categories, linguists have long observed that the number of words (often referred to as "weight" or "heaviness") in a phrase can affect syntactic processing of sentences (Quirk *et al.*, 1985; Wasow, 1997; Rosenbach, 2005). It corresponds roughly to the span feature described in Section 3.2. The effect of grammatical weight often manifests in word order variation. Heavy NP shift, dative alternation, particle movement and extraposition in English are canonical examples where "heavy" chunks get dislocated to the end of a sentence. In his corpus analysis, Wasow (1997) found that weight is a very crucial factor in determining dative alternation. Hawkins (1994) also argued that due to processing constraints, the human syntactic processor tends to group an incoming stream of words as rapidly as possible, preferring smaller chunks on the left.

Tree Topology: CFG-based parsing approach hides the structural properties of the dominated subtree from the associated syntactic tag. Structural topology, or tree shape, however, can be useful in guiding the parser to group tags into phrases. Structures significantly deviating from left/right branching, e.g. center embedding, are much more difficult to process and rare in production (Gibson, 1998). Another example is the resolution of scope ambiguity in coordinate structures (*CS*s). *CS*s are common but notoriously difficult to parse due to scope ambiguity when the conjuncts are complex (Collins, 1999; Kübler *et al.*, 2009). One good cue to the problem is that humans prefer *CS*s with parallel internal syntactic structures (Frazier *et al.*, 2000). In a corpus-based study, Dubey *et al.* (2008) show that structural repetition across conjuncts is significantly more frequent. The implication to parsing is that preference should be given to bracketing in which conjuncts are structurally similar. TT information can inform the parser of the structural properties of phrases.

3 An Ensemble-based Parser

To accommodate a large set of features, we opt for classifier-based parsing because classifiers

can easily handle many features, as pointed out in Ratnaparkhi (1999). This is different from chart parsing models popular in many parsers (e.g. Collins, 2003) which require special statistical modelling. Our parser starts from a string of POS tags without any hints from words. As in other similar approaches (Abney 1991; Ramshaw & Marcus, 1995; Sang, 2001; Sagae & Lavie, 2005), the first and the foremost problem that has to be resolved is to identify the boundary points of phrases, without any explicit grammar rules. Here we adopt the ensemble learning technique to unveil boundary points, or *chunking points* hereafter. Two heterogeneous and mutually independent attribute feature sets are introduced in Section 3.2 and 3.3.

3.1 Basic Architecture of the Parser

Our parser has two modules, namely, a chunker and a phrase recognizer. The chunker locates the boundaries of chunks while the phrase recognizer predicts the non-terminal syntactic tag of the identified chunks, e.g. `NP`, `VP`, etc. In the chunker, we explore a new approach that aims at identifying chunk boundaries. Assume that the input of the chunker is a tag sequence $<x_0 \ldots x_n \ldots x_m>$ where $0 \le n \le m$. Let y_n be the point of focus between two consecutive tags x_n and x_{n+1}. The chunker classifies all focus points as either a chunking point or a merging point at the relevant level. A focus point y_n is a merging point if x_n and x_{n+1} are siblings of the same parent node in the target parse tree. Otherwise, y_n is a chunking point. Consider the tag sequence and the expected classification of points in the example below. Chunking points are marked with "`%`" and merging points with "`+`".

```
PRP % VBZ % DT % RB   +  JJ  % NN
He    is    a    very    nice  guy
```

The point between `RB` and `JJ` is a merging point because they are siblings of the parent node `ADJP` in the target parse tree. The point between `DT` and `RB` is a chunking point since `DT` and `RB` are not siblings and do not share the same parent node. Chunks are defined as the consecutive tag sequences not split up by `%`. When a focus point y_n is classified as a chunking point, it effectively means that no fragment preceding y_n can combine with any fragment following y_n to form a phrase, i.e. a *distituent*.

Both the chunker and the recognizer are trained using the Penn Treebank (Marcus *et al*., 1993). In addition, we adopt the ensemble technique to combine two sets of heterogeneous features. The method yields a much more accurate predictive power (Dietterich, 2000). One necessary and sufficient condition for an ensemble of classifiers to be more accurate than any of its individual members is that the classifiers must be diverse. Table 1 summaries the basic rationale of the parser. The two feature sets will be further explained in Section 3.2 and 3.3.

- Prepare training data from the Treebank based on topological & information-theoretic features
- Train the chunker and phrase recognizer using the ensemble technique
- For any input tag sequence *l*,
  WHILE *l* contains more than one element DO
    IDENTIFY the status, + or %, of each focus point in *l*
    RECOGNIZE the syntactic tag (ST) of each identified chunk
    UPDATE *l* with the new ST sequence
  ENDWHILE
- Display the parse tree

Table 1. Basic rationale of the parser

The learning module acquires the knowledge encoded in the Penn Treebank to support various classification tasks. The input tag sequence is first fed into the chunker. The phrase recognizer then analyzes the chunker's output and assigns non-terminal syntactic tags (e.g. `NP`, `VP`, etc.) to identified chunks. The updated tag sequence is fed back to the chunker for processing at the next level. The iteration continues until a complete parse is formed.

3.2 Tree Topological Feature Set

Tree topological (TT) features describe the shape of subtrees quantitatively. Our approach to addressing this problem involves examining a set of topological features, without any assumption of the word tokens. They all have been implemented for chunking.

Node Coordinates (NCs): NCs include the level of focus (*LF*) and the relative position (*RP*) of the target subtree. The level of focus is defined as the total number of levels under the target node, with the terminal level inclusive while the *RP* indicates the linear position of the target node in that level. As in Figure 1, the *LF* for

subtree *A* and *B* are the same; however, the *RP* for subtree *A* is smaller than that for subtree *B*.

Span Ratio (SR): The *SR* is defined as the total number of terminal nodes spanned under the target node and is divided by the length of the sentence. In Figure 1, the span ratio for the target node `VP` at subtree *B* is 5/12. This ratio illustrates not only how many terminal nodes are covered by the target node, but also how far the target node is from the root `S`.

Aspect Ratio (AR): The *AR* of a target node in a subtree is defined as the ratio of the total number of non-terminal nodes involved to the total number of terminal nodes spanned. The *AR* is also indicative of the average branching factor of the subtree.

Skewness Measure (SM): The *SM* estimates the degree to which the subtree leans towards either left or right. In this research, the *SM* of a subtree is evaluated by the distribution of the length of the paths connecting the target node and each terminal node it dominates. The length of a path from a target node *V* to a terminal node *T* is the number of edges between *V* and *T*. For a tree with *n* terminal nodes, there are *n* paths. A pivot is defined as the [*n*/2]th terminal node when *n* is odd and between [*n*/2]th and [(*n*+1)/2]th terminal nodes if *n* is even, where [ ] is a ceiling function. The *SM* is defined as

$$SM = \frac{1}{\sum_{\rho_i>0} \rho_i} \left(\frac{\sum_{i=1}^{n} \rho_i (x_i - \bar{x})^3}{\sigma^3} \right) \qquad \text{Eqn (1)}$$

where x_i is the length of the *i*-th path pointing to the *i*-th terminal node, $\bar{x}$ and σ are the average and standard deviation of the length of all paths at that level of focus (*LF*). ρ_i is the distance measured from the *i*-th terminal node to the pivot. The distance is positive if the terminal node is to the left of the pivot, zero if it is right at the pivot, and negative if the terminal node is to the right of the pivot. Obviously, if the lengths of all paths are the same in the tree, the numerator of Eqn (1) will be crossed out and the *SM* returns to zero. The pivot also provides an axis of vertical flipping where the *SM* still holds. The farther the terminal node from the pivot, the longer the distance. The distances ρ provide the moment factors to quantify the skewness of trees. For illustration, let us consider subtree *B* with the target node `VP` at level of focus (*LF*) = 4 in Figure 1. Since there are five terminal nodes, the pivot is at the third node `VB`. The lengths of the paths x_i from left to right in the subtree are 1, 2, 3, 4, 4 and the moment factors ρ_i for the paths are 2, 1, 0, -1, -2. Assuming that $\bar{x}$ and σ for all the trees in the Treebank at level 4 are, say, 2.9 and 1.2 respectively, then *SM* = -3.55. It implies that subtree *B* under the target node `VP` has a strong right branching tendency, even though it has a very uniform branching factor which is usually defined as the number of children at each node.

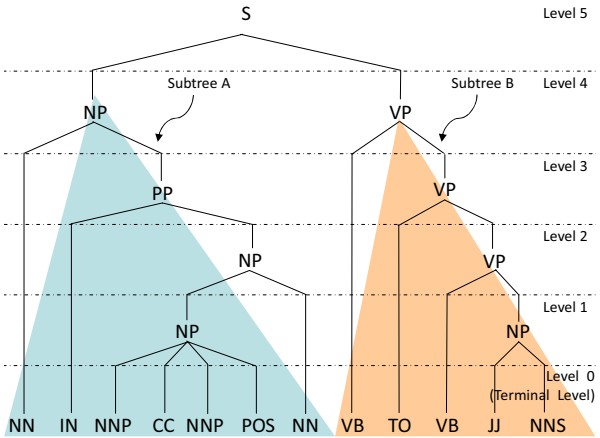


Figure 1. Two different subtrees in the sentence `S`

In our parser, to determine whether the two target nodes at level 4, i.e., `NP` and `VP`, should be merged to form a `S` at level 5 or not, an attribute vector with TT features for both `NP` and `VP` are devised as a training case. The corresponding target attribute is a binary value, i.e., chunking vs. merging. In addition, a set of *if-merged* attributes are introduced. For example, the attribute *SM-if-merged* indicates the changes of the *SM* if both target nodes are merged. This is particularly helpful since they are predictive under our bottom-up derivation strategy.

3.3 Information-Theoretic Feature Set

Context features are usually helpful in many applications of supervised language learning. In modelling context, one of the most central methodological concepts is co-occurrence. While collocation is the probabilistic co-occurrence of pure word tokens, *colligation* is defined as the co-occurrence of word tokens with grammatical patterning such as POS cate-

gories (Hunston, 2001). In this research, to capture the colligation without word tokens, a sliding window of 6 POS tags at the neighborhood of the focus point y_n is defined as our first set of context attributes. In addition, we define a set of information-theoretic (IT) attributes which reflect the likelihood of the fragment collocation. Various adjacent POS fragments around the focus point y_n are constructed, as in Table 2.

x_{n-2}	x_{n-1}	x_n	x_{n+1}	x_{n+2}	x_{n+3}	Colligation meas.
	x_{n-1}	x_n				d_1:$\zeta(x_{n-1}, x_n)$
		x_n	x_{n+1}			d_2:$\zeta(x_n, x_{n+1})$
			x_{n+1}	x_{n+2}		d_3:$\zeta(x_{n+1}, x_{n+2})$
x_{n-2}	x_{n-1}	x_n				d_4:$\zeta(x_{n-2}x_{n-1}, x_n)$
	x_{n-1}	x_n	x_{n+1}			d_5:$\zeta(x_{n-1}x_n, x_{n+1})$
		x_n	x_{n+1}	x_{n+2}		d_6:$\zeta(x_n, x_{n+1}x_{n+2})$
			x_{n+1}	x_{n+2}	x_{n+3}	d_7:$\zeta(x_{n+1}, x_{n+2}x_{n+3})$

Table 2. Colligation as context measure in various adjacent POS fragments where the focus point y_n is between x_n and x_{n+1}

An n-gram is treated as a 2-gram of an n_1-gram and an n_2-gram, where $n_1 + n_2 = n$ (Magerman & Marcus, 1990). The information-theoretic function ζ, namely, mutual information (*MI*), quantifies the co-occurrence of fragments. *MI* compares the probability of observing n_1-gram and n_2-gram together to the probability of observing them by chance (Church & Hanks, 1989). Here is an example illustrating the set of attributes. Take the point y_n between `RB` and `JJ` in Section 3.1 as an example. $d5$ represents the *MI* between (`DT RB`) and `JJ`, i.e. *MI*(`DT/RB`, `JJ`).

3.4 Multiple Classifications using Ensemble Technique

The basic idea of ensemble techniques involves considering several classification methods or multiple outputs to reach a decision. An ensemble of classifiers is a set of classifiers whose individual decisions are combined in some way, typically by weighted or un-weighted voting to classify new examples. Empirically speaking, ensembles methods deliver highly accurate classifiers by combining less accurate ones. They tend to yield better results than a single classifier in those situations when different classifiers have different error characteristics and their errors can compensate each other. Two questions need to be addressed when building and using an ensemble that integrates the predictions of several classifiers. ***First***, what data are used to train the classifiers so that the errors made by one classifier could be remedied by the other? ***Second***, how are the individual classifiers fused or integrated to produce a final ensemble prediction? As shown in the last two sections, we address the first question by introducing two heterogeneous and mutually independent attribute feature sets, namely the tree topological (TT) features and information-theoretic (IT) features. Instead of training all the features to form a single giant classifier, we produce two distinct, sometimes diversified, training sets of data to form two separate moderate classifiers, in the hope that they will produce a highly accurate prediction. The second question is addressed by employing the boosting algorithm. Boosting is an effective method that produces a very accurate prediction rule by combining rough and moderately inaccurate rules of thumb (Schapire & Singer, 2000). It generates the classifiers in an iterative way. At the early beginning, an initial base classifier using a set of training data with equal weight is first constructed. When the prediction of the base classifier differs from the expected outcome, the weight of the poorly predicted data is increased to an extent based on their misclassification rate on the preceding classifiers. As a result, the learning of the subsequent classifier will focus on learning the training data that are misclassified, or poorly predicted. This process continues until a specified number of iterations is reached or a predefined termination condition is met. The ensemble prediction is also a weighted voting process, where the weight of a classifier is based on its errors over the training data used to generate it. The first practical boosting algorithm, *AdaBoost*, was introduced by Freund & Schapire (1997), and solved many practical difficulties of the earlier boosting algorithms. Table 3 illustrates the main idea of the algorithm. Interested readers can refer to the literature for detailed discussion (Freund & Schapire, 1997; Hastie *et al.*, 2001).

Given: $(x_1, y_1),..,(x_m, y_m)$ where $x_i \in X, y_i \in Y = \{-1, +1\}$
Initialize $D_1(i) = 1/m$
For $t = 1, \ldots, T$

- Train a weak learner using distribution D_t
- Get a weak hypothesis $h_t : X \rightarrow \{-1, +1\}$ with error
  $\varepsilon_t = Pr_{i\sim Dt}[h_t(x_i) \neq y_i]$
- Choose $\alpha_t = \frac{1}{2}\ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)$
- Update:

$$D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \begin{cases} e^{-\alpha_t} & \text{if } h_t(x_i) = y_i \\ e^{\alpha_t} & \text{if } h_t(x_i) \neq y_i \end{cases}$$

$$= \frac{D_t(i)\exp(-\alpha_t y_i h_t(x_i))}{Z_t}$$

where Z_t is a normalization factor

- Output:

$$H(x) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$$

Table 3. Adaboost algorithm

4 Experimental Results

Table 4 presents some sampled statistics of the skewness measure (*SM*) of some major phrase types, which include VP, NP, S, and PP, based on Sections 2—21 of the Penn Treebank (Marcus *et al*., 1993).

VP	L2-VP	L3-VP	L4-VP	L5-VP
N	18,406	22,052	18,035	15,911
Mean	-1.022	-4.454	-4.004	-3.738
S.D.	1.018	1.406	1.438	1.405
t_{score}	284.085*	-31.483*	-17.216*	
NP	L2-NP	L3-NP	L4-NP	L5-NP
N	23,270	28,172	10,827	8,375
Mean	1.013	-1.313	-1.432	-2.171
S.D.	1.284	2.013	1.821	1.628
t_{score}	158.748*	5.609*	29.614*	
S	L2-S	L3-S	L4-S	L5-S
N	2,233	5,020	7,049	7,572
Mean	0.688	-1.825	-1.459	-1.517
S.D.	1.229	2.732	2.451	2.128
t_{score}	54.031*	-7.568*	1.523	
PP	L2-PP	L3-PP	L4-PP	L5-PP
N	53,589	11,329	11,537	5,057
Mean	-1.337	-3.322	-3.951	-3.301
S.D.	0.935	1.148	1.112	1.183
t_{score}	172.352*	42.073*	-33.173*	

Table 4. *SM* values for various phrases (* = the mean in the column is statistically significantly different from the mean in the immediately following column, with degree of freedom in all cases greater than 120)

For illustration purpose, the count of Level 2 VP subtrees, their *SM* mean and standard deviation are -1.022 and 1.018 respectively. We performed *t*-tests for difference in means between various levels, even under the same phrase type. For example, the *t* score for the difference in mean between L2-VP and L3-VP is 284.085, which indicates a strong difference in *SM* values between the two levels.

The means of all phrases beyond level 2 are negative, consistent with the fact that English is generally a right branching language. When we compare the *SM* values across phrase types, it is easy to notice that VPs and PPs have larger negative values, meaning that the skewness to the right is more prominent. Even within the same phrase type, the *SM* values may differ significantly as one moves from its current level to parent level. The *SM* offers an indicator that differentiates different phrase types with different syntactic levels. Chunkers can use this additional parameter to do chunking better.

Our parsing models were trained and tested using the Penn Treebank (Marcus *et al*., 1993). Following the convention of previous studies, we pre-processed the trees by removing NULL elements and functional tags and collapsing ADVP and PRT into ADVP. Sections 2—21 are used for training and Section 23 for testing. To evaluate the contribution of the features, five different experiments were set up, as in Table 5.

Experiment	Features involved
E1	POS tags only (=baseline)
E2	POS+IT
E3	POS+IT+TT (*node coordinates* only)
E4	POS+TT (with all features)
E5	All features in E3 & E4

Table 5. Parsing features in five experiments

E1 is the baseline experiment with tag features only. E2 and E4 include additional IT and TT features respectively. E3 and E5 are partial and full mixture of the two feature types. In the evaluation below, the chunker, phrase recognizer and parser are the same throughout the five sets of experiments. They only differ in terms of features used (i.e. E1—E5). We first study the impact of the feature sets on chunking. Five chunkers CH1—CH5 are evaluated.

Table 6 shows the training and test errors in five different chunkers in the respective experiments. All chunkers were trained using the ensemble-based learning. If one compares CH2 and CH4, it is clear that both IT and TT features

enhance sentence chunking but the gain from TT features (i.e. CH4) is much more substantial. The best chunkers (CH4 and CH5) reduce the test error rate from the baseline 4.36% to 3.25%.

Chunkers	Training error %	Test error %
CH1	1.66	4.36
CH2	1.53	4.32
CH3	0.69	3.79
CH4	0.33	3.25
CH5	0.45	3.25

Table 6. Performance of the five chunkers

Similarly, the phrase recognizer uses ensemble learning to capture the rule patterns. Instead of reading off the rules straight from a lookup table, the learning can predict the syntactic tags even when it encounters rules not covered in the treebank. Certainly, the learning allows the recognizer to take into account features more than just the tags. The error rates in training and testing are 0.09% and 0.68% respectively. The chunker and the phrase recognizer were assembled to form a parser. The features described in Table 5 were used to construct five parsers. We use the PARSEVAL measures to compare the performance as shown in Table 7.

	R	P	F	CBs	0 CBs	≤ 2 CBs
P1	78.9	77.6	78.3	1.6	48.7	76.4
P2	81.9	79.7	80.8	1.5	50.6	78.7
P3	85.1	82.8	83.4	1.4	53.3	80.2
P4	84.1	82.2	83.1	1.5	52.7	78.1
P5	84.7	83.4	84.0	1.3	54.6	80.5

Table 7. Performance of five parsers corresponding to five different experiments E1—E5

Our baseline parser (P1) actually performs quite well. With only tag features, it achieves an F-score of 78.3%. Both IT and TT features can separately enhance the parsing performance (P2 and P4). However, the gain from TT features (78.3→83.1%) is much more than that from IT features (78.3→80.8%). When the two feature sets are combined, they consistently produce better results. The best (P5) has an F-score of 84.0%. Even though the test errors in CH4 and CH5 are the same as shown in Table 6, P5 demonstrates that the cooperative effect of utilizing TT and IT features and leads to better parsing results.

5 Discussion

5.1 Tree Topology and Structures

Our study has provided a way to quantitatively capture linguists' various insights that tree topology is helpful in syntactic structure building (e.g. grammatical weight, subtree shape, etc.). The *SM* seems to capture the basic right branching property. It is noteworthy that Collins (2003) found that the parsing model that can learn the branching property of structures delivers a much better parsing performance over the one that cannot. In our case, chunkers refer to TT features to distinguish different phrase types and levels, and assign chunking points in such a way that the resulting phrases can be maximally similar to the trees in the treebank topologically. Apart from the overall accuracy, one may ask in what way TT features improve parsing. Here we provide our preliminary analysis on one syntactic construction that can be benefitted from a TT-feature-aware parser. The structure is coordinate structures (*CS*s). A practical cue is that conjuncts tend to be similar syntactically (and semantically). TT-feature-aware parsers can produce more symmetrical conjuncts. All rules of the form "XP → XP 'and' XP" were extracted from the training data.

NP	*L3 (-CS)*	*L3-(+CS)*	*L4 (-CS)*	*L4-(+CS)*
N	27,950	222	10,222	605
Mean	-1.321	-0.397	-1.448	-1.162
S.D.	2.010	2.190	1.806	2.047
t_{score}	-6.266*		-3.360*	
VP	*L3 (-CS)*	*L3-(+CS)*	*L4 (-CS)*	*L4-(+CS)*
N	21,855	197	17,711	324
Mean	-4.488	-0.628	-4.063	-0.793
S.D.	1.350	2.136	1.364	1.676
t_{score}	-25.319*		-34.908*	

Table 8. TT feature values of coordinate structures (+*CS* = node that immediately dominates a *CS*; -*CS* otherwise; * = the mean in the column is statistically significantly different from the mean in the immediately following column).

We compared the *SM* of *CS* and non-*CS* phrases using *t*-tests for mean difference. The *t*-score is calculated based on unequal sample sizes and unequal variances. As shown in Table 8, we have to reject the null hypothesis that their means of the *SM*, between phrases with and without a *CS*, are equal at $\alpha = 0.0005$ significance level. In other words, phrases with and without a *CS* are statistically different. +*CS* phrases are much more balanced with a smaller *SM* value from -0.4 to -1.2. -*CS* columns generally have a much larger *SM* value, ranging from

-1.321 to -4.488. The *SM* offers information for the chunkers to avoid over- or under-chunking conjuncts in phrases with a coordination marker (e.g. 'and').

5.2 Implications to Parsing

The findings in Section 4 indicate that the presented initial version of the *unlexicalized* parser performs on a par with the first generation *lexicalized* parsers (e.g. Magerman, 1995). The promising results have two implications. First, the integration of IT and TT features produces substantial gain over the baseline model. TT features consistently outperform IT features by a noticeable margin. To the best of our knowledge, TT features have not been systematically investigated in parsing before. The effectiveness of these new features suggests that in addition to improving algorithms, practitioners should not overlook the development of new features. Second, the implementation of TT and IT features is simple and relatively computationally inexpensive. No extra resources or complicated algorithms are needed to compute TT features. Most importantly, they are suitable to the stringent requirements of unlexicalized parsing in which no word token information is allowed. The features can be added to other parsers relatively easily without substantial changes.

5.3 Further Work

The reported parsing results pertain to the initial version of the parser. There is still room for further improvement. First, it would be interesting to integrate TT features in combination with other design features (e.g. rule splitting) into the unlexicalized parser to enhance the results. Moreover, TT features is likely to enhance lexicalized parsers too. Second, more detailed analysis of TT features can be conducted in different syntactic constructions. It is quite possible that TT features are more useful to some syntactic structures than others. TT features seem to be good cues for identifying *CS*s. It is possible to compare the outputs from parsers with and without TT features (e.g. P1 vs. P4). The contribution of TT features towards specific constructions can be estimated empirically. Third, an insight from Collins (2003) is that head words and their POS tags in lexicalized parsing can improve parsing. In unlexicalized models, one can use the head POS tag alone to approximate similar mechanism.

6 Conclusion

This paper has demonstrated that TT features give rise to substantial gain in our classifier-based unlexicalized parser. The IT features have been explored as well, though the performance gain is more moderate. TT features can be inexpensively computed and flexibly incorporated into different types of parsers. Our parsing model matches early lexicalized parsing models in performance, and has good potential to do even better with adjustment and optimization. The statistical analysis of the treebank shows that TT features are effective in capturing basic linguistic properties, such as grammatical weight and branching direction, which are overlooked in previous studies of parsing. We have also hinted how TT features may have reduced chunking errors of *CS*s by producing balanced conjuncts. Though the present study focuses on unlexicalized parsing, it is likely that TT features can contribute to accuracy enhancement in other parsing models as well.

Acknowledgments

The work described in this paper was partially supported by the grants from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project Nos. CUHK440607 and CUHK440609). We also thank Henry Lee, our computer officer, for his network support during the experiments.

References

Abney, Steven. 1991. Parsing by Chunks. In Berwick, R., Abney, S., Tenny, C. (eds.), *Principle-Based Parsing*. Kluwer Academic.

Agirre, Eneko, Timothy Baldwin, and David Martinez. 2008. Improving Parsing and PP Attachment Performance with Sense Information. In *Proceedings of the 46th Annual Meeting of the Human Language Technology Conference (HLT'08)*.

Black, Ezra, Frederick Jelinek, John Lafferty, David Magerman, Robert Mercer, and Salim Roukos. 1992. Towards History-based Grammars: Using Richer Models for Probabilistic Parsing. In *Proceedings of the 5th DARPA Speech and Natural Language Workshop*.

Charniak, Eugene. 2000. A Maximum-Entropy-Inspired Parser. In *Proceedings of the 1st Meeting of the North American Chapter of the Association of Computational Linguistics*.

Church, Kenneth. and Patrick Hanks. 1989. Word Association Norms, Mutual Information and Lexicography. In *Proceedings of the Association for Computational Linguistics 27*.

Collins, Michael. 1999. *Head-driven Statistical Models for Natural Language Parsing.* Ph.D. thesis, University of Pennsylvania, Philadelphia.

Collins, Michael. 2003. Head-Driven Statistical Models for Natural Language Parsing. *Computational Linguistics* 29 (4): 589—637.

Dieterich, Thomas G. 2000. Ensemble Methods in Machine Learning. *Lecture Notes in Computer Science*, v.1857.

Dubey, Amit, Frank Keller, and Patrick Sturt. 2008 A Probabilistic Corpus-based Model of Syntactic Parallelism. *Cognition* 109 (3): 326-344.

Frazier, Lyn, Alan Munn and Charles Clifton 2000. Processing Coordinate Structures. *Journal of Psycholinguistic Research* 29 (4): 343—370.

Freund, Yoav and Robert E. Schapire. 1997. A Decision-Theoretic Generalization of On-line Learning and an Application to Boosting. *Journal of Computer and System Sciences* 55 (1): 119—139.

Gibson, Edward. 1998. Linguistic Complexity: Locality of Syntactic Dependencies. *Cognition* 68 (1): 1—76.

Gildea, Daniel. 2001. Corpus Variation and Parser Performance. In *Proceedings of 2001 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Hastie, Trevor, Robert Tibshirani and Jerome Friedman. 2001. *The Elements of Statistical Learning*. Springer.

Hawkins, John. 1994. *A Performance Theory of Order and Constituency*. Cambridge Univ. Press.

Hunston, Susan. 2001. Colligation, Lexis, Pattern, and Text. In M. Scott and G. Thompson. (ed.), *Patterns of Text: In Honour of Michael Hoey*. Amsterdam, Philadelphia: John Benjamins.

Klein, Dan, and Christopher Manning. 2003. Accurate Unlexicalized Parsing. In *Proceedings of the 41st Meeting of the Association for Computational Linguistics*.

Kübler, Sandra, Ryan McDonald, and Joakim Nivre. 2009. *Dependency Parsing.* Morgan & Claypool Publishers.

Magerman, David. 1995. Statistical Decision-tree Models for Parsing. In *Proceedings of the 33rd Annual Meeting on Association for Computational Linguistics*.

Magerman, David, and Mitchell Marcus. 1990. Parsing a Natural Language Using Mutual Information Statistics. In *Proceedings of 8th National Conference on Artificial Intelligence (AAAI-90)*.

Marcus, Mitchell, Beatrice. Santorini, and Mary Marcinkiewicz 1993. Building a Large Annotated Corpus of English: the Penn Treebank. *Computational Linguistics* 19 (2): 313—330.

Petrov, Slav, and Dan Klein. 2007. Learning and Inference for Hierarchically Split PCFGs. In *Proceedings of the 22nd Conference on Artificial Intelligence*.

Quirk, Randolph, Sidney Greenbaum, Geoffrey Leech, and Jan Svartvik. 1985. *A Grammar of Contemporary English*. London: Longman.

Ramshaw, Lance A., and Mitchell P. Marcus. 1995. Text Chunking Using Transformation-based Learning. In *Proceedings of the 3rd Workshop on Very Large Corpora*.

Ratnaparkhi, Adwait. 1999. Learning to Parse Natural Language with Maximum Entropy Models. *Machine Learning* 34 (1-3): 151—175.

Rosenbach, Anette. 2005. Animacy versus Weight as Determinants of Grammatical Variation in English. *Language* 81 (3): 613-644.

Sagae, Kenji, and Alon Lavie. 2005. A Classifier-Based Parser with Linear Run-Time Complexity. In *Proceedings of the Ninth International Workshop on Parsing Technologies (IWPT)*.

Sang, Erik. 2001. Transforming a Chunker to a Parser. In J. Veenstra, W. Daelemans, K. Sima'an, J. Zavrel (eds.), *Computational Linguistics in the Netherlands 2000*.

Schapire, Robert E., & Yoram Singer. 2000. BoosTexter: A Boosting-based System for Text Categorization. *Machine Learning* 39 (2-3): 135—168.

Stetina, Jiri, and Nagao, Makoto. 1997. Corpus-based PP Attachment Ambiguity Resolution with a Semantic Dictionary. In *Proceedings of the 5th Workshop on Very Large Corpora*.

Wasow, Thomas. 1997. Remarks on Grammatical Weight. *Language Variation and Change* 9: 81—105.

Improving Graph-based Dependency Parsing with Decision History

Wenliang Chen[†], Jun'ichi Kazama[†], Yoshimasa Tsuruoka[†‡] and Kentaro Torisawa[†]
[†]Language Infrastructure Group, MASTAR Project, NICT
{chenwl, kazama, torisawa}@nict.go.jp
[‡]School of Information Science, JAIST
tsuruoka@jaist.ac.jp

Abstract

This paper proposes an approach to improve graph-based dependency parsing by using decision history. We introduce a mechanism that considers short dependencies computed in the earlier stages of parsing to improve the accuracy of long dependencies in the later stages. This relies on the fact that short dependencies are generally more accurate than long dependencies in graph-based models and may be used as features to help parse long dependencies. The mechanism can easily be implemented by modifying a graph-based parsing model and introducing a set of new features. The experimental results show that our system achieves state-of-the-art accuracy on the standard PTB test set for English and the standard Penn Chinese Treebank (CTB) test set for Chinese.

1 Introduction

Dependency parsing is an approach to syntactic analysis inspired by dependency grammar. In recent years, interest in this approach has surged due to its usefulness in such applications as machine translation (Nakazawa et al., 2006), information extraction (Culotta and Sorensen, 2004).

Graph-based parsing models (McDonald and Pereira, 2006; Carreras, 2007) have achieved state-of-the-art accuracy for a wide range of languages as shown in recent CoNLL shared tasks (Buchholz et al., 2006; Nivre et al., 2007). However, to make parsing tractable, these models are forced to restrict features over a very limited history of parsing decisions (McDonald and Pereira, 2006; McDonald and Nivre, 2007). Previous work showed that rich features over a wide range of decision history can lead to significant improvements in accuracy for transition-based models (Yamada and Matsumoto, 2003a; Nivre et al., 2004).

In this paper, we propose an approach to improve graph-based dependency parsing by using decision history. Here, we make an assumption: the dependency relations between words with a short distance are more reliable than ones between words with a long distance. This is supported by the fact that the accuracy of short dependencies is in general greater than that of long dependencies as reported in McDonald and Nivre (2007) for graph-based models. Our idea is to use decision history, which is made in previous scans in a bottom-up procedure, to help parse other words in later scans. In the bottom-up procedure, short dependencies are parsed earlier than long dependencies. Thus, we introduce a mechanism in which we treat short dependencies built earlier as decision history to help parse long dependencies in later stages. It can easily be implemented by modifying a graph-based parsing model and designing a set of features for the decision history.

To demonstrate the effectiveness of the proposed approach, we present experimental results on English and Chinese data. The results indicate that the approach greatly improves the accuracy and that richer history-based features indeed make large contributions. The experimental results show that our system achieves state-of-the-art accuracy on the data.

2 Motivation

In this section, we present an example to show the idea of using decision history in a dependency parsing procedure.

Suppose we have two sentences in Chinese, as shown in Figures 1 and 2, where the correct dependencies are represented by the directed links. For example, in Figure 1 the directed link from

Coling 2008: Poster Volume, pages 126–134,
Beijing, August 2010

w_3:买(bought) to w_5:书(books) mean that w_3 is the head and w_5 is the dependent. In Chinese, the relationship between clauses is often not made explicit and two clauses may simply be put together with only a comma (Li and Thompson, 1997). This makes it hard to parse Chinese sentences with several clauses.

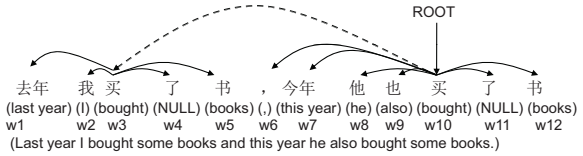


Figure 1: Example A

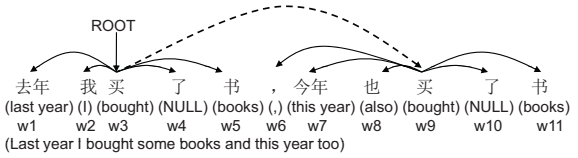


Figure 2: Example B

If we employ a graph-based parsing model, such as the model of (McDonald and Pereira, 2006; Carreras, 2007), it is difficult to assign the relations between w_3 and w_{10} in Example A and between w_3 and w_9 in Example B. For simplicity, we use w_i^A to refer to w_i of Example A and w_i^B to refer to w_i of Example B in what follows.

The key point is whether the second clauses are independent in the sentences. The two sentences are similar except that the second clause of Example A is an independent clause but that of Example B is not. w_{10}^A is the root of the second clause of Example A with subject w_8^A, while w_9^B is the root of the second clause of Example B, but the clause does not have a subject. These mean that the correct decisions are to assign w_{10}^A as the head of w_3^A and w_3^B as the head of w_9^B, as shown by the dash-dot-lines in Figures 1 and 2.

However, the model can use very limited information. Figures 3-(a) and 4-(a) show the right dependency relation cases and Figures 3-(b) and 4-(b) show the left direction cases. For the right direction case of Example A, the model has the information about w_3^A's rightmost child w_5^A and w_{10}^A's leftmost child w_6^A inside w_3^A and w_{10}^A, but it does not have information about the other children

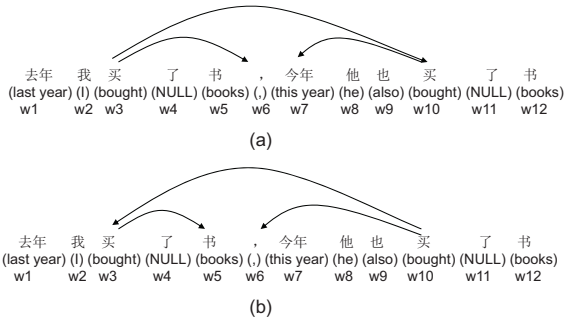


Figure 3: Example A: two directions

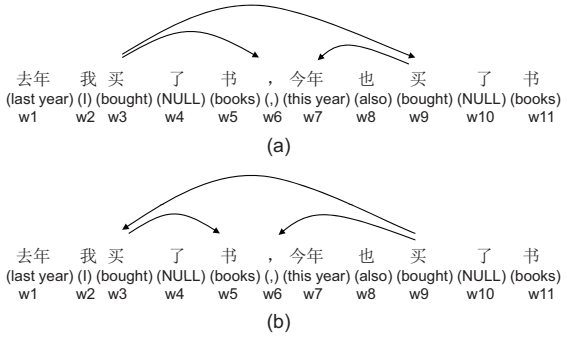


Figure 4: Example B: two directions

(such as w_8^A) of w_3^A and w_{10}^A, which may be useful for judging the relation between w_3^A and w_{10}^A. The parsing model can not find the difference between the syntactic structures of two sentences for pairs (w_3^A, w_{10}^A) and (w_3^B, w_9^B). If we can provide the information about the other children of w_3^A and w_{10}^A to the model, it becomes easier to find the correct direction between w_3^A and w_{10}^A.

Next, we show how to use decision history to help parse w_3^A and w_{10}^A of Example A.

In a bottom up procedure, the relations between the words inside [w_3^A, w_{10}^A] are built as follows before the decision for w_3^A and w_{10}^A. In the first round, we build relations for neighboring words (word distance[1]=1), such as the relations between w_3^A and w_4^A and between w_4^A and w_5^A. In the second round, we build relations for words of distance 2, and then for longer distance words until all the possible relations between the inside words are built. Figure 5 shows all the possible relations inside [w_3^A, w_{10}^A] that we can build. To simplify, we use undirected links to refer to both directions

[1]Word distance between w_i and w_j is $|j-i|$.

of dependency relations between words in the figure.

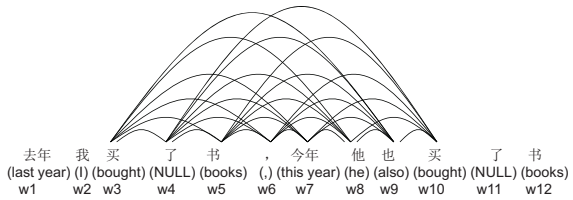


Figure 5: Example A: first step

Then given those inside relations, we choose the inside structure with the highest score for each direction of the dependency relation between w_3^A and w_{10}^A. Figure 6 shows the chosen structures. Note that the chosen structures for two directions could either be identical or different. In Figure 6-(a) and -(b), they are different.

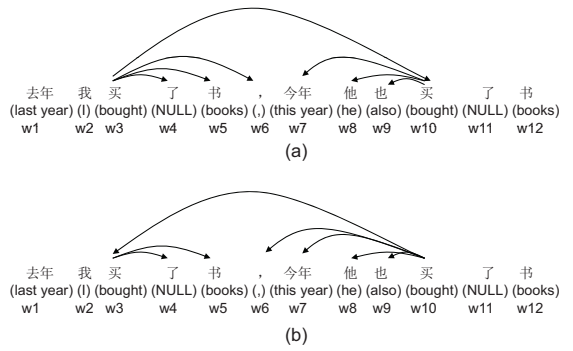


Figure 6: Example A: second step

Finally, we use the chosen structures as decision history to help parse w_3^A and w_{10}^A. For example, the fact that w_8^A is a dependent of w_{10}^A is a clue that suggests that the second clause may be independent. This results in w_{10}^A being the head of w_3^A.

This simple example shows how to use the decision history to help parse the long distance dependencies.

3 Background: graph-based parsing models

Before we describe our method, we briefly introduce the graph-based parsing models. We denote input sentence w by $w = (w_0, w_1, ..., w_n)$, where $w_0 = ROOT$ is an artificial root token inserted at the beginning of the sentence and does not depend on any other token in w and w_i refers to a word.

We employ the second-order projective graph-based parsing model of Carreras (2007), which is an extension of the projective parsing algorithm of Eisner (1996).

The parsing algorithms used in Carreras (2007) independently find the left and right dependents of a word and then combine them later in a bottom-up style based on Eisner (1996). A subtree that spans the words in $[s, t]$ (and roots at s or t) is represented by chart item $[s, t, right/left, C/I]$, where right (left) indicates that the root of the subtree is s (t) and C means that the item is *complete* while I means that the item is *incomplete* (McDonald, 2006). Here, *complete item* in the right (left) direction means that the words other than s (t) cannot have dependents outside $[s, t]$ and *incomplete item* in the right (left) direction, on the other hand, means that t (s) may have dependents outside $[s, t]$. In addition, t (s) is the direct dependent of s (t) in the incomplete item with the right (left) direction.

Larger chart items are created from pairs of smaller chart items by the bottom-up procedure. Figure 7 illustrates the cubic parsing actions of the Eisner's parsing algorithm (Eisner, 1996) in the right direction, where s, r, and t refer to the start and end indices of the chart items. In Figure 7-(a), all the items on the left side are complete and represented by triangles, where the triangle of [s, r] is complete item $[s, r, \rightarrow, C]$ and the triangle of [$r + 1$, t] is complete item $[r + 1, t, \leftarrow, C]$. Then the algorithm creates incomplete item $[s, t, \rightarrow, I]$ (trapezoid on the right side of Figure 7-(a)) by combining the chart items on the left side. This action builds the dependency from s to t. In Figure 7-(b), the item of [s, r] is incomplete and the item of [r, t] is complete. Then the algorithm creates complete item $[s, t, \rightarrow, C]$. For the left direction case, the actions are similar. Note that only the actions of creating the incomplete chart items build new dependency relations between words, while the ones of creating the complete items merge the existing structures without building new relations.

Once the parser has considered the dependency relations between words of distance 1, it goes on

to dependency relations between words of distance 2, and so on by the parsing actions. For words of distance 2 and greater, it considers every possible partition of the structures into two parts and chooses the one with the highest score for each direction. The score is the sum of the feature weights of the chart items. The features are designed over edges of dependency trees and the weights are given by model parameters (McDonald and Pereira, 2006; Carreras, 2007). We store the obtained chart items in a table. The chart item includes the information on the optimal splitting point of itself. Thus, by looking up the table, we can obtain the best tree structure (with the highest score) of any chart item.

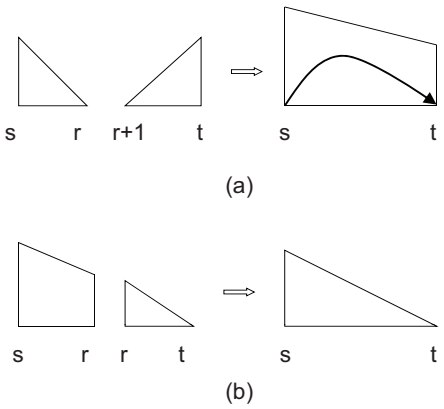


Figure 7: Cubic parsing actions of Eisner (1996)

4 Parsing with decision history

As mentioned above, the actions for creating the incomplete items build the relations between words. In this study, we only consider using history information when creating incomplete items.

4.1 Decision history

Suppose we are going to compute the scores of the relations between w_s and w_t. There are two possible directions for them.

By using the bottom-up style algorithm, the scores of the structures between words with distance $< |s-t|$ are computed in previous scans and the structures are stored in the table. We divide the decision history into two types: history-inside and history-outside. The history-inside type is the decision history made inside [s,t] and the history-outside type is the history made outside [s,t].

4.1.1 History-inside

We obtain the structure with the highest score for each direction of the dependency between w_s and w_t. Figure 8-(b) shows the best solution (with the highest score) of the left direction, where the structure is split into two parts, $[s, r_1, \rightarrow, C]$ and $[r_1 + 1, t, \leftarrow, C]$. Figure 8-(c) shows the best solution of the right case, where the structure is split into two parts, $[s, r_2, \rightarrow, C]$ and $[r_2 + 1, t, \leftarrow, C]$.

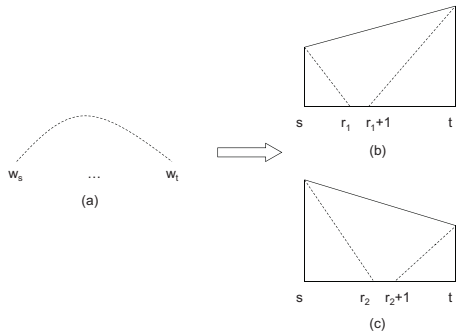


Figure 8: History-inside

By looking up the table, we have a subtree that roots at w_s on the right side of w_s and a subtree that roots at w_t on the left side of w_t. We use these structures as the information on history-inside.

4.1.2 History-outside

For history-outside, we try to obtain the subtree that roots at w_s on the left side of w_s and the one that roots at w_t on the right side of w_t. However, compared to history-inside, obtaining history-outside is more complicated because we do not know the boundaries and the proper structures of the subtrees. Here, we use an simple heuristic method to find a subtree whose root is at w_s on the left side of w_s and one whose root is at w_t on the right side of w_t.

We introduce two assumptions: 1) The structure within a sub-sentence [2] is more reliable than the one that goes across from sub-sentences. 2) More context (more words) can result in a better solution for determining subtree structures.

[2]To simplify, we split one sentence into sub-sentences with punctuation marks.

Algorithm 1 Searching for history-outside boundaries

1: Input: w, s, t
2: **for** $k = s - 1$ to 1 **do**
3: if(isPunct(w_k)) break;
4: if($s - k >= t - s - 1$) break
5: **end for**
6: $b_s = k$
7: **for** $k = t + 1$ to $|w|$ **do**
8: if(isPunct(w_k)) break;
9: if($k - t >= t - s - 1$) break
10: **end for**
11: $b_t = k$
12: Output: b_s, b_t

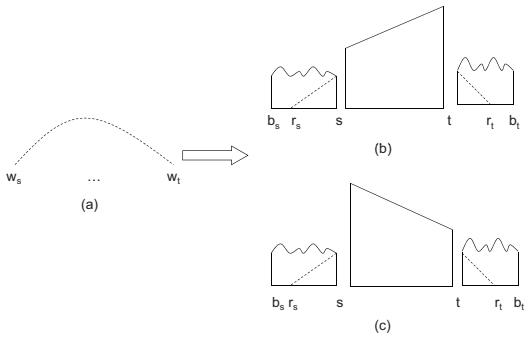


Figure 9: History-outside

Under these two assumptions, Algorithm 1 shows the procedure for searching for history-outside boundaries, where b_s is the boundary for for the descendants on the left side of w_s , b_t is the boundary for searching the descendants on the right side of w_t, and *isPunct* is the function that checks if the word is a punctuation mark. b_s should be in the same sub-sentence with s and $|s - b_s|$ should be less than $|t - s|$. b_t should be in the same sub-sentence with t and $|b_t - t|$ should be less than $|t - s|$.

Next we try to find the subtree structures. First, we collect the part-of-speech (POS) tags of the heads of all the POS tags in training data and remove the tags that occur fewer than 10 times. Then, we determine the directions of the relations by looking up the collected list. For b_s and s, we check if the POS tag of w_s could be the head tag of the POS tag of w_{b_s} by looking up the list. If so, the direction d is $\leftarrow$. Otherwise, we check if the POS tag of w_{b_s} could be the head tag of the POS tag of w_s. If so, d is $\rightarrow$, else d is $\leftarrow$. Finally, we obtain the subtree of w_s from chart item $[b_s, s, d, I]$. Similarly, we obtain the subtree of w_t. Figure 9 shows the history-outside information for w_s and w_t, where the relation between w_{b_s} and w_s and the relation between w_{b_t} and w_t will be determined by the above method. We have subtree $[r_s, s, left, C]$ that roots at w_s on the left side of w_s and subtree $[t, r_t, right, C]$ that roots at w_t on the right side of w_t in Figure 9-(b) and (c).

4.2 Parsing algorithm

Then, we explain how to use these decision history in the parsing algorithm. We use L_{st} to represent the scores of basic features for the left direction and R_{st} for the right case. Then we design history-based features (described in Section 4.3) based on the history-inside and history-outside information, as mentioned above. Finally, we update the scores with the ones of the history-based features by the following equations:

$$L^+_{st} = L_{st} + L^{df}_{st} \quad (1)$$

$$R^+_{st} = R_{st} + R^{df}_{st} \quad (2)$$

where L^+_{st} and R^+_{st} refer to the updated scores, L^{df}_{st} and R^{df}_{st} refer to the scores of the history-based features.

Algorithm 2 Parsing algorithm

1: Initialization: $V[s, s, dir, I/C] = 0.0 \; \forall s, dir$
2: **for** $k = 1$ to n **do**
3: **for** $s = 0$ to $n - k$ **do**
4: $t = s + k$
5: % Create incomplete items
6: L_{st}=$V[s, t, \leftarrow, I]$= $max_{s \leq r < t} VI(r)$;
7: R_{st}=$V[s, t, \rightarrow, I]$= $max_{s \leq r < t} VI(r)$;
8: Calculate L^{df}_{st} and R^{df}_{st};
9: % Update the scores of incomplete chart items
10: $V[s, t, \leftarrow, I]$=L^+_{st}=$L_{st} + L^{df}_{st}$
11: $V[s, t, \rightarrow, I]$=R^+_{st}=$R_{st} + R^{df}_{st}$
12: % Create complete items
13: $V[s, t, \leftarrow, C]$= $max_{s \leq r < t} VC(r)$;
14: $V[s, t, \rightarrow, C]$= $max_{s < r \leq t} VC(r)$;
15: **end for**
16: **end for**

Algorithm 2 is the parsing algorithm with the history-based features, where $V[s, t, dir, I/C]$ refers to the score of chart item $[s, t, dir, I/C]$, $VI(r)$ is a function to search for the optimal sibling and grandchild nodes for the incomplete items (line 6 and 7) (Carreras, 2007) given the

splitting point r and return the score of the structure, and $VC(r)$ is a function to search for the optimal grandchild node for the complete items (line 13 and 14). Compared with the parsing algorithms of Carreras (2007), Algorithm 2 uses history information by adding line 8, 10, and 11.

In Algorithm 2, it first creates chart items with distance 1, then goes on to chart items with distance 2, and so on. In each round, it searches for the structures with the highest scores for incomplete items shown at line 6 and 7 of Algorithm 2. Then we update the scores with the history-based features by Equation 1 and Equation 2 at line 10 and 11 of Algorithm 2. However, note that we can not guarantee to find the candidate with the highest score with Algorithm 2 because new features violate the assumptions of dynamic programming.

4.3 History-based features

In this section, we design features that capture the history information in the recorded decisions.

For a dependency between two words, say s and t, there are four subtrees that root at s or t. We design the features by combining s, t with each child of s and t in the subtrees. The feature templates are shown as follows: (In the following, c means one of the children of s and t, and the nodes in the templates are expanded to their lexical form and POS tags to obtain actual features.):

C+Dir this feature template is a 2-tuple consisting of (1) a c node and (2) the direction of the dependency.

C+Dir+S/C+Dir+T this feature template is a 3-tuple consisting of (1) a c node, (2) the direction of the dependency, and (3) a s or t node.

C+Dir+S+T this feature template is a 4-tuple consisting of (1) a c node, (2) the direction of the dependency, (3) a s node, and (4) a t node.

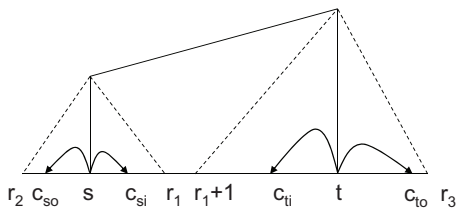


Figure 10: Structure of decision history

We use SHI to represent the subtree of s in the history-inside, THI to represent the one of t in the history-inside, SHO to represent the one of s in the history-outside, and THO to represent the one of t in the history-outside. Based on the subtree types, the features are divided into four sets: F_{SHI}, F_{THI}, F_{SHO}, and F_{THO} refer to the features related to the children that are in subtrees SHI, THI, SHO, and THO respectively.

Figure 10 shows the structure of decision history of a left dependency (between s and t) relation. For the right case, the structure is similar. In the figure, SHI is chart item $[s, r_1, \rightarrow, C]$, THI is chart item $[r_1 + 1, t, \leftarrow, C]$, SHO is chart item $[r_2, s, \leftarrow, C]$, and THO is chart item $[t, r_3, \rightarrow, C]$. We use c_{si}, c_{ti}, c_{so}, and c_{to} to represent a child of s/t in subtrees SHI, THI, SHO, and THO respectively. The lexical form features of F_{SHI} and F_{SHO} are listed as examples in Table 1, where "L" refers to the left direction. We can also expand the nodes in the templates to the POS tags. Compared with the algorithm of Carreras (2007) that only considers the furthest children of s and t, Algorithm 2 considers all the children.

Table 1: Lexical form features of F_{SHI} and F_{SHO}

template	F_{SHI}	F_{SHO}
C+DIR	word-c_{si}+L	word-c_{so}+L
C+DIR+S	word-c_{si}+L+word-s	word-c_{so}+L+word-s
C+DIR+T	word-c_{si}+L+word-t	word-c_{so}+L+word-t
C+DIR +S+T	word-c_{si}+L +word-s+word-t	word-c_{so}+L +word-s+word-t

4.4 Policy of using history

In practice, we define several policies to use the history information for different word pairs as follows:

- All: Use the history-based features for all the word pairs without any restriction.
- Sub-sentences: use the history-based features only for the relation of two words from sub-sentences. Here, we use punctuation marks to split sentences into sub-sentences.
- Distance: use the history-based features for the relation of two words within a predefined distance. We set the thresholds to 3, 5, and 10.

5 Experimental results

In order to evaluate the effectiveness of the history-based features, we conducted experiments on Chinese and English data.

For English, we used the Penn Treebank (Marcus et al., 1993) in our experiments and the tool "Penn2Malt"[3] to convert the data into dependency structures using a standard set of head rules (Yamada and Matsumoto, 2003a). To match previous work (McDonald and Pereira, 2006; Koo et al., 2008), we split the data into a training set (sections 2-21), a development set (Section 22), and a test set (section 23). Following the work of Koo et al. (2008), we used the MXPOST (Ratnaparkhi, 1996) tagger trained on training data to provide part-of-speech tags for the development and the test set, and we used 10-way jackknifing to generate tags for the training set.

For Chinese, we used the Chinese Treebank (CTB) version 4.0[4] in the experiments. We also used the "Penn2Malt" tool to convert the data and created a data split: files 1-270 and files 400-931 for training, files 271-300 for testing, and files 301-325 for development. We used gold standard segmentation and part-of-speech tags in the CTB. The data partition and part-of-speech settings were chosen to match previous work (Chen et al., 2008; Yu et al., 2008).

We measured the parser quality by the unlabeled attachment score (UAS), i.e., the percentage of tokens with the correct HEAD [5]. And we also evaluated on complete dependency analysis.

In our experiments, we implemented our systems on the MSTParser[6] and extended with the parent-child-grandchild structures (McDonald and Pereira, 2006; Carreras, 2007). For the baseline systems, we used the first- and second-order (parent-sibling) features that were used in McDonald and Pereira (2006) and other second-order features (parent-child-grandchild) that were used in Carreras (2007). In the following sections, we call the second-order baseline systems Baseline and our new systems OURS.

5.1 Results with different feature settings

In this section, we test our systems with different settings on the development data.

Table 2: Results with different policies

	Chinese	English
Baseline	89.04	92.43
D_1	88.73	92.27
D_3	88.90	92.36
D_5	89.10	92.59
D_{10}	89.32	92.57
D_{sub}	89.57	92.63

Table 2 shows the parsing results when we used different policies defined in Section 4.4 with all the types of features, where D_{sub} refers to applying the policy: sub-sentence, D_1 refers to applying the policy: all, and $D_{3|5|10}$ refers to applying the policy: distance with the predefined distance 3, 5, or 10. The results indicated that the accuracies of our systems decreased if we used the history information for short distance words. The system with D_{sub} performed the best.

Table 3: Results with different types of Features

	Chinese	English
Baseline	89.04	92.43
+F_{SHI}	89.14	92.53
+F_{THI}	89.33	92.35
+F_{SHO}	89.25	92.47
+F_{THO}	88.99	92.54

Then we investigated the effect of different types of the history-based features. Table 3 shows the results with policy D_{sub}. From the table, we found that F_{THI} provided the largest improvement for Chinese and F_{THO} performed the best for English.

In what follows, we used D_{sub} as the policy for all the languages, the features $F_{SHI} + F_{THI} + F_{SHO}$ for Chinese, and the features $F_{SHI} + F_{SHO} + F_{THO}$ for English.

5.2 Main results

The main results are shown in the upper parts of Tables 4 and 5, where the improvements by OURS over the Baselines are shown in parentheses. The results show that OURS provided better performance over the Baselines by 1.02 points for Chi-

[3] http://w3.msi.vxu.se/~nivre/research/Penn2Malt.html

[4] http://www.cis.upenn.edu/~chinese/.

[5] As in previous work, English evaluation ignores any token whose gold-standard POS tag is one of {'' `` : , .} and Chinese evaluation ignores any token whose tag is "PU".

[6] http://mstparser.sourceforge.net

Table 4: Results for Chinese

	UAS	Complete
Baseline	88.41	48.85
OURS	89.43(+1.02)	50.86
OURS+STACK	89.53	49.42
Zhao2009	87.0	–
Yu2008	87.26	–
STACK	88.95	49.42
Chen2009	89.91	48.56

Table 5: Results for English

	UAS	Complete
Baseline	91.92	44.28
OURS	92.21 (+0.29)	45.24
Y&M2003	90.3	38.4
CO2006	90.8	37.6
Z&C2008	92.1	45.4
STACK	92.53	47.06
KOO2008	93.16	–
Chen2009	93.16	47.15
Suzuki2009	93.79	–

nese and 0.29 points for English. The improvements of (OURS) were significant in McNemar's Test with $p < 10^{-4}$ for Chinese and $p < 10^{-3}$ for English.

5.3 Comparative results

Table 4 shows the comparative results for Chinese, where Zhao2009 refers to the result of (Zhao et al., 2009), Yu2008 refers to the result of Yu et al. (2008), Chen2009 refers to the result of Chen et al. (2009) that is the best reported result on this data, and STACK refers to our implementation of the combination parser of Nivre and McDonald (2008) using our baseline system and the MALTParser[7]. The results indicated that OURS performed better than Zhao2009, Yu2008, and STACK, but worse than Chen2009 that used large-scale unlabeled data (Chen et al., 2009). We also implemented the combination system of OURS and the MALTParser, referred as OURS+STACK in Table 4. The new system achieved further improvement. In future work, we can combine our approach with the parser of Chen et al. (2009).

Table 5 shows the comparative results for English, where Y&M2003 refers to the parser of Yamada and Matsumoto (2003b), CO2006 refers to the parser of Corston-Oliver et al. (2006), Z&C 2008 refers to the combination system of Zhang and Clark (2008), STACK refers to our implementation of the combination parser of Nivre and McDonald (2008), KOO2008 refers to the parser of Koo et al. (2008), Chen2009 refers to the parser of Chen et al. (2009), and Suzuki2009 refers to the parser of Suzuki et al. (2009) that is the best reported result for this data. The results shows that OURS outperformed the first two systems that were based on single models. Z&C 2008 and STACK were the combination systems of graph-based and transition-based models. OURS performed better than Z&C 2008, but worse than STACK. The last three systems that used large-scale unlabeled data performed better than OURS.

[7]http://www.maltparser.org/

6 Related work

There are several studies that tried to overcome the limited feature scope of graph-based dependency parsing models .

Nakagawa (2007) proposed a method to deal with the intractable inference problem in a graph-based model by introducing the Gibbs sampling algorithm. Compared with their approach, our approach is much simpler yet effective. Hall (2007) used a re-ranking scheme to provide global features while we simply augment the features of an existing parser.

Nivre and McDonald (2008) and Zhang and Clark (2008) proposed stacking methods to combine graph-based parsers with transition-based parsers. One parser uses dependency predictions made by another parser. Our results show that our approach can be used in the stacking frameworks to achieve higher accuracy.

7 Conclusions

This paper proposes an approach for improving graph-based dependency parsing by using the decision history. For the graph-based model, we design a set of features over short dependencies computed in the earlier stages to improve the accuracy of long dependencies in the later stages. The results demonstrate that our proposed approach outperforms baseline systems by 1.02 points for Chinese and 0.29 points for English.

References

Buchholz, S., E. Marsi, A. Dubey, and Y. Krymolowski. 2006. CoNLL-X shared task on multilingual dependency parsing. *Proceedings of CoNLL-X*.

Carreras, X. 2007. Experiments with a higher-order projective dependency parser. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007*, pages 957–961.

Chen, WL., D. Kawahara, K. Uchimoto, YJ. Zhang, and H. Isahara. 2008. Dependency parsing with short dependency relations in unlabeled data. In *Proceedings of IJCNLP 2008*.

Chen, WL., J. Kazama, K. Uchimoto, and K. Torisawa. 2009. Improving dependency parsing with subtrees from auto-parsed data. In *Proceedings of EMNLP 2009*, pages 570–579, Singapore, August.

Corston-Oliver, S., A. Aue, Kevin. Duh, and Eric Ringger. 2006. Multilingual dependency parsing using bayes point machines. In *HLT-NAACL2006*.

Culotta, A. and J. Sorensen. 2004. Dependency tree kernels for relation extraction. In *Proceedings of ACL 2004*, pages 423–429.

Eisner, J. 1996. Three new probabilistic models for dependency parsing: An exploration. In *Proc. of COLING 1996*, pages 340–345.

Hall, Keith. 2007. K-best spanning tree parsing. In *Proc. of ACL 2007*, pages 392–399, Prague, Czech Republic, June. Association for Computational Linguistics.

Koo, T., X. Carreras, and M. Collins. 2008. Simple semi-supervised dependency parsing. In *Proceedings of ACL-08: HLT*, Columbus, Ohio, June.

Li, Charles N. and Sandra A. Thompson. 1997. *Mandarin Chinese - A Functional Reference Grammar*. University of California Press.

Marcus, M., B. Santorini, and M. Marcinkiewicz. 1993. Building a large annotated corpus of English: the Penn Treebank. *Computational Linguisticss*, 19(2):313–330.

McDonald, R. and J. Nivre. 2007. Characterizing the errors of data-driven dependency parsing models. In *Proceedings of EMNLP-CoNLL*, pages 122–131.

McDonald, R. and F. Pereira. 2006. Online learning of approximate dependency parsing algorithms. In *Proc. of EACL2006*.

McDonald, Ryan. 2006. *Discriminative Training and Spanning Tree Algorithms for Dependency Parsing*. Ph.D. thesis, University of Pennsylvania.

Nakagawa, Tetsuji. 2007. Multilingual dependency parsing using global features. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007*, pages 952–956.

Nakazawa, T., K. Yu, D. Kawahara, and S. Kurohashi. 2006. Example-based machine translation based on deeper NLP. In *Proceedings of IWSLT 2006*, pages 64–70, Kyoto, Japan.

Nivre, J. and R. McDonald. 2008. Integrating graph-based and transition-based dependency parsers. In *Proceedings of ACL-08: HLT*, Columbus, Ohio, June.

Nivre, J., J. Hall, and J. Nilsson. 2004. Memory-based dependency parsing. In *Proc. of CoNLL 2004*, pages 49–56.

Nivre, J., J. Hall, S. Kübler, R. McDonald, J. Nilsson, S. Riedel, and D. Yuret. 2007. The CoNLL 2007 shared task on dependency parsing. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007*, pages 915–932.

Ratnaparkhi, A. 1996. A maximum entropy model for part-of-speech tagging. In *Proceedings of EMNLP*, pages 133–142.

Suzuki, Jun, Hideki Isozaki, Xavier Carreras, and Michael Collins. 2009. An empirical study of semi-supervised structured conditional models for dependency parsing. In *Proc. of EMNLP 2009*, pages 551–560, Singapore, August. Association for Computational Linguistics.

Yamada, H. and Y. Matsumoto. 2003a. Statistical dependency analysis with support vector machines. In *Proceedings of IWPT2003*, pages 195–206.

Yamada, H. and Y. Matsumoto. 2003b. Statistical dependency analysis with support vector machines. In *Proceedings of IWPT2003*, pages 195–206.

Yu, K., D. Kawahara, and S. Kurohashi. 2008. Chinese dependency parsing with large scale automatically constructed case structures. In *Proceedings of Coling 2008*, pages 1049–1056, Manchester, UK, August.

Zhang, Y. and S. Clark. 2008. A tale of two parsers: Investigating and combining graph-based and transition-based dependency parsing. In *Proceedings of EMNLP 2008*, pages 562–571, Honolulu, Hawaii, October.

Zhao, Hai, Yan Song, Chunyu Kit, and Guodong Zhou. 2009. Cross language dependency parsing using a bilingual lexicon. In *Proceedings of ACL-IJCNLP2009*, pages 55–63, Suntec, Singapore, August. Association for Computational Linguistics.

A comparison of unsupervised methods for Part-of-Speech Tagging in Chinese

Alex Cheng
Microsoft Corporation
alcheng@microsoft.com

Fei Xia
Univ. of Washington
fxia@uw.edu

Jianfeng Gao
Microsoft Research
jfgao@microsoft.com

Abstract

We conduct a series of Part-of-Speech (POS) Tagging experiments using Expectation Maximization (EM), Variational Bayes (VB) and Gibbs Sampling (GS) against the Chinese Penn Treebank. We want to first establish a baseline for unsupervised POS tagging in Chinese, which will facilitate future research in this area. Secondly, by comparing and analyzing the results between Chinese and English, we highlight some of the strengths and weaknesses of each of the algorithms in POS tagging task and attempt to explain the differences based on some preliminary linguistics analysis. Comparing to English, we find that all algorithms perform rather poorly in Chinese in 1-to-1 accuracy result but are more competitive in many-to-1 accuracy. We attribute one possible explanation of this to the algorithms' inability to correctly produce tags that match the desired tag count distribution.

1 Introduction

Recently, there has been much work on unsupervised POS tagging using Hidden Markov Models (Johnson, 2007; Goldwater & Griffiths, 2007). Three common approaches are Expectation Maximization (EM), Variational Bayes (VB) and Gibbs Sampling (GS). EM was first used in POS tagging in (Merialdo, 1994) which showed that except in conditions where there are no labeled training data at all, EM performs very poorly. Gao and Johnson (2008) compared EM, VB and GS in English against the Penn Treebank Wall Street Journal (WSJ) text. Their experiments on English showed that GS outperforms EM and VB in almost all cases. Other notable studies in the unsupervised and semi-supervised POS domain include the use of prototype examples (Haghighi & Klien, 2006), dictionary constraints to guide the algorithms (Elworthy 1994; Banko & Moore 2004) and Bayseian LDA-based model (Toutanova and Johnson, 2007).

To our knowledge, little work has been done on unsupervised POS tagging in Chinese against the Chinese Penn Treebank (CTB). The work in Chinese POS tagging has been predominately in the supervised fashion (Huang et al. 2009; Chang & Chen, 1993; Ng & Low, 2004) and achieve accuracy of 92.25% using a traditional ngram HMM tagger. For English, a supervised trigram tagger achieves an accuracy of 96.7% against the Penn Treebank (Thorsten, 2000).

In this study, we analyze and compare the performance of three classes of unsupervised learning algorithms on Chinese and report the experimental results on the CTB. We establish a baseline for unsupervised POS tagging in Chinese. We then compare and analyze the results between Chinese and English, we explore some of the strengths and weaknesses of each of the algorithms in POS tagging task and attempt to explain the differences based on some preliminary linguistics analysis.

2 Models

In this section, we provide a brief overview of the three unsupervised learning methods for POS tagging as described in (Gao & Johnson, 2008), which all uses a traditional bigram Hidden Markov Model (HMM). HMM is a well-

Coling 2008: Poster Volume, pages 135–143,
Beijing, August 2010

known statistical model, used for sequential modeling. To put it formally, let $T = \{t_1, \ldots, t_i, \ldots, t_j, \ldots, t_{|T|}\}$ be the set of possible states and $W = \{w_1, \ldots, w_k, \ldots, w_{|W|}\}$ be the set of possible observations. In the case for POS tagging using a bigram model, the set T corresponds to the set of POS tags and the set W corresponds to the set of words in the language.

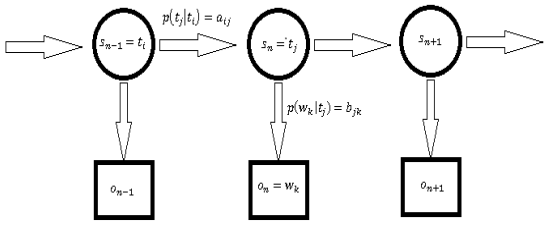


Figure 1: Graphical model of an HMM for a bigram POS tagger. The top row represents a sequence of hidden states where each is conditionally dependent only on the previous state and the bottom row represents a sequence of observations where each is conditionally dependent only on the current state.

An HMM models a sequence of discrete observations $o_{1:N} = \{o_1, \ldots, o_N\}$ where $o_n = w_k$ that are produced by a sequence of hidden states $\boldsymbol{s}_{1:N} = \{s_1, \ldots, s_N\}$ where $s_n = t_i$. The sequence of states is produced by a first order Markov process such that the current state s_n depends only on its previous state s_{n-1}; correspondingly each of the observations o_n depends only on the state s_n:

$$p(s_n|\boldsymbol{s_{1:n-1}}) \cong p(s_n|s_{n-1}) \\ p(o_n|\boldsymbol{s_{1:n}}\, \boldsymbol{o}_{1:n-1}) \cong p(o_n|s_n) \qquad (1)$$

where $a_{ij} = p(s_n|s_{n-1})$ is the probability of transition to state $s_n = t_j$ from $s_{n-1} = t_i$ and $b_{jk} = p(o_n|s_n)$ is the probability of observation $o_n = w_k$ produced by $s_n = t_j$. The parameter θ for the HMM is defined by the transition probability distribution $A \equiv (a_{ij})$, emission (observation) probability distribution $B \equiv (b_{jk})$ and the initial probability $\pi \equiv \prod p(s_0 = t_i)$. Direct calculation of the likelihood $p(s, o|\theta)$ is computationally inefficient, and we can use dynamic programming techniques to speed up the calculation by calculating the forward probability:

$$\alpha_i(n) \equiv p(o_{1:n}, s_n = t_i|\theta) \qquad (2)$$

and backward probability

$$\beta_i(n) \equiv p(o_{n:N}|s_n = t_i, \theta). \qquad (3)$$

See (Mannings & Schutze, 1999) for details on the calculation.

2.1 Expectation Maximization (EM)

EM is a general class of algorithms for finding the maximum likelihood estimator of parameters in probabilistic models. It is an iterative algorithm where we alternate between calculating the expectation of the log likelihood of the model given the parameters:

$$Q(\theta|\theta^t) = \mathbb{E}_{p(s|o,\theta)} \ln p(s, o|\theta) \qquad (4)$$

and then finding the parameters that maximizes the expected log likelihood. Using Lagrange multipliers with constraint that each parameter is a probability distribution, we have these update steps for the well-known forward-backward Algorithm for EM HMM:

$$\pi^{t+1} : \pi_i = \alpha_i(1)\beta_i(1)$$

$$A^{t+1} : a_{ij} = \frac{\sum_{n=2} \alpha_i(n) a_{ij} b_{jo_n} \beta_j(n+1)}{\sum_{i=1}^{T} \alpha_i(n) \beta_{\ i}(n)} \qquad (5)$$

$$B^{t+1} : b_{jk} = \sum_{n=1}^{N} \frac{\alpha_i(n) a_{ij} b_{jo_n} \beta_j(n+1) \delta(o_n, k)}{\sum_{i=1}^{T} \alpha_i(n) \beta_i(n)}$$

where $\delta(o_n, k) = \begin{cases} 1, & o_n = w_k \\ 0, & o.w. \end{cases}$.

2.2 Variational Bayes (VB)

One of the drawbacks of EM is that the resulting distribution is very uniform; that is, EM applies roughly the same number of observations for each state. Instead of using only the best model for decoding, the Bayesian approach uses and considers all the models; that is, the model is treated as a hidden variable. This is done by assigning a probability distribution over the model parameters as a prior distribution, $p(\theta)$.

In HMM, we calculate the probability of the observation by considering all models and integrating over the distribution over the priors:

$$p(\boldsymbol{o}_{1:N}) = \int p(o_{1:N}|\theta)p(\theta)\,d\theta \quad (6)$$

where $p(\theta) = p(\pi)p(A)p(B)$.

As with the standard in the literature, we use Dirichlet Prior as it allows us to model the tag distribution more closely and because they are in the same conjugate exponential family as the log likelihood. The Dirichlet distribution is parameterized by a vector of real values $\boldsymbol{\alpha}$ (hyper-parameters). There are two ways that we can view the vector $\boldsymbol{\alpha}$. First, the parameter controls the sharpness of distribution for each of the components. This is in contrast to the EM model where we essentially have a uniform prior. Thus, we can view $\boldsymbol{\alpha}$ as our prior beliefs on the shape of the distribution and we can make our choices based on our linguistics knowledge. Second, we can view the role of $\boldsymbol{\alpha}$ in terms of predictive distribution based on the statistics from observed counts. For HMM, we can set a separate prior for each state-state transition and word-state emission distribution, effectively giving us control over the distribution of each entry in the transition matrix. However, to simplify the model and without the need to fine tune each parameters, we use two fixed hyper-parameters: all of the state-state probability will have the hyper-parameter α_{TT} and all of the word-state probability will have hyper-parameter α_{WT}.

To begin our estimation and maximization procedure, we create $q(\theta, s_{1:N}) \approx q(\theta)q(s_{1:N})$ as an approximation of the posterior of the log likelihood:

$$\ln p(\boldsymbol{o}_{1:N}) \geq \int \sum_{s} q(\theta, s_{1:N}) \ln \frac{p(o_{1:N}, s_{1:N}|\theta)p(\theta)}{q(\theta, s_{1:N})} d\theta \quad (7)$$

By taking the functional derivative with respect to $q(\theta)$ to find the distribution that maximizes the log likelihood, and following the derivation from (Beal, 2003), we arrive at the following EM-like procedure:

$$\begin{aligned} &\mathrm{E}_{q(s)}[\delta(s_1, t_i)] \\ &= \alpha_i(1)\beta_i(1)\mathrm{E}_{q(s)}\big[\delta(s_{n-1}, t_i)\delta\big(s_n, t_j\big)\big] \\ &= \frac{\alpha_i(n)a_{ij}b_{jo_n}\beta_j(n+1)}{\sum_{i=1}^{T}\alpha_i(n)\beta_i(n)} \end{aligned} \quad (8)$$

$$\begin{aligned} &\mathrm{E}_{q(s)}[\delta(s_n, t_i)\delta(o_n, w_k)] \\ &= \frac{\alpha_i(n)a_{ij}b_{jo_n}\beta_j(n+1)\delta(o_n, w_k)}{\sum_{i=1}^{T}\alpha_i(n)\beta_i(n)} \end{aligned}$$

This is the Expectation step where α and β is the forward and backward probabilities and $\delta(o_n, w_k)$ is the indicator function as in EM.

The Maximization step is as follows:

$$\begin{aligned} &\pi^{t+1} : \pi_i \\ &= \frac{\exp(\psi(\alpha_{TT} + \mathrm{E}_{q(s)}[\delta(s_1, t_i)]))}{\exp(\psi(\sum_{i=1}^{T}\alpha_{TT} + \mathrm{E}_{q(s)}[\delta(s_1, t_i)]))} \\ &A^{t+1} : a_{ij} \\ &= \frac{\exp(\psi(\alpha_{TT} + \sum_{n=2}^{N}\mathrm{E}_{q(s)}[*]))}{\exp(\psi(\sum_{i=1}^{T}\alpha_{TT} + \sum_{n=2}^{N}\mathrm{E}_{q(s)}[*]))} \\ &B^{t+1} : b_{jk} \\ &= \frac{\exp(\psi(\alpha_{WT} + \sum_{n=1}^{N}\mathrm{E}_{q(s)}[**]))}{\exp(\psi(\sum_{i=1}^{T}\alpha_{WT} + \sum_{n=1}^{N}\mathrm{E}_{q(s)}[**]))} \end{aligned} \quad (9)$$

where $\mathrm{E}_{q(s)}[*] = E_{q(s)}[\delta(s_{n-1}, t_i)\delta(s_n, t_j)]$, $\mathrm{E}_{q(s)}[**] = E_{q(s)}[\delta(s_n, t_j)\delta(o_n, w_k)]$ and ψ is the digamma function.

2.3 Gibbs Sampling (GS)

Gibbs sampling (Geman & Geman, 1984) is a widely used MCMC algorithm designed especially for cases where we can sample from the conditional probability easily. It is a straightforward application of the Metropolis Hasting algorithm where we sample a variable t_k while keeping $t_{\setminus k}$ constant where $t_{\setminus k} = \{t_1, \ldots, t_{k-1}, t_{k+1}, \ldots, t_T\}$. We set the proposal distribution to

$$q_k(t^*|t^n) = p\big(t_k\big|t_{\setminus k}\big). \quad (10)$$

So the sampling procedure is the following: initialize the components of $\boldsymbol{t} = \{t_1, , \ldots, t_T\}$. Then sample t_1 from $p(t_1|t_2, \ldots, t_T)$, t_2 from $p(t_2|t_1, t_3, \ldots, t_T)$, and so on for each component of t. For POS tagging, the main idea is that we sample the tag t based on the $p(t|t')$ and $p(w|t)$ distribution.

The main idea for using GS for POS tagging is that in each iteration, we sample the tag t based on the $p(t|t')$ and $p(w|t)$ distribution.

Then from the samples, we count the number for each state-state and word-state pairs and update the probabilities accordingly. How we sample the data depends on whether we are using word based or sentence based sampling (the Expectation Step). Whereas how we update the probabilities depend on whether we are using a collapsed or explicit Gibbs sampler (the Maximization Step).

Word Based vs. Sentence Based: Word-based and sentence-based approaches to GS determine how we sample the each tag t at position n in the data set. For the word-based approach, instead of going through sentence by sentence (as in EM and VB procedures), we pick a word position in the corpus at random (without repetition) and sample a new tag t_i at position n using the probability:

$$p(s_n = t_i) = p(t_i|s_{n-1})p(w_k = o_n|t_i)p(s_{n+1}|t_i) \quad (11)$$

Notice that since we are selecting each position at random, the tag s_{n-1} at position n-1 and s_{n+1} at position n+1 are our samples at the previous iteration or an already updated samples at the current iteration.

The sentence-based approach use the forward and backward probability to sample the tag based on the sentence (Besag, 2004). Specifically, we use the backward probability $\beta_i(n) \equiv p(o_{1:n}|s_n = t_i, \theta)$ to sample the sentence from start $(n = 1)$ to finish $(n = N)$. We sample a new tag t_i at position n using the probability:

$$p(s_n = t_i|o_{1:n}, s_{1:n-1}, \theta) = p(t_i|s_{n-1})p(w_k = o_n|t_i)\beta_i(n) \quad (12)$$

where the transition and emission probability distribution are from the current model parameters. Again s_{n-1} is our "guess" at the previous sampling step of the tag of s_{n-1}.

Explicit vs. Collapsed Based: We use the tags estimated at the previous step to maximize the parameters. Our choice of using Dirichlet distributions over the parameters $P(A)$ and $P(B)$ give us some nice mathematical properties. We show that $p(A|\boldsymbol{o}_{1:N})$ and $p(B|\boldsymbol{o}_{1:N})$ also calculate to be Dirichlet distributions. Following (MacKay & Peto, 1994), the posterior probability of A can be derived as follows:

$$\begin{aligned} p(A|\boldsymbol{o}_{1:N}) &= \frac{p(\boldsymbol{o}_{1:N}|A)p(A)}{p(\boldsymbol{o}_{1:N})} \\ &= \frac{1}{p(\boldsymbol{o}_{1:N})}\prod_{n=1}^{N} a_{s_n,s_{n-1}} \prod_i^T \prod_j^T \frac{a_{ij}^{\alpha-1}\Gamma(\alpha_{TT})}{\Gamma(|T|\alpha_{TT})} \\ &= \prod_{i=1}^{T}\prod_{j=1}^{T} \frac{a_{ij}^{C(t_i,t_j)+\alpha_{TT}-1}\Gamma\left(\alpha_{TT} + C(t_i,t_j)\right)}{\Gamma(|T|\alpha_{TT} + \sum c(t_i,t_j)} \\ &= \prod_j Dir(a_{ij}|\, c(t_i,t_j) + \alpha_{TT}) \end{aligned} \quad (13)$$

where $c(t_i, t_j)$ is the number of times t_i is followed by t_j in the sample from the previous iteration.

Similarly, we can define $p(B|o_{1:N})$ using the count $c(w_k, t_j)$ to show that:

$$p(B|\boldsymbol{o}_{1:N}) = \prod_k Dir(b_{jk}|\, c(w_k,t_j) + \alpha_{WT}) \quad (14)$$

For the collapsed Gibbs sampler, we want to integrate over all possible model parameters A to maximize the new transition probabilities using Maximum a posteriori (MAP) estimator:

$$\begin{aligned} p(t_j|t_i, o_{1:N}) &= \int p(t_j|t_i, A, o_{1:N})p(A|\boldsymbol{o}_{1:N})dA \\ &= \int a_{ij}\prod_j Dir(a_{ij}|\, c(t_i,t_j) + \alpha_{TT})\, dA \\ &= \frac{c(t_i,t_j) + \alpha_{TT}}{\sum c(t_i,t_j) + |T|\alpha_{TT}} \end{aligned} \quad (15)$$

The last equality uses the following result:

$$\int \boldsymbol{\pi} Dir(\boldsymbol{\pi}|u)d\boldsymbol{\pi} = \frac{u}{|\pi|u} \quad (16)$$

We can derive a similar result for $p(w_k|t_i, o_{1:N})$. Then we can use the sample count to update the new parameter values.

An explicit sampler samples the HMM parameters θ in addition to the states. Specifically, in the Bayesian model, we will need to sample from the Dirichlet distribution for the parameters

$$p(A|o_{1:N}) = \prod_j Dir(a_{ij}|\, c(t_i, t_j) + \alpha_{TT}) \quad (17)$$

$$p(B|\boldsymbol{o}_{1:N}) = \prod_k Dir(b_{jk}|\, c(t_j, w_k) + \alpha_{WT})$$

derived above. An n-dimensional Dirichlet distribution variable can be generated from gamma variate (Wolfram Mathematica, 2009):

$$u_{ij} \sim f(x, c(t_i, t_j) + \alpha_{TT}) = \frac{x^{c(t_i,t_j)+\alpha_{TT}-1} e^{-x}}{\Gamma(c(t_i, t_j) + \alpha_{TT})} \quad (18)$$

we can update the transition probability by generating the gamma variate for the Dirichlet distribution:

$$a_{ij} \leftarrow \frac{u_{ij}}{\sum_j u_{ij}}. \quad (19)$$

Similarly, we sample the emission probability using the count for word-tag with $c(w_k, t_j) + \alpha_{WT}$ as the hyper-parameter.

3 Experiment Setup

Our experiment setup is similar to the ones used in (Gao & Johnson, 2007). They are summarized in Table 1:

Parameters	**Values**
Data Size	24k, 120k, 500k
Algorithm	EM, VB, GS(c,w), GS(c,s), GS(e,s), GS(e,w)
# of states	Chinese: 33 English: 50
α_{TT}	0.0001, 0.1, 0.5, 1
α_{WT}	0.0001, 0.1, 0.5, 1

Table 1: The list of experiments conducted. For the hyper-parameters $(\alpha_{TT}, \alpha_{WT})$, we try the combination of the adjacent pairs – (0.0001,0.0001), (0.1,0.0001), (0.0001,0.1), (0.1, 0.1), (0.1, 0.5), etc.

3.1 Data

For our experiments, we use the data set Chinese Penn Treebank (CTB) v5.0. The Chinese Treebank project began at the University of Pennsylvania in 1998 and the team created a set of annotation guidelines for word segmentation, POS tagging and bracketing (Xia, 2000; Xue et al., 2002; Xue et al., 2005). The version used in this paper is the Chinese Treebank 5.0 which consists of over 500k words and over 800k Chinese characters. The text comes from various sources including newswire, magazine articles, website news, transcripts from various broadcast news program.

Chinese POS tagging faces additional challenges because it has very little, if any, inflectional morphology. Words are not inflected with number, gender, case, or tense. For example, a word such as 毁灭 in Chinese corresponds to d*estroy /destroys /destroyed/destruction* in English. This fuels the discussion in Chinese NLP communities on whether the POS tags should be based on meaning or on syntactic distribution (Xia, 2000). If only the meaning is used, 毁灭 should be a verb all the time. If syntactic distribution is used, the word is a verb or a noun depending on the context. For the CTB, syntactic distribution is used, which complies with the principles of contemporary linguistics theories.

Following the experiment done for English in (Gao & Johnson, 2008), we split the data into three sizes: 24k words, 120k words and all words (500k), and used the same data set for training and testing. The idea is to track the effectiveness of an algorithm across different corpus sizes. Instead of using two different tag set sizes (17 and 50) as it is done for English POS tagging, we opt to keep the original 33 tag set for Chinese without further modification. In addition to reporting the results for English from (Gao & Johnson, 2008), we run additional experiments on English using only 500k words for comparison.

3.2 Decoding

For decoding, we use max marginal likelihood estimator (as opposed to using Viterbi algorithm) to assign a tag for each word in the result tag. (Gao & Johnson, 2008) finds that max marginal decoder performs as well as Viterbi algorithm and runs significantly faster as we can reuse the forward and backwards probabilities already calculated during the estimation and update step.

3.3 Hyperparameters

For the Bayesian approaches (VB and GS), we have a choice of hyperparameters. We choose uniform hyperparameters α_{TT} and α_{WT} instead

of choosing a specific hyper-parameter for each of the tag-tag and word-tag distribution. The values for the hyper-parameters are chosen such that we can see more clearly the interactions between the two values. For GS, we use the notation GS(c,s) to denote collapsed sentence-based approach, GS(e,s) for explicit sentence based, GS(c,w) for collapsed word-based and GS(e,w) for explicit word based.

3.4 Evaluation Metrics

We use POS tagging accuracy as our primary evaluation method. There are two commonly used methods to map the state sequences from the system output to POS tags. In both methods, we first create a matrix where each row corresponds to a hidden state, each column corresponds to a POS tag, and each cell (i,j) represents the number of times a word position in the test data comes from the hidden state t_i according to the system output and the position has tag t_j according to the gold standard. In greedy 1-to-1 mapping, we find the largest value in the table – suppose the value is for the cell (i,j). We map state i to tag j, and remove both row i and column j from the table. We repeat the process until all the rows have been removed. Greedy many-to-1 allow multiple hidden states to map to a single POS tag. That is, when the highest value in the table is found, only the corresponding row is removed. In other words, we simply map each hidden state to the POS tag that the hidden state co-occurs with the most.

4 Results and Analysis

We compare and analyze the results between the different algorithms and between Chinese and English using Greedy 1-to-1 accuracy, Greedy many-to-1 accuracy.

4.1 Greedy 1-to-1 accuracy

When measure using 1-to-1 mapping, the best algorithm – Collapsed word based Gibbs Sampling GS(c,w) - achieve 0.358 in Chinese on the full data set but remains close to 0.499 in English for the full dataset. GS(c,w) outperforms other algorithm in almost all categories. But EM posts the highest relative improvement with an increase of 70% when the data size increases from 24k to 500k words. The full result is listed in Table 2.

		Greedy 1-to-1		
		24k	120k	500k
Chinese	**EM**	0.1483	0.1838	0.2406
	VB	0.1925	0.2498	0.3105
	GS(e,w)	0.2167	0.3108	0.3475
	GS(e,s)	0.2262	0.2596	0.3572
	GS(c,s)	0.2351	0.2931	**0.3577**
	GS(c,w)	**0.2932**	**0.3289**	0.3558
Eng	**EM**	0.1862	0.2930	0.3837
	VB	0.2382	0.3468	0.4327
	GS(c,w)	**0.3918**	**0.4276**	**0.4348**

Table 2: Tagging accuracy for Chinese and English with greedy 1-to-1 mapping. The English 24k and 120k results are taken from (Gao & Johnson 2008) with the 50-tag set.

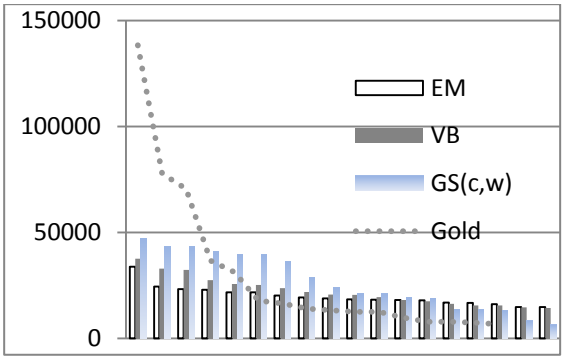


Figure 2: Tag distribution for 1-to-1 greedy mapping in Chinese 500k. Only the top 18 tags are shown. The figure compares the tag distribution between the gold standard for Chinese (33 tags) and the algorithm's results. The gold tags are shown as lines, and each algorithm's result is shown as bar graphs.

As expected, the increase in data size improves the accuracy as EM algorithm optimizes the likelihood better with more data. We ran additional experiments on English using a reduced 500k dataset to match the dataset used for Chinese; EM in this setting achieve an accuracy of 0.384 on average for 50 tags (down from 0.405). So even in the reduced data size setting, EM on English performs better than Chinese although the difference is reduced. We analyze the tag distribution of the 1-to-1 mapping. (Johnson, 2007) finds that EM generally assigns roughly as equal number of words for each state. In Figure 2, we find the same phenomenon for Chinese.

One of the advantages of Bayesian approaches (VB and GS) is that we can assign a prior to attempt to encourage a sparse model distribution. Despite using small values 0.0001 as hyperparameters, we find that the resulting distribution for number of words mapping to a particular state is very different from the gold standard.

4.2 Greedy many-to-1 accuracy

Collapsed Word Based Gibbs Sampler GS(c,w) is the clear winner for both English and Chinese unsupervised POS tagging. Table 3 shows the result of Greedy many-to-1 mapping for Chinese in different data size as well as English with the full data set. In Greedy many-to-1 mapping, GS(c,w) in both Chinese and English achieve 60%+ accuracy. In addition, the size of the dataset does not affect GS(c,w) as much as the other algorithms. In fact, the change from 24k to 500k dataset only increases the relative accuracy by less than 6%.

	Greedy many-to-1			
		24k	120k	500k
Chinese	**EM**	0.4049	0.4564	0.4791
	VB	0.4411	0.5023	0.5390
	GS(e,w)	0.4758	0.4969	0.5499
	GS(e,s)	0.4904	0.5369	0.5658
	GS(c,s)	0.5070	0.5701	0.5757
	GS(c,w)	**0.5874**	**0.6180**	**0.6213**
Eng	**EM**	0.2828	0.44135	0.5872
	VB	0.3595	0.48427	0.6025
	GS(c,w)	**0.5815**	**0.6529**	**0.6644**

Table 3: Many-to-1 accuracy for Chinese and English. The English 24k and 120k results are taken from (Gao & Johnson 2008) with the 50-tag set.

However, despite the relatively high accuracy, when analyzing the result, we notice that there are overwhelmingly many states which maps to a single POS tag (NN). Figure 3 shows the number of states mapping to different POS tags in Chinese over the 500k data size. There are a large number of states mapping to relatively few POS tags. In the most extreme example, for the POS tag NN, GS(e,s) assigns 18 (the most) hidden states, accounts for 44% of the word tokens mapping to NN whereas GS(e,w) assigns 13 states, which is actually the least among all the algorithms and accounts for 31% of the word tokens mapping to NN. Notice that we have only a total of 33 hidden states in our model. This means that over half the states are mapped to NN, which is a rather disappointing result. The actual empirical result for the gold standard in CTB is that only 27% of the word should be mapped to NN. For EM in particular, we see 17 states accounting for 42% of the words tagged as NN.

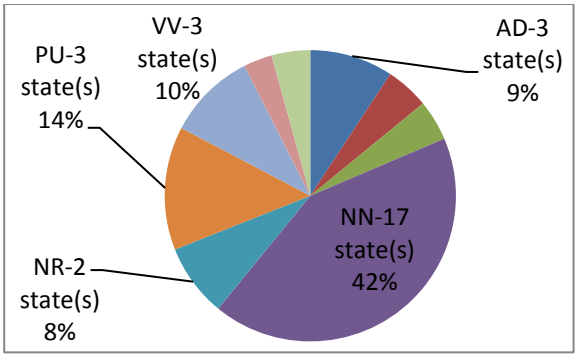


Figure 3: The distribution of POS tags based on the output EM algorithm in Chinese using the 500k dataset. Tag T-N-y% means that there are N hidden states mapped to the specific POS tag T accounting for y% of word tokens tagged with these N states by the EM algorithm.

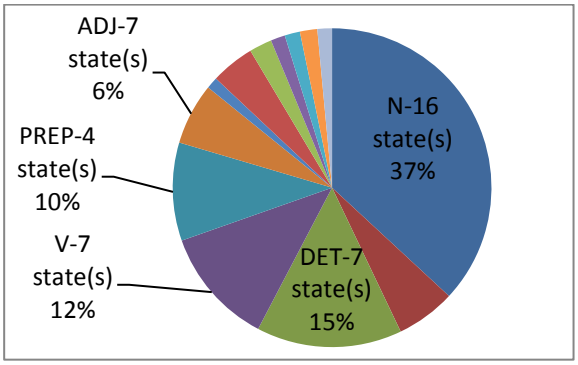


Figure 4: English tag distribution for EM using 500k dataset with 50 states mapping to the 17 pos tag set. Tag T-N-y% means that there are N hidden states mapped to the specific POS tag T accounting for y% of word tokens tagged with these N states.

We also ran additional experiments on the algorithms for English using a reduced data size of 500k to match that of our Chinese experiment to see whether we see the same phenomena. We notice that the tag distribution for English EM is more consistent to the empirical distribution found in the gold standard.

With the English 50 tag set with 500k words, we experiment with mapping the English 50 tag set result to the 17 tag set, we see that in Figure 4, 16 (of 50) states mapped to the N tag, accounting for 37% of the words in the dataset. This is close to the actual empirical distribution for English for 17 tags where N accounts for about 32%.

4.3 Convergence

We analyze how each algorithm converges to its local maxima. Figure 5 shows the change in greedy 1-to-1 accuracy over the 50% of the run.

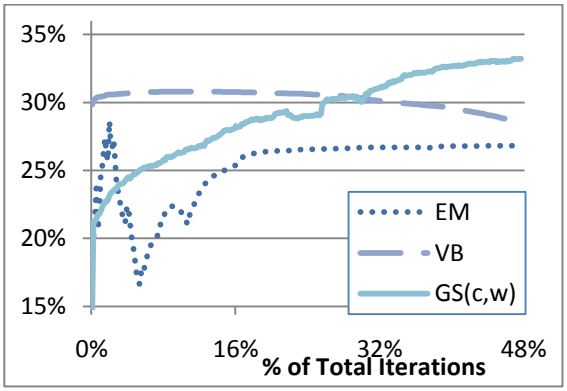


Figure 5: Greedy 1-to-1 accuracy of EM, VB and GS(c,w) over the first 50% of the algorithms' iterations for the Chinese 500k dataset. Note: the percentage of iterations is used here because each algorithms converge at a different number of iterations, thus the progress is scaled accordingly.

The greedy 1-to-1 accuracy actually fluctuates through the run. VB has an interesting dip at around 80% of its iteration before climbing to its max (not showing in the graph). All the Gibbs sampling variations follow a relatively steady hill climb before converging (only GS(c,w) is shown in Figure 5). EM is particularly interesting; Looking at the initial 15% of the algorithm's run, we can see that EM climbs to a "local" max very quickly before dropping and then slowly improving in its accuracy. The greedy 1-to-1 accuracy in the initial top is actually higher than the final convergence value in most runs. This initial peak in value following by a drop and then a slow hill climb in EM for Chinese POS tagging is consistent with the finding in (Johnson, 2007) for English POS tagging.

5 Conclusion and Future Work

We have only scratched the surface of the research in unsupervised techniques in Chinese NLP. We have established a baseline of EM, VB and GS against the CTB 5.0. The experiment shows that for both Chinese and English, GS(c,w) produces the best result. We have also found that Chinese performs rather poorly in the 1-to-1 accuracy when comparing against English in the same data size. We find that in many-to-1 mapping, we have a disproportionate large number of states mapping to individual POS tags comparing to the gold distribution and also in comparison to English against its gold distribution.

Graça et al. (2009) addresses the problem we observe in our resulting tag distributions in our model where EM, VB and GS fails to capture the shape of the true distribution. They propose a Posterior Regularization framework where it poses linear constraints on the posterior expectation. They define a set distributions Q over hidden states with a constraint on the expectation over the features. The log likelihood is penalized using the KL-divergence between the Q distribution and the model. The distributions that their model predicted are far more similar to the gold standard than traditional EM.

Liang and Klein (2009) propose some interesting error analysis techniques for unsupervised POS tagging. One of their analyses on EM is done by observing the approximation errors being created during each iteration of the algorithm's execution. We can also perform these analyses on VB and GS and observe the changes of output tags by starting from the Gold Standard distribution in EM and VB, and gold standard tags in GS. We can then follow how and which set of tags start to deviate from the gold standard. This will allow us to see which categories of errors (ex. noun-verb, adj-adv errors) occur most in these algorithms and how the error progresses.

Acknowledgment: This work is partly supported by the National Science Foundation Grant BCS-0748919. We would also like to thank three anonymous reviewers for their valuable comments.

References

Banko, M., & Moore, R. C. 2004. Part of Speech Tagging in Context. *In Proc. of the 20th International Conference on Computational Linguistics (COLING)*, pp 556-561.

Beal, M. 2003. *Variational Algorithms for Approximate Bayesian Inference*. Ph.D. thesis, Gatsby-Computational Neuroscience unit, University College London.

Besag, J. 2004. An introduction to Markov Chain Monte Carlo methods. *Mathematical Foundations of Speech and Language Processing*, pages 247–270. Springer, New York.

Chang, C.-H., & Chen, C.-D. 1993. HMM-Based Part-Of-Speech Tagging For Chinese Corpora. *Workshop On Very Large Corpora: Academic And Industrial Perspectives.*

Elworthy, D. 1994. Does Baum-Welch Re-estimation Help Taggers? In Proc. of *Applied Natural Language Processing Conference (ANLP)*, pp 53-58.

Gao, J., & Johnson, M. 2008. A comparison of Bayesian estimators for unsupervised Hidden Markov Model POS taggers. *In Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp 344-352.

Geman, S., & Geman, D. 1984. Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp 721–741.

Goldwater, S., & Griffiths, T. 2007. A Fully Bayesian Approach to Unsupervised Part-of-Speech Tagging. *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics (ACL)* , pp 744-751.

Graca, M.J., Ganchev, K., Taskar B. & Pereira, F. 2009. Posterior vs. Parameter Sparsity in Latent Variable. *Advances in Neural Information Processing Systems 22 (NIPS)*. MIT Press.

Haghighi, A., & Klein, D. 2006. Prototype-driven learning for sequence models. *In Proceedings of the Human Language Technology Conference (HLT- NAACL)* , pp 320-327.

Huang, Z., Eidelman, V., Harper, M. 2009. Improving A Simple Bigram HMM Part-of-Speech Tagger by Latent Annotation and Self-Training. In Proc. of *Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL), Companion Volume: Short Papers.*

Johnson, M. 2007. Why Doesn't EM Find Good HMM POS-Taggers? In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pp 296-305.

Liang, P., & Klein, D. 2008. Analyzing the errors of unsupervised learning. *The Forty Sixth Annual Meeting of the Association for Computational Linguistics (ACL)*, pp 879–887. Columbus, OH.

MacKay, D. J., & Peto, L. C. 1994. A Hierarchical Dirichlet Language Model. *Natural Language Engineering*, 1-19.

Manning, C. & Schutze, H. 1999. *Foundations of Statistical Natural Language Processing.* The MIT Press, Cambridge, Massachusetts.

Merialdo, B. 1994. Tagging English text with a probabilistic model. *Computational Linguistics*, 20(2).

Ng, H. T., & Low, J. K. 2004. Chinese Part-Of-Speech Tagging: One-At-A-Time Or All-At-Once? Word-Based Or Character-Based? In Proc. of *EMNLP*.

Thorsten Brants, 2000. TnT - A Statistical Part-of-Speech Tagger. In *Proceedings of the Sixth Applied Natural Language Processing Conference (ANLP)*, Seattle, WA.

Toutanova, K., & Johnson, M. 2007. A Bayesian LDA-based model for semi-supervised. In *Proceedings of NIPS 21* .

Wolfram Mathematica. (2009, 10 3). *Random Number Generation*. http://reference.wolfram.com/mathematica/tutorial/RandomNumberGeneration.html .

Xia, F. 2000. The Part-of-Speech Guidelines for the Penn Chinese Treebank (3.0). University of *Pennsylvania: IRCS Report 00-07.*

Xue, N., Chiou, F.-D., & Palmer, M. 2002. Building a Large-Scale Annotated Chinese Corpus. *Proceedings of the 19th. International Conference on Computational Linguistics (COLING).* Taipei, Taiwan.

Xue, N., Xia, F., Chiou, F.-D., & Palmer, M. 2005. The Penn Chinese TreeBank: Phrase Structure Annotation of a Large Corpus. *Natural Language Engineering*, 11(2), pp 207-238.

The True Score of Statistical Paraphrase Generation

Jonathan Chevelu[1,2] **Ghislain Putois**[2] **Yves Lepage**[3]
(1) GREYC, université de Caen Basse-Normandie
(2) Orange Labs
(3) Waseda University
{jonathan.chevelu,ghislain.putois}@orange-ftgroup.com,
yves.lepage@aoni.waseda.jp

Abstract

This article delves into the scoring function of the statistical paraphrase generation model. It presents an algorithm for exact computation and two applicative experiments. The first experiment analyses the behaviour of a statistical paraphrase generation decoder, and raises some issues with the ordering of n-best outputs. The second experiment shows that a major boost of performance can be obtained by embedding a true score computation inside a Monte-Carlo sampling based paraphrase generator.

1 Introduction

A paraphrase generator is a program which, given a source sentence, produces a new sentence with almost the same meaning. The modification place is not imposed but the paraphrase has to differ from the original sentence.

Paraphrase generation is useful in applications where it is needed to choose between different forms to keep the most fit. For instance, automatic summary can be seen as a particular paraphrasing task (Barzilay and Lee, 2003) by selecting the shortest paraphrase. They can help human writers by proposing alternatives and having them choose the most appropriate (Max and Zock, 2008).

Paraphrases can also be used to improve natural language processing (NLP) systems. In this direction, (Callison-Burch et al., 2006) tried to improve machine translations by enlarging the coverage of patterns that can be translated. In the same way, most NLP systems like information retrieval (Sekine, 2005) or question-answering (Duclaye et al., 2003), based on pattern recognition, can be improved by a paraphrase generator.

Most of these applications need a n-best set of solutions in order to rerank them according to a task-specific criterion.

In order to produce the paraphrases, a promising approach is to see the paraphrase generation problem as a statistical translation problem. In that approach, the target language becomes the same as the source language (Quirk et al., 2004; Bannard and Callison-Burch, 2005; Max and Zock, 2008).

The first difficulty of this approach is the need of a paraphrase table. A paraphrase table is a monolingual version of a translation table in the statistical machine translation (SMT) field. In this field, the difficulty is basically overcome by using huge aligned bilingual corpora like the Europarl (Koehn, 2005) corpus. In the paraphrase generation field, one needs a huge aligned monolingual corpus to build a paraphrase table.

The low availability of such monolingual corpora nurtures researches in order to find heuristics to produce them (Barzilay and Lee, 2003; Quirk et al., 2004). On the other hand, an interesting method proposed by (Bannard and Callison-Burch, 2005) tries to make a paraphrase table using a translation table learned on bilingual corpora. The method uses a well-known heuristic (Lepage and Denoual, 2005) which says that if two sentences have the same translation, then they should be paraphrases of each others.

Another aspect, less studied, is the generation process of paraphrases, *i.e.* the decoding process in SMT. This process is subject to combinatorial

Coling 2008: Poster Volume, pages 144–152,
Beijing, August 2010

explosions. Heuristics are then frequently used to drive the exploration process in the *a priori* intractable high dimensional spaces. On the one hand, these heuristics are used to build a paraphrase step by step according to the paraphrase table. On the other hand, they try to evaluate the relevance of a step according to the global paraphrase generation model. The SMT model score is related to the path followed to generate a paraphrase. Because of the step-by-step computation, different ways can produce the same paraphrase, but with different scores. Amongst these scores, the best one is the true score of a paraphrase according to the SMT model.

Most paraphrase generators use some standard SMT decoding algorithms (Quirk et al., 2004) or some off-the-shelf decoding tools like MOSES. The goal of these decoders is to find the best path in the lattice produced by the paraphrase table. This is basically achieved by using dynamic programming – especially the *Viterbi* algorithm – and beam searching (Koehn et al., 2007). The best paraphrase proposed by these programs is known not to be the optimal paraphrase. One can even question if the score returned is the true score.

We first show in Section 2 that in the particular domain of statistical paraphrase generation, one can compute true *a posteriori* scores of generated paraphrases. We then explore some applications of the true score algorithm in the paraphrase generation field. In Section 3, we show that scores returned by SMT decoders are not always true scores and they plague the ranking of output n-best solutions. In Section 4, we show that the true score can give a major boost for holistic paraphrases generators which do not rely on decoding approaches.

2 True Score Computing

2.1 Context

The phrase based SMT model (Koehn et al., 2003) can be transposed to paraphrase generation as follows:

$$t^* = \arg\max_t P(t) \times P(s|t, B)$$

where s is the source sentence, t the target sentence *i.e.* the paraphrase, t^* the best paraphrase and B a model of the noisy channel between the source and target languages *i.e.* the paraphrase table. This can be decomposed into:

$$t^* \approx \arg\max_{t,I} P(t) \prod_{i \in I} P(s_i^I | t_i^I, B)$$

where I is a partition of the source sentence and x_i^I the i^{th} segment in the sentence x. For a given couple of s, t sentences, it exists several segmentations I with different probabilities.

This is illustrated in Example 1. Depending on the quality of the paraphrase table, one can find up to thousands of paraphrase segments for a source sentence. Note that the generated paraphrases are not always semantically or even syntactically correct, as in P2. P3 illustrates the score evaluation problem: it can be generated by applying to the source sentence the sequences of transformations $\{T1, T2\}$, $\{T1, T4, T5\}$ or even $\{T5, T1, T4\}$ ...

Example 1 Decoding

Source sentence:

The dog runs after the young cat.

Paraphrase table excerpt:

T1: P(the beast | the dog) = 0.8

T2: P(the kitten | the young cat) = 0.7

T3: P(after it | after the) = 0.4

T4: P(the | the young) = 0.05

T5: P(cat | kitten) = 0.1

Some possible generated paraphrases:

P1: the beast runs after the young cat.

P2: *the dog runs after it young cat.

P3: the beast runs after the kitten.

We define the score of a potential paraphrase t following a segmentation I as:

$$Z_t^I = P(t) \prod_{i \in I} P(s_i^I | t_i^I, B)$$

The true score of a potential paraphrase t is defined as:

$$Z_t^* = \max_I Z_t^I$$

Because of high-dimension problems, decoders apply sub-optimal algorithms to search for t^*. They produce estimated solutions over all possible paraphrases t and over all possible segmentations I. Actually, for a given paraphrase t, they consider only some Z_t^I where they should estimate Z_I^*. SMT decoders are overlooking the partitioning step in their computations.

There is no reason for the decoder solution to reach the true score. Troubles arise when one needs the scores of generated paraphrases, for instance when the system must produce an ordered n-best solution. What is the relevance of the estimated scores – and orders – with respect to the true scores – and orders – of the model? Is the true score able to help the generation process?

2.2 Algorithm

Let us first adopt the point of view proposed in (Chevelu et al., 2009). The paraphrase generation problem can be seen as an exploration problem. We seek the best paraphrase according to a scoring function in a space to search by applying successive transformations. This space is composed of states connected by actions. An action is a transformation rule with a place where it applies in the sentence. States are a sentence with a set of possible actions. Applying an action in a given state consists in transforming the sentence of the state and removing all rules that are no more applicable. In this framework, each state, except the root, can be a final state.

The SMT approach fits within this point of view. However, generation and evaluation need not to be coupled any longer. Computing the true score of a generated paraphrase is in reality a task computationally easier than generating the best paraphrases. Once the target result is fixed, the number of sequences transforming the source sentence into the target paraphrase becomes computationally tractable under a reasonable set of assumptions:

A1: the transformation rules have disjoint supports (meaning that no rule in the sequence should transform a segment of the sentence already transformed by one of of the previous applied rules) ;

A2: no reordering model is applied during the paraphrasing transformation.

Under this set of assumptions, the sequence (ordered) of transformation rules becomes a set (unordered) of transformation rules. One can therefore easily determine all the sets of transformation rules from the source sentence to the target paraphrase: they are a subset of the cross-product set of every transformation rule with a source included in the source sentence and with a result included in the target paraphrase. And this cross-product set remains computationally tractable. Note that to guarantee a solution, the corpus of all rules should be augmented with an identity rule for each word of the source sentence (with an associated probability of applicability set to 1) missing in the paraphrase table.

The algorithm for computing *ex post* the true score is given on algorithm 1.

Algorithm 1 Algorithm for true score

Let S be the source sentence.
Let T be the target sentence.
Let $R : s_R \rightarrow t_R$ be a transformation rule

Let $map : (S, T) \rightarrow C$ be a function

Let $C = \{\emptyset\}$

$\forall s_{head} | S = s_{head}.s_{tail}$,

$\forall R \in \{\Omega | s_R = s_{head}, T = t_R.t_{tail}\}$

$C = C \cup (\{R\} \otimes map(S_{tail}, T_{tail}))$

return C

Let $score$ be the scoring function for a transformation rule set

$$truescore_{S,\Omega}(T) = \arg \max_{c \in map(S,T)} (score(c))$$

For our toy example, we would get the steps shown in Example 2.

3 True Score of SMT Decoders

We have shown that it is possible to compute the true score according to the paraphrase model. We now evaluate scores from a state-of-the-art

Example 2 True Score Computation

Generated sets:

$\{R1\}, \{R1, R3\}, \{R1, R2\},$

$\{R1, R4\}, \{R1, R4, R5\},$

$\{R3\},$

$\{R2\},$

$\{R4\}, \{R4, R5\},$

$\{R5\}$

For a better readability, all identity rules are omitted.

The true scores are computed as in the following examples:

$score($ "the dog runs after the small cat." $\rightarrow$
"the beast runs after it small cat")
$= score(\{R1\})$

$score($ "the dog runs after the small cat." $\rightarrow$
"the beast runs after the kitten")
$= \max(score(\{R1, R2\}), score(\{R1, R4, R5\}))$

decoder against this baseline. In particular, we are interested in the order of n-best outputs. We use the MOSES decoder (Koehn et al., 2007) as a representative SMT decoder inside the system described below.

3.1 System description

Paraphrase generation tools based on SMT methods need a language model and a paraphrase table. Both are computed on a training corpus.

The language models we use are n-gram language models with back-off. We use SRILM (Stolcke, 2002) with its default parameters for this purpose. The length of the n-grams is five.

To build a paraphrase table, we use a variant of the construction method via a pivot language proposed in (Bannard and Callison-Burch, 2005). The first step consists in building a bilingual translation table from the aligned corpus. Given a source phrase s^i and another phrase t^i in a different language, a bilingual translation table provides the two probabilities $p(s^i|t^i)$ and $p(t^i|s^i)$. We use GIZA++ (Och and Ney, 2003) with its default parameters to produce phrase alignments. The paraphrase table is then built from the phrase translation table. The probability for a phrase s^i to be paraphrased by a phrase s'^i in the same language is estimated by the sum of each round-trip from s^i to s'^i through any phrase t^i of a pivot language.

The construction of this table is very simple. Given a bilingual translation table sorted by pivot phrases, the algorithm retrieves all the phrases linked with the same pivot (named a *pivot cluster*). For each ordered pair of phrases, the program assigns a probability that is the product of there probabilities. This process realizes a self-join of the bilingual translation table. It produces a paraphrase table composed of tokens, instead of items. The program just needs to sum up all probabilities for all entries with identical paraphrase tokens to produce the final paraphrase table.

Three heuristics are used to prune the paraphrase table. The first heuristic prunes any entry in the paraphrase table composed of tokens with a probability lower than a threshold ϵ. The second, called *pruning pivot heuristic*, consists in deleting all pivot clusters larger than a threshold τ. The last heuristic keeps only the κ most probable paraphrases for each source phrase in the final paraphrase table. For this study, we empirically fix $\epsilon = 10^{-5}$, $\tau = 200$ and $\kappa = 20$.

The MOSES scoring function is set by four weighting factors $\alpha_\Phi, \alpha_{LM}, \alpha_D, \alpha_W$. Conventionally, these four weights are adjusted during a tuning step on a training corpus. The tuning step is inappropriate for paraphrasing because there is no such tuning corpus available. We empirically set $\alpha_\Phi = 1$, $\alpha_{LM} = 1$, $\alpha_D = 10$ and $\alpha_W = 0$. This means that the paraphrase table and the language model are given the same weight, no reordering is allowed and no specific sentence length is favored.

3.2 Experimental Protocol

For experiments reported in this paper, we use one of the largest, multi-lingual, freely available aligned corpus, Europarl (Koehn, 2005). It consists of European parliament debates. We choose French as the language for paraphrases and English as the pivot language. For this pair of languages, the corpus consists of 1,723,705 sentences. Note that the sentences in this corpus are long, with an average length of 30 words per French sentence and 27.8 for English. We randomly extract 100 French sentences as a test cor-

pus.

For each source sentence from the test corpus, the SMT decoder tries to produce a 100-best distinct paraphrase sequence. Using the algorithm 1, we compute the true score of each paraphrase and rerank them. We then compare orders output by the decoder with the true score order by using the Kendall rank correlation coefficient (τ_A) (Kendall, 1938). In this context, the Kendall rank correlation coefficient considers each couple of paraphrases and checks if their relative order is preserved by the reranking. The τ_A formula is:

$$\tau_A = \frac{n_p - n_i}{\frac{1}{2}n(n-1)}$$

where n_p the number of preserved orders, n_d the number of inverted orders and n the number of elements in the sequence. The coefficient provides a score – between -1 and 1 – that can be interpreted as a correlation coefficient between the two orders. In order to compare same length sequences, we filter out source sentences when MOSES can not produce enough distinct paraphrases. The test corpus is therefore reduced to 94 sentences.

3.3 Results

The evolution of τ_A means relative to the length of the n-best sequence is given Figure 1. The τ_A means drops to 0.73 with a standard deviation of 0.41 for a 5-best sequence which means that the orders are clearly different but not decorrelated.

A finer study of the results reveals that amongst the generated paraphrases, 32% have seen their score modified. 18% of the MOSES 1-best paraphrases were not optimal anymore after the true score reranking. After reranking, the old top best solutions have dropped to a mean rank of 2.0 ± 17.7 (40th rank at worse). When considering only the paraphrases no longer optimal, they have dropped to a mean rank of 6.8 ± 12.9.

From the opposite point of view, new top paraphrases after reranking have come from a mean rank of 4.4 ± 12.1. When considering only the paraphrases that were not optimal, they have come from a mean rank of 21.2 ± 23.5. Some have come from the 67th rank. Even an *a posteriori* reranking would not have retrieved this top solution if the size of MOSES n-best list were too short. This

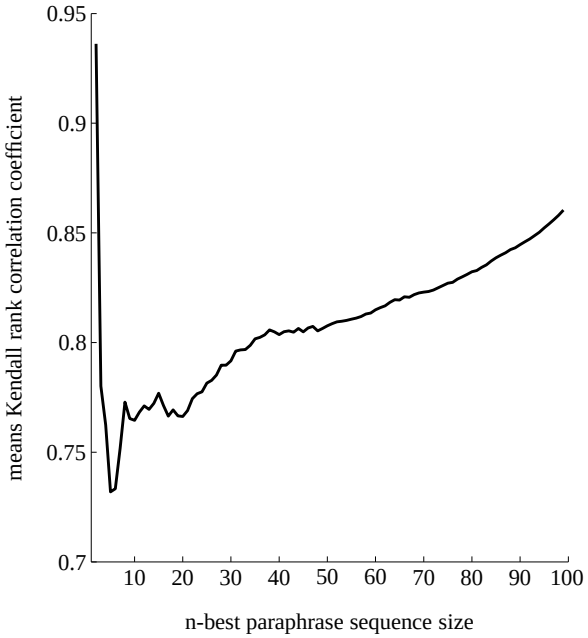


Figure 1: Evolution of τ_A means relative to the length of the n-best sequence

advocates for a direct embedding of the true score function inside the generation process.

In this section we have shown that MOSES scores are not consistent with the true score as expected from the paraphrase model. In particular, the n-best paraphrase sequence computed by MOSES is not trustworthy while it is an input for the task system.

4 True Score to boost Monte-Carlo based Paraphrase Generation

There exist other less common approaches more lenient than the *Viterbi* algorithm, which are holistic, *i.e.* they work on the whole sentence rather than step-by-step. The *Monte-Carlo based Paraphrase Generation* algorithm (MCPG) proposed in (Chevelu et al., 2009) turns out to be an interesting algorithm for the study of paraphrase generation. It does not constraint the scoring function to be incremental. In this section, we embed the non incremental true score function in MCPG to drive the generation step and produce n-best orders compliant with the paraphrase model, and show that the true score function can be used to provide a major boost to the performance of such

an algorithm.

4.1 Description

The MCPG algorithm is a derivative of the *Upper Confidence bound applied to Tree* algorithm (UCT). UCT (Kocsis and Szepesvári, 2006), a *Monte-Carlo* planning algorithm, has recently become popular in two-player game problems.

UCT has some interesting properties:

- it expands the search tree non-uniformly and favours the most promising sequences, without pruning branch;
- it can deal with high branching factors;
- it is an any-time algorithm and returns best solutions found so far when interrupted;
- it does not require expert domain knowledge to evaluate states.

These properties make it ideally suited for problems with high branching factors and for which there is no strong evaluation function.

For the same reasons, this algorithm is interesting for paraphrase generation. In particular, it does not put constraint on the scoring function. A diagram of the MCPG algorithm is presented Figure 2.

The main part of the algorithm is the sampling step. An episode of this step is a sequence of states and actions, $s_1, a_1, s_2, a_2, \ldots, s_T$, from the root state to a final state. Basically, a state is a partially generated paraphrase associated with a set of available actions. A final state is a potential paraphrase. An action is a transformation rule from the paraphrase table. During an episode construction, there are two ways to select the action a_i to perform from a state s_i.

If the current state was already explored in a previous episode, the action is selected according to a compromise between exploration and exploitation. This compromise is computed using the UCB-Tunned formula (Auer et al., 2001) associated with the RAVE heuristic (Gelly and Silver, 2007). If the current state is explored for the first time, its score is estimated using *Monte-Carlo* sampling. In other words, to complete the

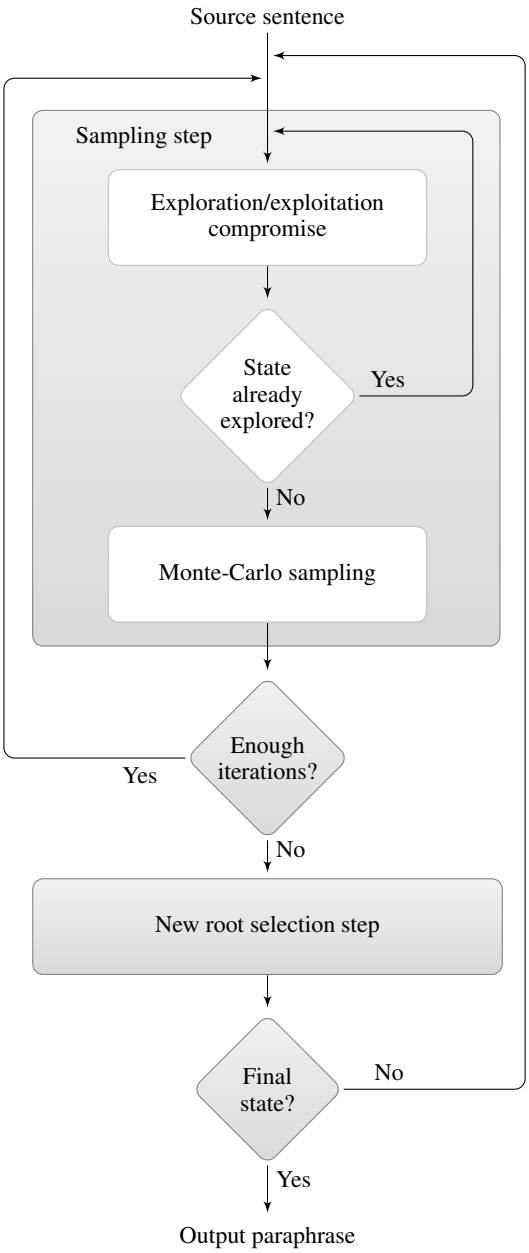


Figure 2: The MCPG algorithm.

episode, the actions $a_i, a_{i+1}, \ldots, a_{T-1}, a_T$ are selected randomly until reaching a final state.

At the end of each episode, a reward is computed for the final state s_T using a scoring function, and the value of each (state, action) pair of the episode is updated. Then, the algorithm computes another episode with the new values.

Periodically, the sampling step is stopped and the best action at the root state is selected. This action is then definitively applied and a sampling is restarted from the new root state. The action sequence is incrementally built and selected after being sufficiently sampled. For our experiment, we have chosen to stop sampling regularly after a fixed amount η of episodes.

The adaptation of the original algorithm takes place in the (state, action) value updating procedure. Since the goal of the algorithm is to maximise a scoring function, it uses the maximum reachable score from a state as value instead of the score expectation. This algorithm suits the paradigm recalled in Section 2 for paraphrase generation.

To provide scores comparable with the paraphrase model scores, the standard version of MCPG has to apply rules until the whole source sentence is covered. With this behaviour, MCPG acts in a monolingual "translator" mode.

The embedding of the true score algorithm in MCPG has given meaningful scores to all states. The algorithm needs not to "translate" the whole sentence to get a potential paraphrase and its score. This MCPG algorithm in "true-score" mode can choose to stop its processing with segments still unchanged, which solves, amongst others, out-of-vocabulary questions found in decoder-based approaches.

4.2 Experimental Protocol

For this experiment, we reuse the paraphrase table and the corpora generated for the experiment presented in Section 3.2;

We compare the 1-best outputs from MOSES reranked by the true score function and from MCPG in both "translator" and "true-score" modes. For MCPG systems, we set the following parameters: $\eta = 100{,}000$ iterations.

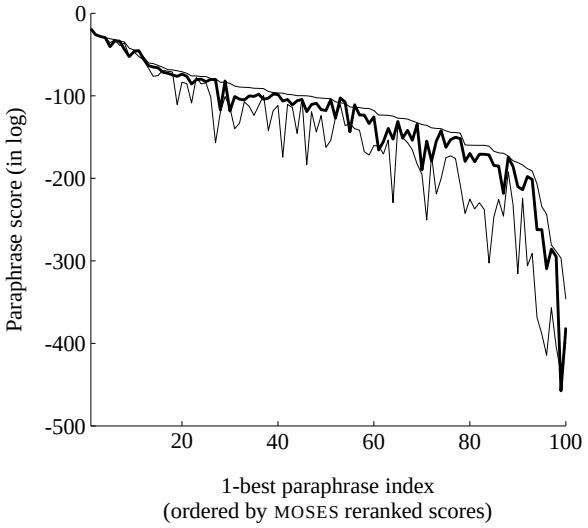


Figure 3: Comparison of paraphrase generators. Top: the MOSES baseline; middle and bold: the "true-score" MCPG; down: the "translator" MCPG. The use of "true-score" improves the MCPG performances. MCPG reaches MOSES performance level.

4.3 Results

Figure 3 presents a comparison between the scores from each systems, ordered by MOSES reranked scores.

The boost of performance gained by using true scores inside the MCPG algorithm reaches a means of 28.79 with a standard deviation of 34.19. The mean difference between "true-score" MCPG and MOSES is -14.13 (standard deviation 19.99). Although the performance remains inferior to the MOSES true score baseline, it still leads to an improvement over the "translator" MCPG system. The later system has a mean difference of performance with MOSES of -42.92 (standard deviation of 40.14).

The true score reduces the number of transformations needed to generate a paraphrase, which simplifies the exploration task. Moreover, it reduces the number of states in the exploration space: two sets of transformations producing the same paraphrase now leads to the same state. These points explain why MCPG has become more efficient.

Although MCPG is improved by embedding the

true score algorithm, there is still room for improvement. In its current version, MCPG does not adapt the number of exploration episodes to the input sentence.

5 Conclusion and perspectives

In this paper, we have developed a true scoring algorithm adapted to the statistical paraphrase generation model. We have studied its impacts on a common SMT decoder and a Monte-Carlo sampling based paraphrase generator. It has revealed that the n-best outputs by SMT decoders were not viable. It has also proved useful in simplifying the exploration task and in improving holistic paraphrase generators.

Thanks to the boost introduced by the true score algorithm in holistic paraphrase generators, their performances are now on a par with scores produced by statistical translation decoders. Moreover, they produce guaranteed ordering, and enable the integration of a global task scoring function, which seems still out of reach for decoder-based systems.

A more general problem remains open: what do the scores and the orders output by the model mean when compared to a human subjective evaluation?

In preliminary results on our test corpus, less than 37% of the MOSES generated paraphrases can be considered both syntactically correct and semantically a paraphrase of their original sentence. One could study the relations between scores from the model and subjective evaluations to create predictive regression models. The true score algorithm can autonomously score existing paraphrase corpora which could be used to adapt the SMT tuning step for paraphrase generation.

We note that the hundredth best paraphrases from MOSES have a score close to the best paraphrase: the mean difference is 5.9 (standard deviation 4.5) on our test corpus. This is smaller than the mean difference score between MOSES and MCPG. In (Chevelu et al., 2009), both systems were rated similar by a subjective evaluation. One could question the relevance of small score differences and why the best paraphrase should be selected instead of the hundred next ones. Given the current state of the art, the next step to improve paraphrase generation does not lie in score optimisation but in refining the model and its components: the language model and the paraphrase table.

Human based evaluations reveal that the current most important issue of paraphrase generation lies in the syntax (Chevelu et al., 2009). It seems difficult to assess the syntax of a potential paraphrase while not considering it as a whole, which is impossible with a local scoring function inherent to the SMT decoding paradigm. Holistic paraphrase generators have now reached a level of performance comparable to SMT decoders, without suffering from their limitations. They are paving the way for experiments with more complex semantic and linguistic models to improve paraphrase generation.

References

Auer, P., N. Cesa-Bianchi, and C. Gentile. 2001. Adaptive and self-confident on-line learning algorithms. *Machine Learning*.

Bannard, Colin and Chris Callison-Burch. 2005. Paraphrasing with bilingual parallel corpora. In *ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 597–604, Morristown, NJ, USA. Association for Computational Linguistics.

Barzilay, Regina and Lillian Lee. 2003. Learning to paraphrase: An unsupervised approach using multiple-sequence alignment. In *HLT-NAACL 2003: Main Proceedings*, pages 16–23.

Callison-Burch, Chris, Philipp Koehn, and Miles Osborne. 2006. Improved statistical machine translation using paraphrases. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 17–24, Morristown, NJ, USA. Association for Computational Linguistics.

Chevelu, Jonathan, Thomas Lavergne, Yves Lepage, and Thierry Moudenc. 2009. Introduction of a new paraphrase generation tool based on Monte-Carlo sampling. In Su, Keh-Yih, Jian Su, Janyce Wiebe, and Haizhou Li, editors, *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 249–252, Singapoure, August. Association for Computational Linguistics.

Duclaye, Florence, François Yvon, and Olivier Collin. 2003. Learning paraphrases to improve a question-answering system. In *In Proceedings of the 10th Conference of EACL Workshop Natural Language Processing for Question-Answering*, page 3541.

Gelly, Sylvain and David Silver. 2007. Combining online and offline knowledge in UCT. In *24th International Conference on Machine Learning (*ICML*'07)*, pages 273–280, June.

Kendall, Maurice G. 1938. A New Measure of Rank Correlation. *Biometrika*, 1–2(30):81–89, June.

Kocsis, Levente and Csaba Szepesvári. 2006. Bandit based monte-carlo planning. In *17th European Conference on Machine Learning, (*ECML*'06)*, pages 282–293, September.

Koehn, Philipp, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL)*, pages 48–54, Edmonton, May. Association for Computational Linguistics.

Koehn, Philipp, Hieu Hoang, Alexandra Birch Mayne, Christopher Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Annual Meeting of the Association for Computation Linguistics (ACL), Demonstration Session*, pages 177–180, June.

Koehn, Philipp. 2005. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of MT Summit*.

Lepage, Yves and Etienne Denoual. 2005. Automatic generation of paraphrases to be used as translation references in objective evaluation measures of machine translation. In *IWP2005*.

Max, Aurélien and Michael Zock. 2008. Looking up phrase rephrasings via a pivot language. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 97–104, Manchester, UK, August. Coling 2008 Organizing Committee.

Och, Franz Josef and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Quirk, Chris, Chris Brockett, and Bill Dolan. 2004. Monolingual machine translation for paraphrase generation. In Lin, Dekang and Dekai Wu, editors, *the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 142–149., Barcelona, Spain, 25-26 July. Association for Computational Linguistics.

Sekine, Satoshi. 2005. Automatic paraphrase discovery based on context and keywords between ne pairs. In *Proceedings of International Workshop on Paraphrase (IWP2005)*.

Stolcke, Andreas. 2002. Srilm – an extensible language modeling toolkit. In *Proceedings of International Conference on Spoken Language Processing*.

Acquisition of Unknown Word Paradigms for Large-Scale Grammars

Kostadin Cholakov
University of Groningen
The Netherlands
k.cholakov@rug.nl

Gertjan van Noord
University of Groningen
The Netherlands
g.j.m.van.noord@rug.nl

Abstract

Unknown words are a major issue for large-scale grammars of natural language. We propose a machine learning based algorithm for acquiring lexical entries for all forms in the paradigm of a given unknown word. The main advantages of our method are the usage of word paradigms to obtain valuable morphological knowledge, the consideration of different contexts which the unknown word and all members of its paradigm occur in and the employment of a full-blown syntactic parser and the grammar we want to improve to analyse these contexts and provide elaborate syntactic constraints. We test our algorithm on a large-scale grammar of Dutch and show that its application leads to an improved parsing accuracy.

1 Introduction

In this paper, we present an efficient machine learning based method for automated lexical acquisition (LA) which improves the performance of large-scale computational grammars on real-life tasks.

Our approach has three main advantages which distinguish it from other methods applied to the same task. First, it enables the acquisition of the *whole paradigm* of a given unknown word while other approaches are only concerned with the particular word form encountered in the data subject to LA. Second, we analyse *different contexts* which the unknown word occurs in. Third, the analysis of these contexts is provided by a *full-blown syntactic parser* and the *grammar* we aim to improve which gives the grammar the opportunity to participate *directly* in the LA process.

Our method achieves an F-measure of 84.6% on unknown words in experiments with the wide-coverage Alpino grammar (van Noord, 2006) of Dutch. The integration of this method in the parser leads to a 4.2% error reduction in terms of labelled dependencies.

To predict a lexical entry for a given unknown word, we take into account two factors– its morphology and the syntactic constraints imposed by its context. As for the former, the acquisition of the whole paradigm provides us with a valuable source of morphological information. If we were to deal with only one form of the unknown word, this information would not be accessible.

Further, looking at different contexts of the unknown word gives us the possibility to work with linguistically diverse data and to incorporate more syntactic information into the LA process. Cases where this is particularly important include morphologically ambiguous words and verbs which subcategorize for various types of syntactic arguments. We also consider contexts of the other members of the paradigm of the unknown word in order to increase the amount of linguistic data our method has access to.

Finally, the usage of a *full-blown syntactic parser* and the *grammar* we want to acquire lexical entries for has two advantages. First, LA can benefit from the high-quality analyses such a parser produces and the elaborate syntactic information they provide. Second, this information comes directly from the grammar, thus allowing the LA process to make predictions based on what the grammar considers to be best suited for it.

Coling 2008: Poster Volume, pages 153–161,
Beijing, August 2010

The remainder of the paper is organised as follows. Section 2 describes the basic steps in our LA algorithm. Section 3 presents initial experiments conducted with Alpino and shows that the main problems our LA method encounters are the acquisition of morphologically ambiguous words, the learning of the proper subcategorization frames for verbs and the acquisition of particular types of adjectives. In Section 4 we make extensive use of the paradigms of the unknown words to develop specific solutions for these problems. Section 5 describes experiments with our LA method applied to a set of real unknown words. Section 6 provides a comparison between our approach and work previously done on LA. This section also discusses the application of our method to other systems and languages.

2 Basic Algorithm

The Alpino wide-coverage dependency parser is based on a large stochastic attribute value grammar. The grammar takes a 'constructional' approach, with rich lexical representations stored in the lexicon and a large number of detailed, construction specific rules (about 800). Currently, the lexicon contains about 100K lexical entries and a list of about 200K named entities. Each word is assigned one or more lexical types. For example, the verb *amuseert* (to amuse) is assigned two lexical types– *verb(hebben,sg3,intransitive)* and *verb(hebben,sg3,transitive)*– because it can be used either transitively or intransitively. The other type features indicate that it is a present third person singular verb and it forms perfect tense with the auxiliary verb *hebben*.

The goal of our LA method is to *assign* the correct lexical type(s) to a given unknown word. The method takes into account only open-class lexical types: nouns, adjectives and verbs, under the assumption that the grammar is already able to handle all closed-class cases. We call the types considered by our method *universal types*. The adjectives can be used as adverbs in Dutch and thus, we do not consider the latter to be an open class.

We employ a ME-based classifier which, for some unknown word, takes various morphological and syntactic features as input and outputs lexical types. The probability of a lexical type *t*, given an unknown word and its context *c* is:

$$(1) \qquad p(t|c) = \frac{exp(\sum_i \Theta_i f_i(t,c))}{\sum_{t' \in T} exp(\sum_i \Theta_i f_i(t',c))}$$

where $f_i(t, c)$ may encode arbitrary characteristics of the context and $< \Theta_1, \Theta_2, ... >$ can be evaluated by maximising the pseudo-likelihood on a training corpus (Malouf, 2002).

Table 1 shows the features for the noun *inspraakprocedures* (consultation procedures). Row **(i)** contains 4 separate features derived from the prefix of the word and 4 other suffix features are given in row **(ii)**. The two features in rows **(iii)** and **(iv)** indicate whether the word starts with a particle and if it contains a hyphen, respectively.

Another source of morphological features is the paradigm of the unknown word which provides information that is otherwise inaccessible. For example, in Dutch, neuter nouns always take the *het* definite article while all other noun forms are used with the *de* article. Since the article is distinguishable only in the singular noun form, the correct article of a word, assigned a plural noun type, can be determined if we know its singular form.

We adopt the method presented in Cholakov and van Noord (2009) where a finite state morphology is applied to generate the paradigm(s) of a given word. The morphology does not have access to any additional linguistic information and thus, it generates all possible paradigms allowed by the word structure. Then, the number of search hits Yahoo returns for each form in a given paradigm is combined with some simple heuristics to determine the correct paradigm(s).

However, we make some modifications to this method because it deals only with *regular* morphological phenomena. Though all typical irregularities are included in the Alpino lexicon, there are cases of irregular verbs composed with particles which are not listed there. One such example is the irregular verb *meevliegen* (to fly with someone) for which no paradigm would be generated.

To avoid this, we use a list of common particles to strip off any particle from a given unknown word. Once we have removed a particle, we check if what is left from the word is listed in the lexicon as a verb (e.g. *vliegen* in the case of ***mee****vliegen*). If so, we extract all members of its paradigm from

Features
i) i, in, ins, insp
ii) s, es, res, ures
iii) particle_yes #in this case *in*
iv) hyphen_no
v) noun⟨de,pl⟩
vi) noun(de,count,pl), tmp_noun(de,count,sg)
vii) noun(de), noun(count), noun(pl), tmp_noun(de) tmp_noun(count), tmp_noun(sg)

Table 1: Features for *inspraakprocedures*

the lexicon and use them to build the paradigm of the unknown word. All forms are validated by using the same web-based heuristics as in the original model of Cholakov and van Noord (2009).

A single paradigm is generated for *inspraakprocedures* indicating that this word is a plural *de* noun. This information is explicitly used as a feature in the classifier which is shown in row **(v)** of Table 1.

Next, we obtain syntactic features for *inspraakprocedures* by extracting a number of sentences which it occurs in from large corpora or Internet. These sentences are parsed with a different 'mode' of Alpino where this word is assigned all universal types, i.e. it is treated as being maximally *ambiguous*. For each sentence only the best parse is preserved. Then, the lexical type that has been assigned to *inspraakprocedures* in this parse is stored. During parsing, Alpino's POS tagger (Prins and van Noord, 2001) keeps filtering implausible type combinations. For example, if a determiner occurs before the unknown word, all verb types are typically not taken into consideration. This heavily reduces the computational overload and makes parsing with universal types computationally feasible. When all sentences have been parsed, a list can be drawn up with the types that have been used and their frequency:

(2)
noun(de,count,pl) 78
tmp_noun(de,count,sg) 7
tmp_noun(het,count,pl) 6
proper_name(pl,'PER') 5
proper_name(pl,'ORG') 3
verb(hebben,pl,vp) 1

The lexical types assigned to *inspraakprocedures* in at least 80% of the parses are used as features in the classifier. These are the two features in row **(vi)** of Table 1. Further, as illustrated in row **(vii)**, each attribute of the considered types is also taken as a separate feature. By doing this, we let the grammar decide which lexical type is best suited for a given unknown word. This is a new and effective way to include the *syntactic constraints* of the context in the LA process.

However, for the parsing method to work properly, the disambiguation model of the parser needs to be adapted. The model heavily relies on the lexicon and it has learnt preferences how to parse certain phrases. For example, it has learnt a preference to parse prepositional phrases as verb complements, if the verb includes such a subcategorization frame. This is problematic when parsing with universal types. If the unknown word is a verb and it occurs together with a PP, it would always get analysed as a verb which subcategorizes for a PP.

To avoid this, the disambiguation model is retrained on a specific set of sentences meant to make it more robust to input containing many unknown words. We have selected words with low frequency in large corpora and removed them temporarily from the Alpino lexicon. Less frequent words are typically not listed in the lexicon and the selected words are meant to simulate their behaviour. Then, all sentences from the Alpino treebank which contain these words are extracted and used to retrain the disambiguation model.

3 Initial Experiments and Evaluation

To evaluate the performance of the classifier, we conduct an experiment with a target type inventory of 611 universal types. A type is considered universal only if it is assigned to at least 15 distinct words occurring in large Dutch newspaper corpora ($\sim$16M sentences) automatically parsed with Alpino.

In order to train the classifier, 2000 words are temporarily removed from the Alpino lexicon. The same is done for another 500 words which are used as a test set. All words have between 50 and 100 occurrences in the corpora. This selection is again meant to simulate the behaviour of unknown words. Experiments with a minimum lower than 50 occurrences have shown that this is a reasonable threshold to filter out typos, words written together, etc.

The classifier yields a probability score for each predicted type. Since a given unknown word can have more than one correct type, we want to predict multiple types. However, the least frequent types, accounting together for less than 5% of probability mass, are discarded.

We evaluate the results in terms of precision and recall. Precision indicates how many types found by the method are correct and recall indicates how many of the lexical types of a given word are actually found. The presented results are the average precision and recall for the 500 test words.

Additionally, there are three baseline methods:

- *Naive*– each unknown word is assigned the most frequent type in the lexicon: *noun(de,count,sg)*
- *POS tagger*– the unknown word is given the type most frequently assigned by the Alpino POS tagger in the parsing stage
- *Alpino*– the unknown word is assigned the most frequently used type in the parsing stage

The overall results are given in Table 2. Table 3 shows the results for each POS in our model.

Model	Precision(%)	Recall(%)	F-measure(%)
Naive	19.60	18.77	19.17
POS tagger	30	26.21	27.98
Alpino	44.60	37.59	40.80
Our model	86.59	78.62	82.41

Table 2: Overall experiment results

POS	Precision(%)	Recall(%)	F-measure(%)
Nouns	93.83	88.61	91.15
Adjectives	75.50	73.12	74.29
Verbs	77.32	55.37	64.53

Table 3: Detailed results for our model

Our LA method clearly improves upon the baselines. However, as we see in Table 3, adjectives and especially verbs remain difficult to predict.

The problems with the former are due to the fact that Alpino employs a rather complicated adjective system. The classifier has difficulties distinguishing between 3 kinds of adjectives: **i)** adjectives which can attach to and modify verbs and verbal phrases (VPs) (3-a), **ii)** adjectives which can attach to verbs and VPs but modify one of the complements of the verb, typically the subject (3-b) and **iii)** adjectives which cannot attach to verbs and VPs (3-c).

(3)
a. De hardloper loopt *mooi*.
DET runner walks nice
'The runner runs nicely = The runner has a good running technique'
b. Hij loopt *dronken* naar huis.
he walks drunk to home
'He walks home drunk = He is walking home while being drunk'
c. *Hij loopt *nederlandstalig*.
he walks Dutch speaking
'He walks Dutch speaking.'

Each of these is marked by a special attribute in the lexical type definitions– *adv*, *padv* and *nonadv*, respectively. Since all three of them are seen in 'typical' adjectival contexts where they modify nouns, it is hard for the classifier to make a distinction. The predictions appear to be arbitrary and there are many cases where the unknown word is classified both as a *nonadv* and an *adv* adjective. It is even more difficult to distinguish between *padv* and *adv* adjectives since this is a solely semantic distinction.

The main issue with verbs is the prediction of the correct subcategorization frame. The classifier tends to predict mostly transitive and intransitive verb types. As a result, it either fails to capture infrequent frames which decreases the recall or, in cases where it is very uncertain what to predict, it assigns a lot of types that differ only in the subcat frame, thus damaging the precision. For example, *onderschrijf* ('to agree with') has 2 correct subcat frames but receives 8 predictions which differ only in the subcat features.

One last issue is the prediction, in some rare cases, of types of the wrong POS for morphologically ambiguous words. In most of these cases adjectives are wrongly assigned a past participle type but also some nouns receive verb predictions. For instance, *OESO-landen* ('countries of the OESO organisation') has one correct noun type but because *landen* is also the Dutch verb for 'to land' the classifier wrongly assigns a verb type as well.

4 Improving LA

4.1 POS Correction

Since the vast majority of wrong POS predictions has to do with the assignment of incorrect verb types, we decided to explicitly use the generated verb paradigms as a filtering mechanism. For each word which is assigned a verb type, we check if there is a verb paradigm generated for it. If not, all verb types predicted for the word are discarded.

In very rare cases a word is assigned *only* verb types and therefore, it ends up with no predictions. For such words, we examine the ranked list of predicted types yielded by the classifier and the word receives the non-verb lexical type with the highest probability score. If this type happens to be an adjective one, we first check whether there is an adjective paradigm generated for the word in question. If not, the word gets the noun type with the highest probability score.

The same procedure is also applied to all words which are assigned an adjective type. However, it is not used for words predicted to be nouns because the classifier is already very good at predicting nouns. Further, the generated noun paradigms are not reliable enough to be a filtering mechanism because there are mass nouns with no plural forms and thus with no paradigms generated.

Another modification we make to the classifier output has to do with the fact that past participles (psp) in Dutch can also be used as adjectives. This systematic ambiguity, however, is not treated as such in Alpino. Each psp should also have a separate adjective lexical entry but this is not always the case. That is why, in some cases, the classifier fails to capture the adjective type of a given psp. To account for it, all words predicted to be past participles but not adjectives are assigned two additional adjective types– one with the *nonadv* and one with the *adv* feature. For reasons explained later on, a type with the *padv* feature is not added.

After the application of these techniques, all cases of words wrongly predicted to be verbs or adjectives have been eliminated.

4.2 Guessing Subcategorization Frames

Our next step is to guess the correct subcategorization feature for verbs. Learning the proper subcat frame is well studied (Brent, 1993; Manning, 1993; Briscoe and Caroll, 1997; Kinyon and Prolo, 2002; O'Donovan et al., 2005). Most of the work follows the 'classical' Briscoe and Caroll (1997) approach where the verb and the subcategorized complements are extracted from the output analyses of a probabilistic parser and stored as syntactic patterns. Further, some statistical techniques are applied to select the most probable frames out of the proposed syntactic patterns.

Following the observations made in Korhonen et al. (2000), Lapata (1999) and Messiant (2008), we employ a maximum likelihood estimate (MLE) from observed relative frequencies with an empirical threshold to filter out low probability frames. For each word predicted to be a verb, we look up the verb types assigned to it during the parsing with universal types. Then, the MLE for each subcat frame is determined and only frames with MLE of 0.2 and above are considered. For example, *jammert* (to moan.3SG.PRES) is assigned a single type– *verb(hebben,sg3,intransitive)*. However, the correct subcat features for it are *intransitive* and *sbar*. Here is the list of all verb types assigned to *jammert* during the parsing with universal types:

(4)
verb(hebben,sg3,intransitive) 48
verb(hebben,sg3,transitive) 15
verb(hebben,past(sg),np_sbar) 3
verb(hebben,past(sg),tr_sbar) 3
verb(zijn,sg3,intransitive) 2
verb(hebben,past(sg),ld_pp) 2
verb(hebben,sg3,sbar) 1

The MLE for the intransitive subcat feature is 0.68 and for the transitive one– 0.2. All previously predicted verb types are discarded and each considered subcat frame is used to create a new lexical type. That is how *jammert* gets two types at the end– the correct *verb(hebben,sg3,intransitive)* and the incorrect *verb(hebben,sg3,transitive)*. The *sbar* frame is wrongly discarded.

To avoid such cases, the generated word paradigms are used to increase the number of contexts observed for a given verb. Up to 200 sentences are extracted for each form in the paradigm of a given word predicted to be a verb. These sentences are again parsed with the universal types and then, the MLE for each subcat frame is recal-

culated.

We evaluated the performance of our MLE-based method on the 116 test words predicted to be verbs. We extracted the subcat features from their type definitions in the Alpino lexicon to create a gold standard of subcat frames. Additionally, we developed two baseline methods: **i)** all frames assigned during parsing are considered and **ii)** each verb is taken to be both transitive and intransitive. Since most verbs have both or one of these frames, the purpose of the second baseline is to see if there is a simpler solution to the problem of finding the correct subcat frame. The results are given in Table 4.

Model	Precision(%)	Recall(%)	F-measure(%)
all frames	16.76	94.34	28.46
tr./intr.	62.29	69.17	65.55
our model	85.82	67.28	75.43

Table 4: Subcat frames guessing results

Our method significantly outperforms both baselines. It is able to correctly identify the transitive and/or the intransitive frames. Since they are the most frequent ones in the test data, this boosts up the precision. However, the method is also able to capture other, less frequent subcat frames. For example, after parsing the additional sentences for *jammert*, the *sbar* frame had enough occurrences to get above the threshold. The MLE for the transitive one, on the other hand, fell below 0.2 and it was correctly discarded.

4.3 Guessing Adjective Types

We follow a similar approach for finding the correct adjective type. It should be noted that the distinction among *nonadv*, *adv* and *padv* does not exist for every adjective form. Most adjectives in Dutch get an *-e* suffix when used attributively– *de mooi**e**/mooier**e**/mooist**e** jongen* (the nice/nicer/nicest boy). Since these inflected forms can only occur before nouns, the distinction we are dealing with is not relevant for them. Thus we are only interested in the noninflected base, comparative and superlative adjective forms.

One of the possible output formats of Alpino is dependency triples. Here is the output for the sentence in (3-a):

(5)
```
verb:loop|hd/su|noun:hardloper
noun:hardloper|hd/det|det:de
verb:loop|hd/mod|adj:mooi
verb:loop|–/–|punct:.
```

Each line is a single dependency triple. The line contains three fields separated by the '|' character. The first field contains the root of the head word and its POS, the second field indicates the type of the dependency relation and the third one contains the root of the dependent word and its POS. The third line in (5) shows that the adjective *mooi* is a modifier of the head, in this case the verb *loopt*. Such a dependency relation indicates that this adjective can modify a verb and therefore, it belongs to the *adv* type.

As already mentioned, *padv* adjectives cannot be distinguished from the ones of the *adv* kind. That is why, if the classifier has decided to assign a *padv* type to a given unknown word, we discard all other adjective types assigned to it (if any) and do not apply the technique described below to this word.

For each of the 59 words assigned an non-inflected adjective type after the POS correction stage, we extract up to 200 sentences for all non-inflected forms in its paradigm. These sentences are parsed with Alpino and the universal types and the output is dependency triples. All triples where the unknown word occurs as a dependent word in a head modifier dependency (*hd/mod*, as shown in (5)) and its POS is adjective are extracted from the parse output. We calculate the MLE of the cases where the head word is a verb, i.e. where the unknown word modifies a verb. If the MLE is 0.05 or larger, the word is assigned an *adv* lexical type.

For example, the classifier correctly identifies the word *doortimmerd* (solid) as being of the *adjective(no_e(nonadv))* type but it also predicts the *adjective(no_e(adv))*[1] type for it. Since we have not found enough sentences where this word modifies a verb, the latter type is correctly discarded. Our technique produced correct results for 53 out of the 59 adjectives processed.

[1]The *no_e* type attribute denotes a noninflected base adjective form.

4.4 Improved Results and Discussion

Table 5 presents the results obtained after applying the improvement techniques described in this section to the output of the classifier (the 'Model 2' rows). For comparison, we also give the results from Table 3 again (the 'Model 1' rows). The numbers for the nouns happen to remain unchanged and that is why they are not shown in Table 5.

POS	Models	Prec.(%)	Rec.(%)	F-meas.(%)
Adj	Model 1	75.50	73.12	74.29
	Model 2	85.16	80.16	82.58
Verbs	Model 1	77.32	55.37	64.53
	Model 2	80.56	56.24	66.24
Overall	Model 1	86.59	78.62	82.41
	Model 2	89.08	80.52	84.58

Table 5: Improved results

The automatic addition of adjective types for past participles improved significantly the recall for adjectives and our method for choosing between *adv* and *nonadv* types caused a 10% increase in precision.

However, these procedures also revealed some incomplete lexical entries in Alpino. For example, there are two past participles not listed as adjectives in the lexicon though they should be. Thus when our method *correctly* assigned them adjective types, it got punished since these types were not in the gold standard.

We see in Table 5 that the increase in precision for the verbs is small and recall remains practically unchanged. The unimproved recall shows that we have not gained much from the subcat frame heuristics. Even when the number of the observed sentences was increased, less frequent frames often remained unrecognisable from the noise in the parsed data. This could be seen as a proof that in the vast majority of cases verbs are used *transitively* and/or *intransitively*. Since the MLE method we employ proved to be good at recognising these two frames and differentiating between them, we have decided to continue using it.

The overall F-score improved by only 2% because the modified verb and adjective predictions are less than 30% of the total predictions made by the classifier.

5 Experiment with Real Unknown Words

To investigate whether the proposed LA method is also beneficial for the parser, we observe how parsing accuracy changes when the method is employed. Accuracy in Alpino is measured in terms of labelled dependencies.

We have conducted an experiment with a test set of 300 sentences which contain 188 real unknown words. The sentences have been randomly selected from the manually annotated LASSY corpus (van Noord, 2009) which contains text from various domains. The average sentence length is 26.54 tokens.

The results are given in Table 6. The standard Alpino model uses its guesser to assign types to the unknown words. Model 1 employs the trained ME-based classifier to predict lexical entries for the unknown words offline and then uses them during parsing. Model 2 uses lexical entries modified by applying the methods described in Section 4 to the output of the classifier (Model 1).

Model	Accuracy (%)	msec/sentence
Alpino	88.77	8658
Model 1	89.06	8772
Model 2	89.24	8906

Table 6: Results with real unknown words

Our LA system as a whole shows an error reduction rate of more than 4% with parse times remaining similar to those of the standard Alpino version. It should also be noted that though much of the unknown words are generally nouns, we see from the results that it makes sense to also employ the methods for improving the predictions for the other POS types. A wrong verb or even adjective prediction can cause much more damage to the analysis than a wrong noun one.

These results illustrate that the integration of our method in the parser can improve its performance on real-life data.

6 Discussion

6.1 Comparison to Previous Work

The performance of the LA method we presented in this paper can be compared to the performance

of a number of other approaches previously applied to the same task.

Baldwin (2005) uses a set of binary classifiers to learn lexical entries for a large-scale grammar of English (ERG; (Copestake and Flickinger, 2000)). The main disadvantage of the method is that it uses information obtained from secondary language resources– POS taggers, chunkers, etc. Therefore, the grammar takes no part in the LA process and the method acquires lexical entries based on incomplete linguistic information provided by the various resources. The highest F-measure (about 65%) is achieved by using features from a chunker but it is still 20% lower than the results we report here. Further, no evaluation is done on how the method affects the performance of the ERG when the grammar is used for parsing.

Zhang and Kordoni (2006) and Cholakov et al. (2008), on the other hand, include features from the grammar in a maximum entropy (ME) classifier to predict new lexical entries for the ERG and a large German grammar (GG; (Crysmann, 2003)), respectively. The development data for this method consist of linguistically annotated sentences from treebanks and the grammar features used in the classifier are derived from this annotation. However, when the method is applied to open-text unannotated data, the grammar features are replaced with POS tags. Therefore, the grammar is no longer directly involved in the LA process which affects the quality of the predictions. Evaluation on sentences containing real unknown words shows improvement of the coverage for the GG when LA is employed but the accuracy decreases by 2%. Such evaluation has not been done for the ERG. The results on the development data are not comparable with ours because evaluation is done only in terms of precision while we are also able to measure recall.

Statistical LA has previously been applied to Alpino as well (van de Cruys, 2006). However, his method employs less morphosyntactic features in comparison to our approach and does not make use of word paradigms. Further, though experiments on development data are performed on a smaller scale, the results in terms of F-measure are 10% lower than those reported in our case study. Experiments with real unknown words have not been performed.

Other, non-statistical LA methods also exist. Cussens and Pulman (2000) describe a symbolic approach which employs *inductive logic programming* and Barg and Walther (1998) and Fouvry (2003) follow a unification-based approach. However, the generated lexical entries might be both too general or too specific and it is doubtful if these methods can be used on a large scale. They have not been applied to broad-coverage grammars and no evaluation is provided.

6.2 Application to Other Systems and Languages

We stress the fact that the experiments with Alpino represent only a case study. The proposed LA method can be applied to other computational grammars and languages providing that the following conditions are fulfilled.

First, words have to be mapped onto some finite set of labels of which a subset of open-class (universal) labels has to be selected. This subset represents the labels which the ME-based classifier can predict for unknown words. Second, a (large) corpus has to be available, so that various sentences in which a given unknown word occurs can be extracted. This is crucial for obtaining different contexts in which this word is found.

Next, we need a parser to analyse the extracted sentences which allows for the syntactic constraints imposed by these contexts to be included in the prediction process.

Finally, as for the paradigm generation, the idea of combining a finite state morphology and web heuristics is general enough to be implemented for different languages. It is also important to note that the classifier allows for arbitrary combinations of features and therefore, a researcher is free to include any (language-specific) features he or she considers useful for performing LA.

We have already started investigating the applicability of our LA method to large-scale grammars of German and French and the initial experiments and results we have obtained are promising.

References

Baldwin, Tim. 2005. Bootstrapping deep lexical resources: Resources for courses. In *Proceedings of the ACL-SIGLEX 2005 Workshop on Deep Lexical Acquisition*, Ann Arbor, USA.

Barg, Petra and Markus Walther. 1998. Processing unknown words in HPSG. In *Proceedings of the 36th Conference of the ACL*, Montreal, Quebec, Canada.

Brent, Michael R. 1993. From grammar to lexicon: unsupervised learning of lexical syntax. *Computational Linguistics*, 19(2):243–262.

Briscoe, Ted and John Caroll. 1997. Automatic extraction of subcategorization from corpora. In *Proceedings of the 5th ACL Conference on Applied Natural Language Processing*, Washington, DC.

Cholakov, Kostadin and Gertjan van Noord. 2009. Combining finite state and corpus-based techniques for unknown word prediction. In *Proceedings of the 7th Recent Advances in Natural Language Processing (RANLP) conference*, Borovets, Bulgaria.

Cholakov, Kostadin, Valia Kordoni, and Yi Zhang. 2008. Towards domain-independent deep linguistic processing: Ensuring portability and re-usability of lexicalised grammars. In *Proceedings of COLING 2008 Workshop on Grammar Engineering Across Frameworks (GEAF08)*, Manchester, UK.

Copestake, Ann and Dan Flickinger. 2000. An open-source grammar development environment and broad-coverage English grammar using HPSG. In *Proceedings of the 2nd International Conference on Language Resource and Evaluation (LREC 2000)*, Athens, Greece.

Crysmann, Berthold. 2003. On the efficient implementation of German verb placement in HPSG. In *Proceedings of RANLP 2003*, Borovets, Bulgaria.

Cussens, James and Stephen Pulman. 2000. Incorporating linguistic constraints into inductive logic programming. In *Proceedings of the Fourth Conference on Computational Natural Language Learning*.

Fouvry, Frederik. 2003. Lexicon acquisition with a large-coverage unification-based grammar. In *Companion to the 10th Conference of EACL*, pages 87–90, Budapest, Hungary.

Kinyon, Alexandra and Carlos A Prolo. 2002. Identifying verb arguments and their syntactic function in the Penn Treebank. In *Proceedings of the 3rd International Conference on Language Resource and Evaluation (LREC 2002)*, Las Palmas de Gran Canaria, Spain.

Korhonen, Anna, Genevieve Gorell, and Diana McCarthy. 2000. Statistical filtering and subcategorization frame acquisition. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, Hong Kong, China.

Lapata, Mirella. 1999. Acquiring lexical generalizations from corpora. A case study for diathesis alternations. In *Proceedings of the 37th Annual Meeting of ACL*, Maryland, USA.

Malouf, Robert. 2002. A comparison of algorithms for maximum entropy parameter estimation. In *Proceedings of the 6th conference on Natural Language Learning (CoNLL-2002)*, pages 49–55, Taipei, Taiwan.

Manning, Christopher. 1993. Automatic acquisition of a large subcategorization dictionary from corpora. In *Proceedings of the 31st Annual Meeting of ACL*, Columbus, OH.

Messiant, Cedric. 2008. A subcategorization acquisition system for French verbs. In *Proceedings of the ACL 2008 Student Research Workshop*, Columbus, OH.

O'Donovan, Ruth, Michael Burke, Aoife Cahill, Josef van Genabith, and Andy Way. 2005. Large-scale induction and evaluation of lexical resources from the Penn-II and Penn-III Treebanks. *Computational Linguistics*, 31(3):329–365.

Prins, Robbert and Gertjan van Noord. 2001. Unsupervised POS-tagging improves parcing accuracy and parsing efficiency. In *Proceedings of IWPT*, Beijing, China.

van de Cruys, Tim. 2006. Automatically extending the lexicon for parsing. In Huitnik, Janneje and Sophia Katrenko, editors, *Proceedings of the Eleventh ESSLLI Student Session*, pages 180–189.

van Noord, Gertjan. 2006. At last parsing is now operational. In *Proceedings of TALN*, Leuven, Belgium.

van Noord, Gertjan. 2009. Huge parsed corpora in LASSY. In *Proceedings of the Seventh International Workshop on Treebanks and Linguistic Theories (TLT 7)*, Groningen, The Netherlands.

Zhang, Yi and Valia Kordoni. 2006. Automated deep lexical acquisition for robust open text processing. In *Proceedings of the Fifth International Conference on Language Resourses and Evaluation (LREC 2006)*, Genoa, Italy.

Global topology of word co-occurrence networks: Beyond the two-regime power-law

Monojit Choudhury
Microsoft Research Lab India
monojitc@microsoft.com

Diptesh Chatterjee
Indian Institute of Technology Kharagpur
diptesh.chh.1987@gmail.com

Animesh Mukherjee
Complex Systems Lagrange Lab, ISI Foundation
animesh.mukherjee@isi.it

Abstract

Word co-occurrence networks are one of the most common linguistic networks studied in the past and they are known to exhibit several interesting topological characteristics. In this article, we investigate the global topological properties of word co-occurrence networks and, in particular, present a detailed study of their spectrum. Our experiments reveal certain universal trends found across the networks for seven different languages from three different language families, which are neither reported nor explained by any of the previous studies and models of word-cooccurrence networks. We hypothesize that since word co-occurrences are governed by syntactic properties of a language, the network has much constrained topology than that predicted by the previously proposed growth model. A deeper empirical and theoretical investigation into the evolution of these networks further suggests that they have a core-periphery structure, where the core hardly evolves with time and new words are only attached to the periphery of the network. These properties are fundamental to the nature of word co-occurrence across languages.

1 Introduction

In a natural language, words interact among themselves in different ways – some words co-occur with certain words at a very high probability than other words. These co-occurrences are non-trivial, as in their patterns cannot be inferred from the frequency distribution of the individual words. Understanding the structure and the emergence of these patterns can present us with important clues and insights about how we evolved this extremely complex phenomenon, that is language.

In this paper, we present an in-depth study of the word co-occurrence patterns of a language in the framework of complex networks. The choice of this framework is strongly motivated by its success in explaining various properties of word co-occurrences previously (Ferrer-i-Cancho and Solé, 2001; Ferrer-i-Cancho et al, 2007; Kapustin and Jamsen, 2007). Local properties, such as the degree distribution and clustering coefficient of the word co-occurrence networks, have been thoroughly studied for a few languages (Ferrer-i-Cancho and Solé, 2001; Ferrer-i-Cancho et al, 2007; Kapustin and Jamsen, 2007) and many interesting conclusions have been drawn. For instance, it has been found that these networks are small-world in nature and are characterized by a *two regime power-law* degree distribution. Efforts have also been made to explain the emergence of such a two regime degree distribution through network growth models (Dorogovstev and Mendes, 2001). Although it is tempting to believe that a lot is known about word co-occurrences, in order to obtain a deeper insight into how these co-occurrence patterns emerged there are many other interesting properties that need to be investigated. One such property is the *spectrum* of the word co-

Coling 2008: Poster Volume, pages 162–170,
Beijing, August 2010

occurrence network which can provide important information about its global organization. In fact, the application of this powerful mathematical machinery to infer global patterns in linguistic networks is rarely found in the literature (few exceptions are (Belkin and Goldsmith, 2002; Mukherjee et al, 2009)). However, note that spectral analysis has been quite successfully applied in the analysis of biological and social networks (Banerjee and Jost, 2007; Farkas et al, 2001).

The aim of the present work is to investigate the spectral properties of a word co-occurrence network in order to understand its global structure. In particular, we study the properties of seven different languages namely Bangla (Indo-European family), English (Indo-European family), Estonian (Finno-Ugric family), French (Indo-European family), German (Indo-European family), Hindi (Indo-European family) and Tamil (Dravidian family). Quite importantly, as we shall see, the most popular growth model proposed by Dorogovtsev and Mendes (DM) (Dorogovstev and Mendes, 2001) for explaining the degree distribution of such a network is not adequate to reproduce the spectrum of the network. This observation holds for all the seven different languages under investigation. We shall further attempt to identify the precise (linguistic) reasons behind this difference in the spectrum of the empirical network and the one reproduced by the model. Finally, as an additional objective, we shall present a hitherto unreported deeper analysis of this popular model and show how its most important parameter is correlated to the size of the corpus from which the empirical network is constructed.

The rest of the paper is laid out as follows. In section 2, we shall present a brief review of the previous works on word co-occurrence networks. This is followed by a short primer to spectral analysis. In section 4, we outline the construction methodology of the word co-occurrence networks and present the experiments comparing the spectrum of these real networks with those generated by the DM model. Section 5 shows how the most important parameter of the DM model varies with the size of the corpus from which the co-occurrence networks are constructed. Finally, we conclude in section 6 by summarizing our contributions and pointing out some of the implications of the current work.

2 Word Co-occurrence Networks

In this section, we present a short review of the earlier works on word co-occurrence networks, where the nodes are the words and an edge between two words indicate that the words have co-occurred in a language in certain context(s). The most basic and well studied form of word co-occurrence networks are the *word collocation networks*, where two words are linked by an edge if they are neighbors (i.e., they collocate) in a sentence (Ferrer-i-Cancho and Solé, 2001).

In (Ferrer-i-Cancho and Solé, 2001), the authors study the properties of two types of collocation networks for English, namely the *unrestricted* and the *restricted* ones. While in the unrestricted network, all the collocation edges are preserved, in the restricted one only those edges are preserved for which the probability of occurrence of the edge is higher than the case when the two words collocate independently. They found that both the networks exhibit small-world properties; while the average path length between any two nodes in these networks is small (between 2 and 3), the clustering coefficients are high (0.69 for the unrestricted and 0.44 for the restricted networks). Nevertheless, the most striking observation about these networks is that the degree distributions follow a two regime power-law. The degree distribution of the 5000 most connected words (i.e., the kernel lexicon) follow a power-law with an exponent -3.07, which is very close to that predicted by the Barabási-Albert growth model (Barabási and Albert, 1999). These findings led the authors to argue that the word usage of the human languages is preferential in nature, where the frequency of a word defines the comprehensibility and production capability. Thus, higher the usage frequency of a word, higher is the probability that the speakers will be able to produce it easily and the listeners will comprehend it fast. This idea is closely related to the *recency effect* in linguistics (Akmajian, 1995).

Properties of word collocation networks have also been studied for languages other than English (Ferrer-i-Cancho et al, 2007; Kapustin and

Jamsen, 2007). The basic topological characteristics of all these networks (e.g., scale-free, small world, assortative) are similar across languages and thus, point to the fact that like Zipf's law, these are also linguistic universals whose emergence and existence call for a non-trivial psycholinguistic account.

In order to explain the two regime power-law in word collocation networks, Dorogovtsev and Mendes (Dorogovstev and Mendes, 2001) proposed a preferential attachment based growth model (henceforth referred to as the DM model). In this model, at every time step t, a new word (i.e., a node) enters the language (i.e., the network) and connects itself preferentially to one of the pre-existing nodes. Simultaneously, ct (where c is a positive constant and a parameter of the model) new edges are grown between pairs of old nodes that are chosen preferentially. Through mathematical analysis and simulations, the authors successfully establish that this model gives rise to a two regime power-law with exponents very close to those observed in (Ferrer-i-Cancho and Solé, 2001). In fact, for English, the values k_{cross} (i.e., the point where the two power law regimes intersect) and k_{cut} (i.e., the point where the degree distribution cuts the x-axis) obtained from the model are in perfect agreement with those observed for the empirical network.

Although the DM model is capable of explaining the local topological properties of the word collocation network, as we shall see in the forthcoming sections, it is unable to reproduce the global properties (e.g., the *spectrum*) of the network.

3 A Primer to Spectral Analysis

Spectral analysis[1] is a powerful mathematical method capable of revealing the global structural patterns underlying an enormous and complicated environment of interacting entities. Essentially, it refers to the systematic investigation of the eigenvalues and the eigenvectors of the adjacency matrix of the network of these interacting entities. In this section, we shall briefly outline the basic concepts involved in spectral analysis and discuss some of its applications (see (Chung, 1994) for details).

A network consisting of n nodes (labeled as 1 through n) can be represented by an $n \times n$ square matrix $\mathbf{A}$, where the entry a_{ij} represents the weight of the edge from node i to node j. Note that $\mathbf{A}$, which is known as the *adjacency matrix*, is symmetric for an undirected graph and have binary entries for an unweighted graph. λ is an eigenvalue of $\mathbf{A}$ if there is an n-dimensional vector $\mathbf{x}$ such that

$$\mathbf{Ax} = \lambda\mathbf{x}$$

Any real symmetric matrix $\mathbf{A}$ has n (possibly non-distinct) eigenvalues $\lambda_0 \leq \lambda_1 \leq \ldots \leq \lambda_{n-1}$, and corresponding n eigenvectors that are mutually orthogonal. The *spectrum* of a network is the set of the distinct eigenvalues of the graph and their corresponding multiplicities. It is a distribution usually represented in the form of a plot with the eigenvalues in x-axis and their multiplicities in the y-axis.

The spectrum of real and random networks display several interesting properties. Banerjee and Jost (Banerjee and Jost, 2007) report the spectrum of several biological networks and show that these are significantly different from the spectrum of artificially generated networks. It is worthwhile to mention here that spectral analysis is also closely related to *Principal Component Analysis* and *Multidimensional Scaling*. If the first few (say d) eigenvalues of a matrix are much higher than the rest of the eigenvalues, then one can conclude that the rows of the matrix can be approximately represented as linear combinations of d orthogonal vectors. This further implies that the corresponding graph has a few motifs (subgraphs) that are repeated a large number of time to obtain the global structure of the graph (Banerjee and Jost, 2009).

In the next section, we shall present a thorough study of the spectrum of the word co-occurrence networks across various languages.

4 Experiments and Results

For the purpose of our experiments, we construct word collocation networks for seven different languages namely, Bangla, English, Esto-

[1]The term spectral analysis is also used in the context of signal processing, where it refers to the study of the frequency spectrum of a signal.

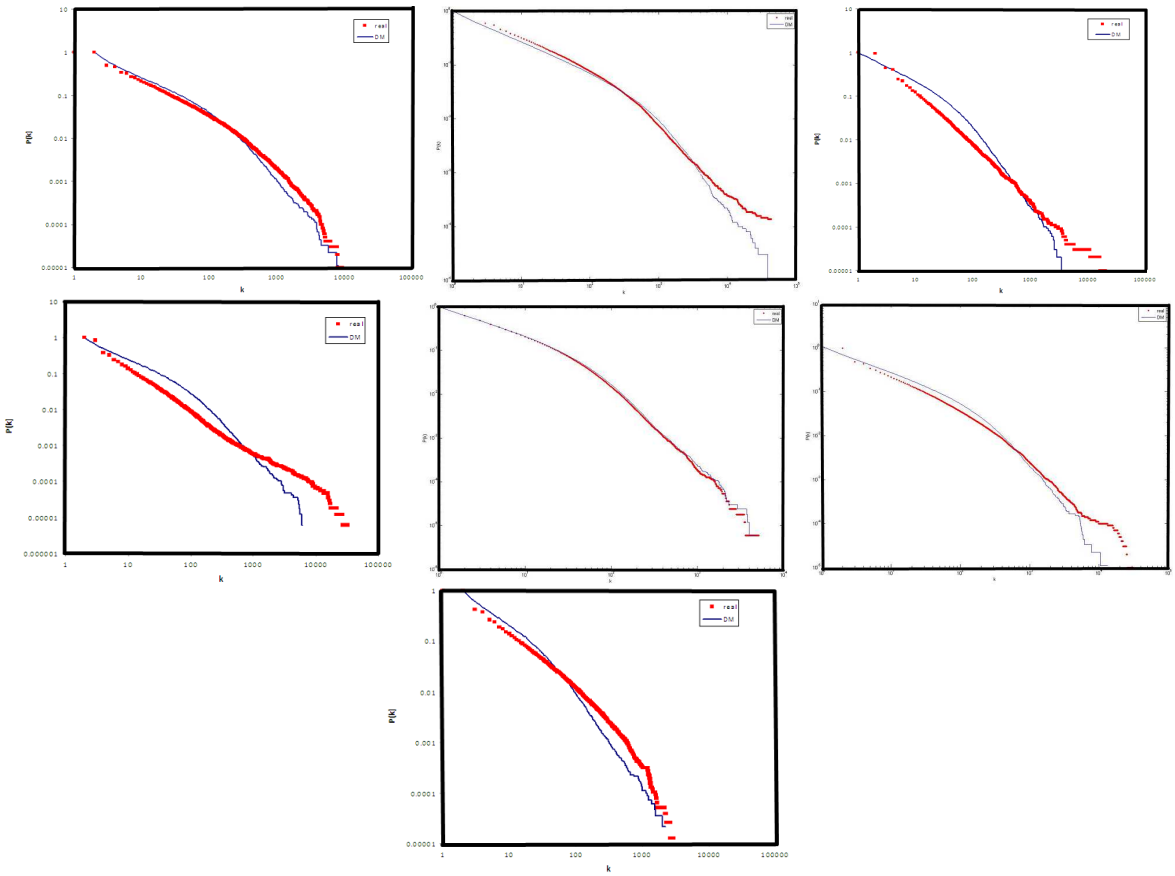


Figure 1: Cumulative degree distributions for Bangla, English, Estonian, French, German, Hindi and Tamil respectively. Each red line signifies the degree distribution for the empirical network while each blue line signifies the one obtained from the DM model.

Lang.	Tokens (Mill.)	Words	KLD	c	Max. Eig. (Real)	Max. Eig. (DM)
English	32.5	97144	0.21	5.0e-4	849.1	756.8
Hindi	20.2	99210	0.32	2.3e-4	472.5	329.5
Bangla	12.7	100000	0.29	2.0e-3	326.2	245.0
German	5.0	159842	0.19	6.3e-5	192.3	110.7
Estonian	4.0	100000	0.25	1.1e-4	158.6	124.0
Tamil	2.3	75929	0.24	9.9e-4	116.4	73.06
French	1.8	100006	0.44	8.0e-5	236.1	170.1

Table 1: Summary of results comparing the structural properties of the empirical networks for the seven languages and the corresponding best fits (in terms of KLD) obtained from the DM model.

nian, French, German, Hindi and Tamil. We used the corpora available in the *Lipezig Corpora Collection* (http://corpora.informatik.uni-leipzig.de/) for English, Estonian, French and German. The Hindi, Bangla and Tamil corpora were collected by crawling some online newspapers. In these networks, each distinct word corresponds to a vertex and two vertices are connected by an edge if the corresponding two words are adjacent in one or more sentences in the corpus. We assume the network to be undirected and unweighted (as in (Ferrer-i-Cancho and Solé, 2001)).

As a following step, we simulate the DM model and reproduce the degree distribution of the collocation networks for the seven languages. We vary the parameter c in order to minimize the KL

divergence (KLD) (Kullback and Leibler, 1951) between the empirical and the synthesized distributions and, thereby, obtain the best match. The results of these experiments are summarized through Figure 1 and Table 1. The results clearly show that the DM model is indeed capable of generating the degree distribution of the collocation networks to a very close approximation for certain values of the parameter c (see Table 1 for the values of c and the corresponding KLD).

Subsequently, for the purpose of spectral analysis, we construct subgraphs induced by the top 5000 nodes for each of the seven empirical networks as well as those generated by the DM model (i.e., those for which the degree distribution fits best in terms of KLD with the real data). We then compute and compare the spectrum of the real and the synthesized networks (see Figure 2 and Table 1). It is quite apparent from these results that the spectra of the empirical networks are significantly different from those obtained using the DM model. In general, the spectral plots indicate that the adjacency matrices for networks obtained from the DM model have a higher rank than those for the empirical networks. Further, in case of the synthesized networks, the first eigenvalue is significantly larger than the second whereas for the empirical networks the top 3 to 4 eigenvalues are found to dominate. Interestingly, this property is observed across all the languages under investigation.

We believe that the difference in the spectra is due to the fact that the ordering of the words in a sentence are strongly governed by the grammar or the syntax of the language. Words belong to a smaller set of lexico-syntactic categories, which are more commonly known as the parts-of-speech (POS). The co-occurrence patterns of the words are influenced, primarily, by its POS category. For instance, *nouns* are typically preceded by *articles* or *adjectives*, whereas *verbs* might be preceded by *auxiliary verbs*, *adverbs* or *nouns*, but never *articles* or *adjectives*. Therefore, the words "car" and "camera" are more likely to be structurally similar in the word co-occurrence network, than "car" and "jumped". In general, the local neighborhoods of the words belonging to a particular POS is expected to be very similar, which means that several rows in the adjacency matrix will be very similar to each other. Thus, the matrix is expected to have low rank.

In fact, this property is not only applicable to syntax, but also semantics. For instance, even though adjectives are typically followed by nouns, semantic constraints make certain adjective-noun co-occurrences (e.g., "green leaves") much more likely than some others (e.g., "green dreams" or "happy leaves"). These notions are at the core of latent semantics and vector space models of semantics (see, for instance, Turney and Pantel (Turney and Pantel, 2010) for a recent study). The DM model, on the other hand, is based on the *recency effect* that says that the words which are produced most recently are easier to remember and therefore, easier to produce in the future. Preferential attachment models the recency effect in word production, which perhaps is sufficient to replicate the degree distribution of the networks. However, the model fails to explain the global properties, precisely because it does not take into account the constraints that govern the distribution of the words.

It is quite well known that the spectrum of a network can be usually obtained by iteratively powering the adjacency matrix of the network (aka power iteration method). Note that if the adjacency matrices of the empirical and the synthesized networks are powered even once (i.e., they are squared)[2], their degree distributions match no longer (see Figure 3). This result further corroborates that although the degree distribution of a word co-occurrence network is quite appropriately reproduced by the DM model, more global structural properties remain unexplained. We believe that word association in human languages is not arbitrary and therefore, a model which accounts for the clustering of words around their POS categories might possibly turn out to present a more accurate explanation of the spectral properties of the co-occurrence networks.

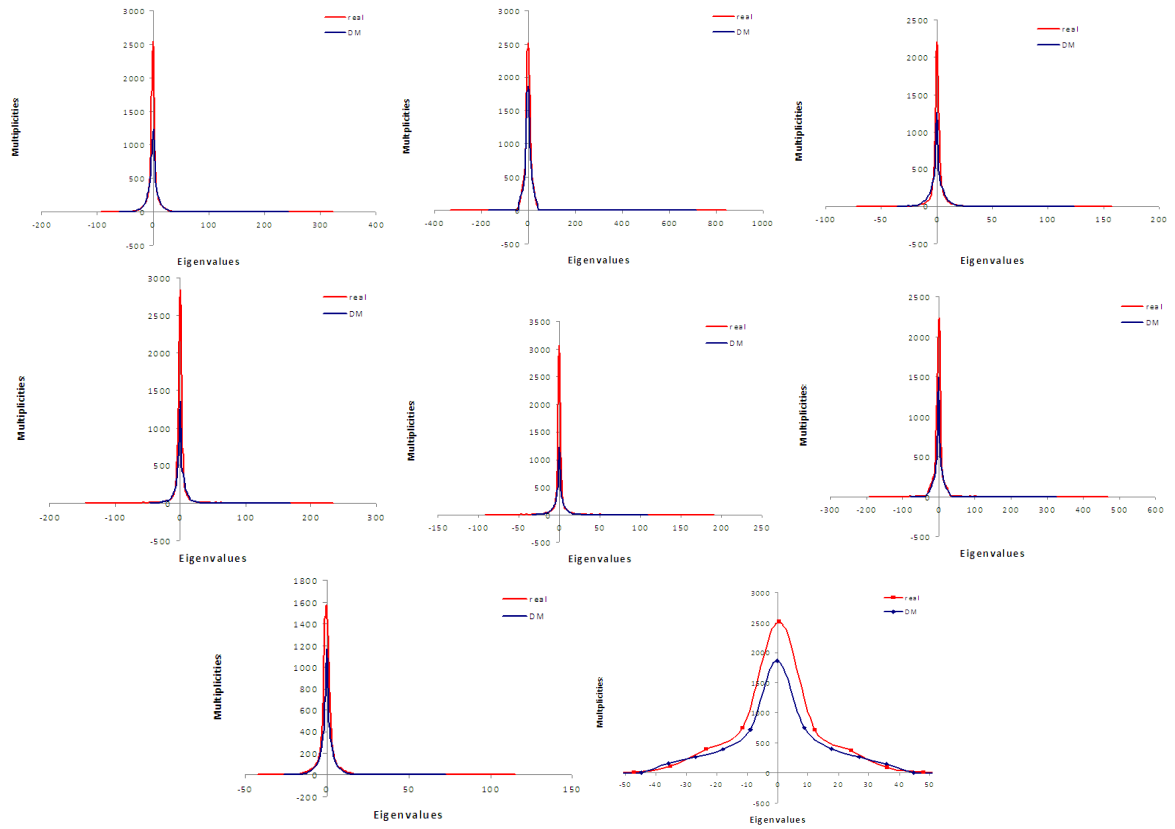


Figure 2: The spectrum for Bangla, English, Estonian, French, German, Hindi and Tamil respectively. The last plot shows a portion of the spectrum for English magnified around 0 for better visualization. All the curves are binned distributions with bin size = 100. The blue line in each case is the spectrum for the network obtained from the DM model while each red line corresponds to the spectrum for the empirical network.

5 Reinvestigating the DM Model

In this section, we shall delve deeper into exploring the properties of the DM model since it is one of the most popular and well accepted models for explaining the emergence of word associations in a language. In particular, we shall investigate the influence of the model parameter c on the emergent results.

If we plot the value of the parameter c (from Table 1) versus the size of the corpora (from Table 1) used to construct the empirical networks for the different languages we find that the two are highly correlated (see Figure 4).

[2]Note that this squared network is weighted in nature. We threshold all edges below the weight 0.07 so that the resultant network is neither too dense nor too sparse. The value of the threshold is chosen based on the inspection of the data.

In order to further check the dependence of c on the corpus size we perform the following experiment. We draw samples of varying corpus size and construct empirical networks from each of them. We then simulate the DM model and attempt to reproduce the degree distribution for each of these empirical networks. In each case, we note the value c for which the KLD between the empirical and the corresponding synthesized network is minimum. Figure 5 shows the result of the above experiment for English. The figure clearly indicates that as the corpus size increases the value of the parameter c decreases. Similar trends are observed for all the other languages.

In general, one can mathematically prove that the parameter c is equal to the rate of change of the average degree of the network with respect to

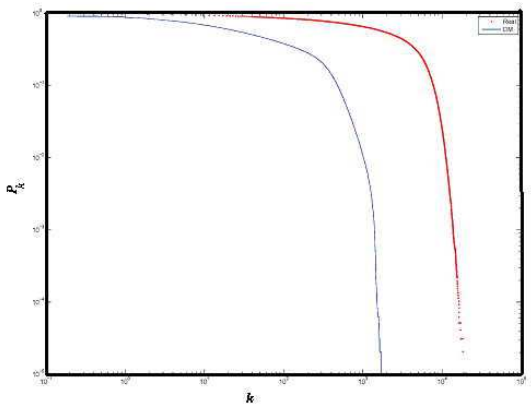


Figure 3: Cumulative degree distribution for the squared version of the networks for English. The red line is the degree distribution for the squared version of the empirical network while the blue line is degree distribution of the squared version of the network obtained from the DM model. The trends are similar for all the other languages.

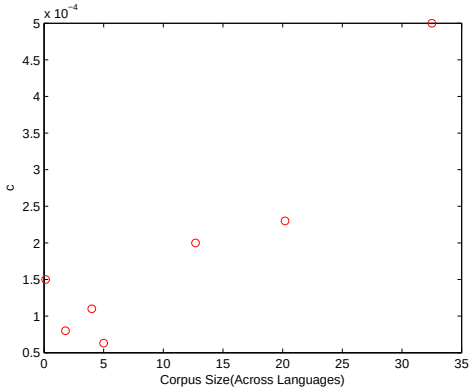


Figure 4: The parameter c versus the corpus size for the seven languages.

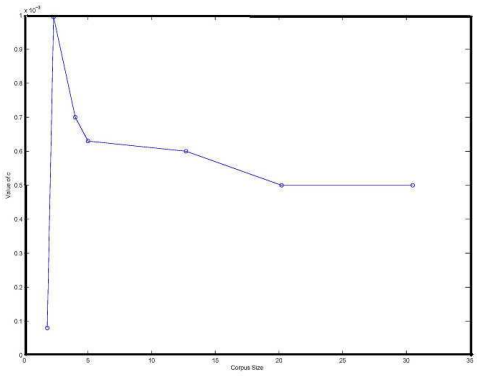


Figure 5: The parameter c versus the corpus size for English.

the time t. The proof is as follows.

At every time step t, the number of new edges formed is (1+ct). Since each edge contributes to a total degree of 2 to the network, the sum of the degrees of all the nodes in the network (k_{tot}) is

$$k_{tot} = 2\sum_{t=1}^{T}(1+ct) = 2T + cT(T+1) \quad (1)$$

At every time step, only one new node is added to the network and therefore the total number of nodes at the end of time T is exactly equal to T. Thus the average degree of the network is

$$\langle k \rangle = \frac{2T + cT(T+1)}{T} = 2 + c(T+1) \quad (2)$$

The rate of change of average degree is

$$\frac{d\langle k \rangle}{dT} = c \quad (3)$$

and this completes the proof.

In fact, it is also possible to make a precise empirical estimate of the value of the parameter c. One can express the average degree of the co-occurrence networks as the ratio of twice the bigram frequency (i.e., twice the number of edges in the network) to the unigram frequency (i.e., the number of nodes or unique words in the network). Therefore, if we can estimate this ratio we can easily estimate the value of c using equation 3. Let us denote the total number of distinct bigrams and unigrams after processing a corpus of size N by $B(N)$ and $W(N)$ respectively. Hence we have

$$\langle k \rangle = \frac{2B(N)}{W(N)} \quad (4)$$

Further, the number of distinct new unigrams after

Language	$B(N)$	$W(N)$	c
English	$29.2N^{.67}$	$59.3N^{.43}$	$.009N^{-.20}$
Hindi	$26.2N^{.66}$	$49.7N^{.46}$	$.009N^{-.26}$
Tamil	$1.9N^{.91}$	$6.4N^{.71}$	$.207N^{-.50}$

Table 2: Summary of expressions for $B(N)$, $W(N)$ and c for English, Hindi and Tamil.

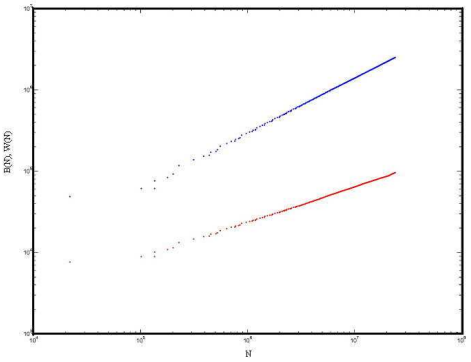


Figure 6: Variation of $B(N)$ and $W(N)$ with N for English (in doubly-logarithmic scale). The blue dots correspond to variation of $B(N)$ while the red dots correspond to the variation of $W(N)$.

processing a corpus of size N is equivalent to T and therefore

$$T = W(N) \tag{5}$$

Sampling experiments across different languages demonstrate that $W(N)$ and $B(N)$ are of the form ηN^{α} ($\alpha < 1$) where η and α are constants. For instance, Figure 6 shows in doubly-logarithmic scale how $B(N)$ and $W(N)$ varies with N for English. The R^2 values obtained as a result of fitting the $B(N)$ versus N and the $W(N)$ versus N plots using equations of the form ηN^{α} for English, Hindi and Tamil are greater than 0.99. This reflects the high accuracy of the fits. Similar trends are observed for all the other languages.

Finally, using equations 3, 4 and 5 we have

$$c = \frac{d\langle k \rangle}{dT} = \frac{d\langle k \rangle}{dN}\frac{dN}{dT} \tag{6}$$

and plugging the values of $B(N)$ and $W(N)$ in equation 6 we find that c has the form $\kappa N^{-\beta}$ ($\beta < 1$) where κ and β are language dependent positive constants. The values of c obtained in this way for three different languages English, Hindi and Tamil are noted in Table 5.

Thus, we find that as $N \rightarrow \infty$, $c \rightarrow 0$. In other words, as the corpus size grows the number of distinct new bigrams goes on decreasing and ultimately reaches (almost) zero for a very large sized corpus. Now, if one plugs in the values of c and T obtained above in the expressions for k_{cross} and k_{cut} in (Dorogovstev and Mendes, 2001), one observes that $\lim_{N\rightarrow\infty} \frac{k_{cross}}{k_{cut}} = 0$. This implies that as the corpus size becomes very large, the two-regime power law (almost) converges to a single regime with an exponent equal to -3 as is exhibited by the Barabási-Albert model (Barabási and Albert, 1999). Therefore, it is reasonable to conclude that although the DM model provides a good explanation of the degree distribution of a word co-occurrence network built from a medium sized corpora, it does not perform well for very small or very large sized corpora.

6 Conclusions

In this paper, we have tried to investigate in detail the co-occurrence properties of words in a language. Some of our important observations are: (a) while the DM model is able to reproduce the degree distributions of the word co-occurrence networks, it is not quite appropriate for explaining the spectrum of these networks; (b) the parameter c in the DM model signifies the rate of change of the average degree of the network with respect to time; and (c) the DM model does not perform well in explaining the degree distribution of a word co-occurrence network when the corpus size is very large.

It is worthwhile to mention here that our analysis of the DM model leads us to a very important observation. As N grows, the value of k_{cut} grows at a much faster rate than the value of k_{cross} and in the limit $N \rightarrow \infty$ the value of k_{cut} is so high as compared to k_{cross} that the ratio $\frac{k_{cross}}{k_{cut}}$ becomes (almost) zero. In other words, the kernel lexicon, formed of the words in the first regime of the two regime power-law and required to "say everything or almost everything" (Ferrer-i-Cancho and Solé, 2001) in a language, grows quite slowly as new words creep into the language. In contrast, the peripheral lexicon making the other part of the two regime grows very fast as new words enter the language. Consequently, it may be argued that since the kernel lexicon remains almost unaffected, the effort to learn and retain a language by its speakers increases only negligibly as new words creep into the language.

References

A. Akmajian. *Linguistics: An introduction to Language and Communication*. MIT Press, Cambridge, MA, 1995.

A. Banerjee and J. Jost. Spectral plots and the representation and interpretation of biological data. *Theory in Biosciences*, 126(1), 15-21, 2007.

A. Banerjee and J. Jost. Graph spectra as a systematic tool in computational biology. *Discrete Applied Mathematics*, 157(10), 2425–2431, 2009.

A.-L. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 286, 509-512, 1999.

M. Belkin and J. Goldsmith. Using eigenvectors of the bigram graph to infer morpheme identity. In *Proceedings of Morphological and Phonological Learning*, Association for Computational Linguistics, 41-47, 2002.

F. R. K. Chung. *Spectral Graph Theory*. Number 2 in CBMS Regional Conference Series in Mathematics, American Mathematical Society, 1994.

S. N. Dorogovstev and J. F .F. Mendes. Language as an evolving word Web. *Proceedings of the Royal Society of London B*, 268, 2603-2606, 2001.

I. J. Farkas, I. Derényi, A. -L. Barabási and T. Vicsek. Spectra of "real-world" graphs: Beyond the semicircle law, *Physical Review E*, 64, 026704, 2001.

R. Ferrer-i-Cancho and R. V. Solé. The small-world of human language. *Proceedings of the Royal Society of London B*, 268, 2261–2266, 2001.

R. Ferrer-i-Cancho, A. Mehler, O. Pustylnikov and A. Díaz-Guilera. Correlations in the organization of large-scale syntactic dependency networks. In *Proceedings of TextGraphs-2: Graph-Based Algorithms for Natural Language Processing*, 65-72, Association for Computational Linguistics, 2007.

V. Kapustin and A. Jamsen. Vertex degree distribution for the graph of word co-occurrences in Russian. In *Proceedings of TextGraphs-2: Graph-Based Algorithms for Natural Language Processing*, 89-92, Association for Computational Linguistics, 2007.

S. Kullback and R. A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics* 22(1), 79-86, 1951.

A. Mukerjee, M. Choudhury and R. Kannan. Discovering global patterns in linguistic networks through spectral analysis: A case study of the consonant inventories. In *Proceedings of EACL*, 585–593, Association for Computational Linguistics, 2009.

P. D. Turney and P. Pantel. From frequency to meaning: Vector space models of semantics. In *JAIR*, 37, 141-188, 2010.

Exploiting Paraphrases and Deferred Sense Commitment to Interpret Questions more Reliably

Peter Clark and Phil Harrison
Boeing Research & Technology
{peter.e.clark,philip.harrison}@boeing.com

Abstract

Creating correct, semantic representations of questions is essential for applications that can use formal reasoning to answer them. However, even within a restricted domain, it is hard to anticipate all the possible ways that a question might be phrased, and engineer reliable processing modules to produce a correct semantic interpretation for the reasoner. In our work on posing questions to a biology knowledge base, we address this brittleness in two ways: First, we exploit the DIRT paraphrase database to introduce alternative phrasings of a question; Second, we defer word sense and semantic role commitment until question answering. Resulting ambiguities are then resolved by interleaving additional interpretation with question-answering, allowing the combinatorics of alternatives to be controlled and domain knowledge to guide paraphrase and sense selection. Our evaluation suggests that the resulting system is able to understand exam-style questions more reliably.

1 Introduction

Our goal is to allow users to pose exam-style questions to a biology knowledge base (KB), containing formal representations of biological structures and processes expressed in first-order logic. As the questions typically require automated reasoning to answer them, a semantic interpretation of each question is needed. In our earlier work (Clark et al, 2007), questions were interpreted using a conventional pipeline (parse, coreference, sense and role disambiguation). However, despite moderate performance, the original ("base") system suffered from well-known problems of brittleness, arising from both premature commitments in the pipeline and the system's limited knowledge of the multiple ways that questions can be expressed. In this paper, we describe how deferred commitment and a large paraphrase database can be used to reduce these problems, drawing on prior work and applying it in the context of a large KB being available. In particular, by interleaving interpretation and answering, we are able to control the combinatorics of alternatives that would otherwise arise. An evaluation suggests that this improves the ability of the system to correctly interpret, and hence answer, questions.

2 Context and Related Work

Our system aims to interpret and answer high-school level, exam-style biology questions, expressed in sentence form. Our source of answers is a formal knowledge-base and reasoning engine (rather than a text corpus), placing specific requirements on the interpretation process - in particular, a full semantic interpretation of the question is required. Questions are typically one or two sentences long, for example:

(1) Does a prokaryotic cell contain ribosomes?
(2) A eukaryotic cell has a nucleus. Does that nucleus contain RRNA?
(3) Is adenine found in RNA molecules?
(4) Does a prokaryotic cell have a region consisting of cytosol?
(5) Do ribosomes synthesize proteins in the cytoplasm?
(6) What is the material, containing DNA and protein, that forms into chromosomes during mitosis?

Coling 2008: Poster Volume, pages 171–179,
Beijing, August 2010

Interpreting and answering this style of question has a long history in NLP, both for answers found via database retrieval and formal reasoning, and for answers extracted from a large text corpus.

For answers found using reasoning, the focus of this paper, early NL systems typically used a pipelined architecture for question interpretation (e.g., Bobrow, 1964; Woods 1977), with later systems also using semantic constraints to guide disambiguation decisions (e.g., Novak, 1977). More recently, as well as there being significant improvements in the performance of typical pipeline modules, e.g., word sense disambiguation (Navigli, 2009), there has been substantial work on various forms of deferred commitment, underspecification, and paraphrasing to expand the space of interpretations considered, and thus improve interpretation. Underspecified representations (e.g., van Deemter and Peters, 1996; Pinkal, 1999) allow ambiguity (in particular scope ambiguity) to be preserved in a single structure and commitments deferred until later, allowing multiple interpretations to be carried through the system. Similarly, a system can defer commitment by simply carrying multiple, alternative interpretations forward as individual structures, or packed together into a single structure (e.g., Alshawi and van Eijck, 1989, Bobrow et al., 2005; Kim et al., 2010a,b). Finally, canonicalized representations are often used to represent (and hence carry through the system) multiple, equivalent surface forms as a single structure, e.g., normalizing active and passive forms, or alternative forms of noun modification (Rinaldi et al., 2003). All these techniques help avoid premature commitment in interpretation.

As well as avoiding early rejection of interpretations in these ways, there has been substantial, recent work on expanding the space of possible interpretations considered through the use of paraphases (e.g., Sekine and Inui, 2007). Paraphrasing is based on the observation that there are many ways of saying (roughly) the same thing, and that syntatic manipulation alone is not sufficient to enumerate them all. Paraphrases aim to enumerate these additional alternatives, and may be generated synthetically (e.g., Rinaldi et al., 2003), drawn from similar texts (e.g., from similar questions for QA, Harabagiu et al., 2000), or mined from a corpus using machine learning techniques (e.g., Lin and Pantel, 2001). They have proved to be particularly useful in the context of textual entailment (e.g., Bentivogli et al., 2009), and in corpus-based question answering (e.g., Harabagiu et al., 2003).

Our work builds on this prior work, applying and extending these ideas to the context where a formal knowledge base and reasoning engine is available. In particular, we interleave the process of expanding the space of interpretations considered (using paraphrases and deferred commitment) with the process of question answering (which narrows down that space by selecting interpretations supported by the KB), thus controlling the otherwise combinatorial explosion of alternatives. This makes it feasible to use the DIRT paraphrase database (12 million paraphrases) for generating a full semantic interpretation of the original question, extending its previous use in the semi-formal context of textual entailment (Bentivogli et al., 2009). Our use of reasoning to guide disambiguation follows Hobbs et als (1993) method of "interpretation as abduction", where the system searches a space of possible interpretations for one(s) that are provable from the KB, preferring those interpretations.

3 The Problem

Although the biology KB we are using contains the knowledge to answer the six earlier questions (1)-(6), only the first two are correctly answered with the original pipelined ("base") system. For question (3):

(3) Is adenine found in RNA molecules?

the system (mis-)interprets this as referring to some actual "finding" event, not recognizing that this is an alternative way of phrasing a question about physical structure. Similarly, the notion of "consisting of" in question (4) is an unexpected phrasing that the system does not understand. Questions (5) and (6) are also answered incorrectly by the base system due to errors in semantic role labeling during interpretation. In (5):

(5) Do ribosomes synthesize proteins in the cytoplasm?

"in" is (mis-)interpreted by the language interpreter as an is-inside(x,y) relation, while the KB itself represents this relationship as site(x,y), hence the system fails to produce the correct answer (yes). Similarly, for (6) "into" is (mis)interpreted as destination(x,y) but represented in the KB as result(x,y).

Clearly, one can tweak the original interpreter to overcome these particular problems. However, it is a slow, expensive process, and in general it is impossible to anticipate all such problems up front. Statistical methods (e.g., Manning and Schutze, 1999) offer an alternative approach but one that is similarly noisy, problematic for question-answering applications.

4 Solution Approach

The brittleness of the base system can be partially attributed to its eager commitments, ahead of specifics that might be discovered during question-answering itself. To address this, we have modified the system in two ways. First, we have added use of paraphrases to explore additional interpretations of the question during question-answering. Second, we defer sense and semantic role disambiguation until question answering. As a result, part of interpretation occurs during answering itself: multiple interpretations are tried and a commitment is made to the one(s) that produce a non-null answer. The justification for this commitment is a **benevolent user** assumption, namely that the interpretation that "makes sense" with respect to the KB (i.e., produces a non-null answer) is the one that the user intended.

This use of question-answering to drive disambiguation follows Hobbs et als. (1993) work on Interpretation as Abduction. In that framework, a system searches for an interpretation that is provable from the KB plus a minimal cost set of assumptions, the interpretation corresponding to a particular way to disambiguate the text. In our work we do a similar thing, although restrict the assumptions to disambiguation decisions and exclude assuming new knowledge, as we are dealing with questions rather than assertions (if no interpretations are provable, then we treat the answer as "no" rather than treating the unproven query as something that should be asserted as true).

4.1 Paraphrases

Several paraphrase databases are now available to the NLP community[1], typically built by automatically finding phrases that occur in distributionally similar contexts (e.g., Dras et al, 2005). To date, paraphrase databases have primarily been exploited for recognizing textual entailment (e.g., Bentivogli et al., 2009, Clark et al, 2009), and for corpus-based question answering (e.g., Harabagiu et al., 2003). Here we use them for generating a full semantic interpretation in the context of querying a formal knowledge resource.

We use the DIRT paraphrase database (Lin and Pantel, 2001), containing approximately 12 million automatically learned rules of the form:

IF X *relation* Y THEN X *relation'* Y

where *relation* is a path in the dependency tree between constitutents X and Y, or equivalently (as we use later) a *chain of literals*:

$$\{p_0(x_0,x_1), w_1(x_1), \ldots p_{n-1}(x_{n-1},x_n)\}$$

where p_i is the syntactic relation between (non-prepositional) constituents x_i and x_{i+1}, and w_i is the word used for x_i. An example from DIRT is:

IF X is found in Y THEN X is inside Y

The condition "X is found in Y" can be expressed as the chain of literals:

{ object-of(x,f), "find"(f), "in"(f,y) }

The database itself is noisy, containing both good and nonsensical paraphrases. Interestingly, their use in question-answering tends to filter out most bad paraphrases, as it is rare that a nonsensical paraphrases will by chance produce an answer (i.e., the question + KB together help "triangulate" on good paraphrases). Nevertheless, bad paraphrases can sometimes produce incorrect answers. To handle this in a practical setting, we are adding an interactive interface (outside the scope of this paper) that shows the user any paraphrases used, and allows him/her to verify/block them as desired.

4.2 Deferred Sense Commitment

A second, common cause of failure of the base system was incorrect assignment of senses and

[1] e.g., http://www.aclweb.org/aclwiki/index.php?title=RTE_Knowledge_Resources

semantic relations during word sense disambiguation (WSD) and semantic role labeling (SRL). While domain-specific terms are generally reliably disambiguated, disambiguation of general terms (e.g., whether "split" denotes the concept of Separate or Divide) and semantic roles (e.g., whether "into" denotes destination(x,y) or result(x,y)) is less reliable, with only limited improvement attainable through manual engineering or machine learning. The problem is compounded by a degree of subjectivity in the way knowledge is encoded in the KB, for example whether the KB engineer chose to conceptualize a biological object as the "agent" or "instrument" or "site" of an activity is to a degree a matter of viewpoint.

To overcome this, we defer WSD and SRL commitments until question-answering itself. One can view this as a trivial form of preserving underspecification (eg. Pinkal, 1999) in the initial language processing, where the words themselves denote their possible meanings.

4.3 Algorithm and Implementation

Questions are first parsed using a broad coverage, phrase structure parser, followed by coreference resolution, producing an initial "syntactic" logical form, for example:

Question: *Do mitotic spindles consist of hollow microtubules?*

Logical Form (LF): "mitotic-spindle"(s), "consist"(c), "hollow"(h), "microtubule"(m), subject(c,s), "of"(c,m), modifier(m,h).

Next, rather than attempting word sense disambiguation (WSD) and semantic role labeling (SRL) as would be done in the base system, the system immediately starts work on answering the question, even though a complete semantic interpretation has not yet been produced. In the process of answering, the system explores alternative word senses, semantic roles, and paraphrases for the particular literals it is working on (described shortly), and if any are provable from the knowledge in the knowledge base then those branch(es) of the search are explored further. There are two basic steps in this process:

(a) **setup:** create an instance X0 of the object being universally quantified over[2] (identified during initial language interpretation)

(b) **query:** for each literal in the LF with at least one bound variable, iteratively query the KB to see if some interpretation of those literals are provable i.e., already known.

In this example, illustrated in Figure 1, for step (a) the system first creates an instance X0 of a mitotic spindle, i.e., asserts the instantiated first literal isa(X0,Mitotic-Spindle), and then queries the inference engine with the remaining LF literals. (If there are multiple senses for "mitotic spindle", then an instance for each sense is created, to be explored in parallel). For step (b), the system uses the algorithm as follows:

repeat
  select a chain C_u of "syntactic" literals in the LF with at least 1 bound variable
    C_u = {p(x,y)} or {w(x)} or
      {p_1(x,z), w(z), p_2(z,y)}
  select some interpretation C of C_u where:
    C is a possible interpretation of C_u
    or C'_u is a possible paraphrase for C_u and
      C is a possible interpretation of C'_u
  try prove C[bindings] → new-bindings
  If success:
    replace C_u with C
    add new-bindings to bindings
until
  all clauses are proved

where:

- A *syntactic literal* is a literal whose predicate is a word or syntactic role (subject, object, modifier, etc.) All literals in the initial LF are syntactic literals.
- A *chain of literals* is a set of syntactic literals in the LF of the form {p(x,y)} or {w(x)} or {p_1(x,z), w(z), p_2(z,y)}, where p_i, w are words or syntactic roles (subject, mod, etc).
- A *possible paraphrase* is a possible substitution of one chain of literals with another, listed in the DIRT paraphrase database.

[2] If the system can prove the answer for a (new) instance X0 of the universally quantified class, then it holds for all instances, i.e., if KB ∪ f(X0) ├ g(X0) then KB ├ f(X0)→g(X0), hence KB ├ ∀x f(x)→g(x) via the principle of universal generalization (UG).

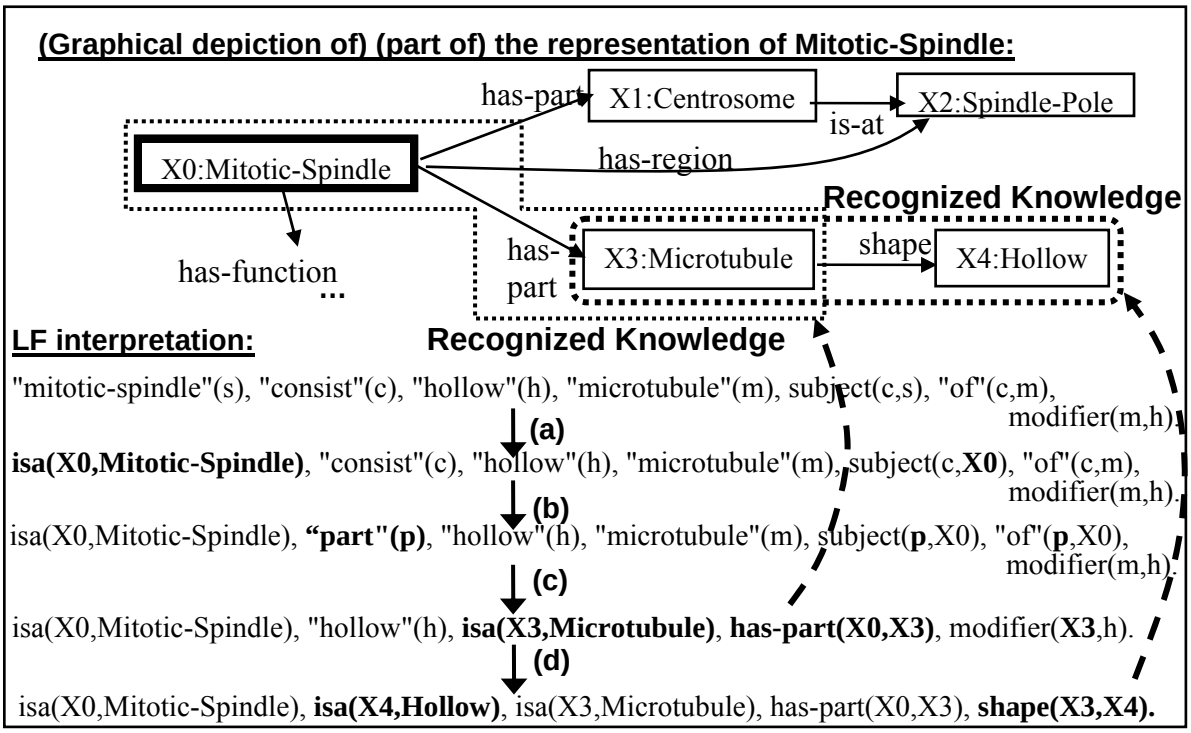


Figure 1: The path found through the search space for an interpretation of the example question. (a) setup (b) paraphrase substitution (IF X consists of Y THEN Y is part of X) (c) interpretation of {subject-of(X0,p),"part"(p), "of"(p,X0)} as has-part(X0,m), preferred as it is provable from the KB, resulting in m=X3 (d) interpretation of the syntactic modifier(X3,h) relation (from "hollow microtubule") as shape(X3,h) as it is provable from the KB.

- A *possible interpretation* of the singleton chain of literals {w(x)} is isa(x,*class*), where *class* is a possible sense of word w.
- A *possible interpretation* of a chain of literals {p(x,y)} or {p1(x,z),w(z),p2(z,y)} is r(x,y), where r is a semantic relation corresponding to syntactic relation p (e.g., "in"(x,y) → is-inside(x,y)) or word w (e.g., {subject-of(e,h), "have"(h), "of"(h,n)} → has-part(e,n)).

Possible word-to-class and word-to-predicate mappings are specified in the KB.

Figure 1 illustrates this procedure for the example sentence. The procedure iteratively replaces syntactic literals with semantic literals that correspond to an interpretation that is provable from the KB. If all the literals are proved, then the answer is "yes", as there exists an interpretation under which it can be proved from the KB, under the benevolent user assumption that this is the interpretation that the user intended.

As there are several points of non-determinism in the algorithm, e.g., which literals to select, which interpretation to explore, it is a search process. Our current implementation uses most-instantiated-first query ordering plus breadth-first search, although other implementations could traverse the space in other ways.

5 Evaluation

To evaluate the system, we measured its question-answering performance on a set of 141 true/false biology questions, ablating paraphrases and deferred commitment to measure their impact. The 141 questions were sentencized versions of the multiple choice options in 22 original AP-level exam questions that, in an earlier evaluation (Clark, 2009), users had difficulty rephrasing into a form that the system understood. Each original multiple choice option was minimally rewritten as a complete sentence (most multiple choice questions were partial se-

Configuration	Accuracy (score = y/y+n/n)	system/actual answers			
		y/y	n/y	y/n	n/n
Naive(all false)	67% (94)	0	47	0	94
Base system	72% (102)	8	41	0	94
+ Paraphrases	75% (106)	13	34	1	93
+ Deferred commitment	76% (107)	13	34	0	94
+ Both (full system)	84% (118)	25	22	1	93

Table 1: Performance of different configurations of the system. The y/y column shows the number of questions for which the system answered "yes" and the correct answer is "yes", etc.

ntences), while preserving the original English phrasing. For example the original question:

73. Which of the following best describes the DNA molecule?
   a. Two parallel strands of nitrogen bases held together by hydrogen bonding
   b. Two complementary strands of deoxyribose and phosphates held together by hydrogen bonding
   c. Two antiparallel strands of nucleotides held together by hydrogen bonding
   d. A single strand of nitrogen bases coiled upon itself by hydrogen bonding
   e. A single strand of nucleotides coiled into a helix.

was rewritten as five questions:

- Does a DNA molecule have two parallel strands of nitrogen bases held together by hydrogen bonding?
- Does a DNA molecule have two complementary strands of deoxyribose and phosphates held together by hydrogen bonding?
- Does a DNA molecule have two antiparallel strands of nucleotides held together by hydrogen bonding?
- Does a DNA molecule have a single strand of nitrogen bases coiled upon itself by hydrogen bonding?
- Does a DNA molecule have a single strand of nucleotides coiled into a helix?

Similarly:

79. All of the following organelles are associated with protein synthesis EXCEPT:
   a. ribosomes; b. Golgi bodies;...; e...

was rewritten as five questions:

- Are ribosomes associated with protein synthesis?
- Are Golgi bodies associated with...*etc*.

For 18 of the original questions, each of the 5 options expanded to 1 true/false question. For 3 comparison questions ("Which X is in Y but not Z?"), each option expanded into 2 questions ("Is X in Y?" "Is X in Z?"). Finally 1 question involved parallelism ("Which of the following A,B,C do X,Y,Z respectively?") which expanded into 21 questions ("Does A do X?" "Does A do Y?" etc.) after removing duplicates. Of the resulting 141 questions, 47 had the "gold" answer of true, 94 false. Of the 47 positives, 4 were out of scope of the reasoning engine, involving questions about possibility rather than truth, for example:

- Can a DNA adenine bond to an RNA uracil?

Another 3 were out of scope of the knowledge in the KB (2 requiring unrepresented temporal knowledge and 1 requiring commonsense knowledge). Thus the upper bound on performance, given the particular KB and reasoning engine that we are using, is 134/141 (95%).

We ran the base system alone, with paraphrasing (only), with deferred commitment (only), and with both. The results are shown in Table 1. As can be seen, true negatives (n/y) are a substantially larger challenge than false positives (y/n), as the system answers "no" by defalt if it is unable to prove the facts in the interpreted question from the KB. During interpretation, the base "pipeline" system commits to disambiguation decisions at each step, and if any commitment is wrong then it will also get the answer wrong, as reflected

by the only small (8) increase in number correctly answered.

Paraphrases allow the system to search for alternative interpretations, adding five more questions to be answered correctly but also introducing one false positive (y/n). The false positive was for the question:

Do peroxisomes make proteins?

This was (incorrectly) answered "yes" by the system as it used a bad DIRT paraphrase (IF X makes Y THEN X is made from Y), selected because it led to a provable interpretation (peroxisomes are made (synthesized) from proteins), but not the one the author intended. It is an interesting and perhaps somewhat surprising result that this was the only false positive, given that the DIRT database is noisy (approximately half its paraphrases are questionable or invalid). The low number of false positives appears to be due to the fact that the vast number of invalid paraphrases produce nonsensical, hence unprovable and rejected, interpretations.

Similarly, deferred commitment (alone) allowed five additional questions (different to those for paraphrasing) to be answered, again as premature word sense and semantic role labeling was avoided. For example, for "...the polymerase builds a strand...", the pipeline prematurely commits to the strand being the object of the build, while in the KB it is represented as the result of the build. Deferred commitment allows the system to search and find such alternatives.

Finally there were several (7) questions requiring both paraphrases and deferred commitment to answer. For example, "Do mitochondria provide cellular energy?" was answered using both a paraphrase (IF X provides Y THEN X creates Y) and deferred commitment (mitochondria was correctly interpreted as the site of the creation, as represented in the KB, while the pipeline prematurely committed to agent).

Although deferring SRL and WSD commitment, the final system still eagerly commits to a single syntactic analysis, and in some cases that analysis was wrong (e.g., wrong PP attachment), causing failure for some of the 16 in-scope, positive examples that the final system failed to answer. Clearly deferred commitment can be further extended to explore alternative syntactic analyses. The remaining failures were due to incorrect semantic interpretation of the syntactic analysis, primarily due to poor handling of coordination.

The median, average, and maximum cpu times per question were 0.7, 4.9, and 20.3 seconds respectively.

6 Discussion and Conclusion

Although question interpretation is challenging, we are in the unusual position of having substantial, formal domain (biology) knowledge available. We have illustrated how this knowledge can be exploited to improve question understanding by interleaving interpretation and answering together, allowing the DIRT paraphrase database to be feasibly used and avoiding premature sense commitment. The result is an improved understanding of the original biology questions.

Our work extends previous work (Section 2) on exploring multiple interpretations and exploiting paraphrases, doing so in the context of a task involving formal reasoning. In particular, by interleaving the expansion of possible interpretations with reasoning (that contracts those alternatives), a viable system can be constructed in which the combinatorics are controlled. However, although the system defers WSD and SRL commitment, there are other sources of brittleness – in particular its commitment to a single semantic analysis – that could also benefit from exploration of alternatives, e.g., by using packed representations (Bobrow et al., 2005).

A second limitation of the current approach is that it assumes the (semantics of the) question is a generalized subset of information in (or inferrable from) the KB, i.e., questions are "pure queries" about the KB that do not posit any new information. However some questions, in particular hypotheticals ("X is true. Does Y follow?"), violate this "pure query" assumption by asserting a novel premise (X) that is not in the KB, and hence cannot be disambiguated by searching for the premise X. Although such questions are relatively rare

in biology, they are common in other sciences (e.g., physics). Handling such questions would require extension of this approach, eg by matching a generalized form of the assertion X against the KB to identify how to disambiguate it. Similarly, if we wished to use the system to read new knowledge, as opposed to identify old knowledge, further extensions would be needed, as new knowledge by definition cannot be proved from the KB.

Finally, this work suggests that paraphrase databases such as DIRT offer potential for language understanding in the context of posing formal questions to a reasoning system or database, by bridging gaps that would otherwise have to be hand-engineered, extending their previous use in semi-formal settings such as textual entailment (Bentivogli et al., 2009). Despite noise, the question plus KB help "triangulate" on good paraphrases, and with a suitable user interface to expose their use, this work suggests that there is substantial potential for deploying them in a practical, end-user environment.

Acknowledgements

We are grateful to Vulcan Inc., who funded this work as part of Project Halo.

References

Alshawi H., van Eijck, J. 1989. Logical Forms in the Core Language Engine. Proc ACL, pp25-32.

Bentivogli, L., Dagan, I., Dang, Hoa, Giampiccolo, D., Magnini, B. 2009. The Fifth PASCAL Recognizing Textual Entailment Challenge. In *Proc Text Analysis Conference (TAC'09).*

Bobrow, D. 1964. A Question-Answering System for High School Algebra Word Problems. *AFIPS conference proceedings*, 16: 591-614.

Bobrow, D. G., Condoravdi, Crouch, R. Kaplan, R. Karttunen, L., King, L.T.H., de Paiva, V., Zaenen, A. 2005. A Basic Logic for Textual Inference. In *Proceedings of the AAAI Workshop on Inference for Textual Question Answering*, Pittsburgh, PA.

Chierchia, G. 1993. Questions with Quantifiers. In *Natural Language Semantics* 1, 181-234.

Clark, P. 2009. *A Study of Some "Hard to Formulate" Biology Questions*. Working Note 33, Boeing Technical Report.

Clark, P., Chaw, J., Chaudhri, V., Harrison, P. 2007. Capturing and Answering Questions Posed to a Knowledge-Based System. In *Proc. KCap 2007.*

Clark, P. Harrison, P. 2009. An inference-based approach to textual entailment. In Proc TAC 2009 (Text Analysis conference).

Curtis, J., Matthews, G., Baxter, D. 2005. On the Effective Use of Cyc in a Question-Answering System. *Proc Workshop on Knowledge and Reasoning for Answering Questions*, IJCAI'05, pp 61-70.

Dras, M., Yamamoto, K. (Eds). 2005. *Proc 3rd Internationl Workshop of Paraphrasing*. South Korea.

Harabagiu, S., Moldovan, D., Pasca, M., Mihalcea, R.,et al., 2000. FALCON: Boosting Knowledge for Answer Engines. *Proc TREC'2000 (9th Text Retrieval Conf)*, pp 479-488.

Hobbs, J. Stickel, M., Appelt, D., Martin, P. 1993. Interpretation as Abduction. *Artificial Intelligence* 63 (1-2), pp 69-142.

Kim, D., Barker, K., Porter, B. 2010a. Building an End-to-End Text Reading System based on a Packed Representation. *Proc NAACL-HLT Workshop on Machine Reading*.

Kim, D., Barker, K., Porter, B. 2010b. Improving the Quality of Text Understanding by Delaying Ambiguity Resolution. *Proc COLING 2010*.

Lin, D. and Pantel, P. 2001. Discovery of Inference Rules for Question Answering. *Natural Language Engineering* 7 (4) pp 343-360.

Manning, C., Schutze, H. 1999. *Foundations of Statistical Natural Language Processing.* MA: MIT Press.

Navigli. R. 2009. Word Sense Disambiguation: a Survey. *ACM Computing Surveys,* 41(2), ACM Press, pp. 1-69

Novak, G. 1977. Representations of Knowledge in a Program for Solving Physics Problems, *IJCAI'77*, pp. 286-291

Pinkal, M. 1999. On Semantic Underspecification. In Bunt, H./Muskens, R. (Eds.). *Proceedings of the 2nd International Workshop on Compuational Linguistics (IWCS 2).*

Rinaldi, F., Dowall, J. et al., 2003. Exploiting Paraphrases in a Question Answering System. In *Proc 2003 ACL Workshop on Paraphrasing (IWP 2003).*

Sekine, S., Inui, K. 2007. *Proc ACL-PASCAL Workshop on Textual Entailment and Paraphrasing.*

van Deemter, K., Peters, S. 1996. *Semantic Ambiguity and Underspecification*. CA: CSLI.

Woods, W. 1977. Lunar rocks in natural English: Explorations in natural language question answering. *Fundamental Studies in Computer Science.* A. Zampolli, Ed. North Holland, 521-569.

Two Methods for Extending Hierarchical Rules from the Bilingual Chart Parsing

Martin Čmejrek and **Bowen Zhou**
IBM T. J. Watson Research Center
{martin.cmejrek, zhou}@us.ibm.com

Abstract

This paper studies two methods for training hierarchical MT rules independently of word alignments. Bilingual chart parsing and EM algorithm are used to train bitext correspondences. The first method, rule arithmetic, constructs new rules as combinations of existing and reliable rules used in the bilingual chart, significantly improving the translation accuracy on the German-English and Farsi-English translation task. The second method is proposed to construct additional rules directly from the chart using inside and outside probabilities to determine the span of the rule and its non-terminals. The paper also presents evidence that the rule arithmetic can recover from alignment errors, and that it can learn rules that are difficult to learn from bilingual alignments.

1 Introduction

Hierarchical phrase-based systems for machine translation usually share the same pattern for obtaining rules: using heuristic approaches to extract phrase and rule pairs from word alignments. Although these approaches are very successful in handling local linguistic phenomena, handling longer distance reorderings can be more difficult. To avoid the combinatorial explosion, various restrictions, such as limitations of the phrase length or non-terminal span are used, that sometimes prevent from extracting good rules. Another reason is the deterministic nature of those heuristics that does not easily recover from errors in the word alignment.

In this work, we learn rules for hierarchical phrase based MT systems directly from the parallel data, independently of bilingual word alignments.

Let us have an example of a German-English sentence pair from the Europarl corpus (Koehn, 2005).

(1) GER: die herausforderung besteht darin diese systeme zu den besten der welt zu machen
ENG: the challenge is to make the system the very best

The two pairs of corresponding sequences *diese systeme ... der welt—the system ... best* and *zu machen—to make* are swapped. We believe that the following rule could handle long distance reorderings, still with a reasonably low number of terminals, for example:

(2) $X \rightarrow \langle$besteht darin X_1 zu X_2, is to $X_2 X_1 \rangle$,

There are 127 sentence pairs out of 300K of the training data that contain this pattern, but this rule was not learned using the conventional approach (Chiang, 2007). There are three potential risks: (1) alignment errors (the first *zu* aligned to *to*, or *der welt* (*of the world*) aligned to null); (2) maximum phrase length for extracting rules lower than 11 words; (3) requirement of non-terminals spanning at least 2 words.

The *rule arithmetic* (Cmejrek et al., 2009) constructs the new rule (2) as a combination of good rule usages:

(3) $X \rightarrow \langle$besteht darin, is $\rangle$
$X \rightarrow \langle X_1$ zu X_2, to $X_2 X_1 \rangle$

Coling 2008: Poster Volume, pages 180–188,
Beijing, August 2010

The approach consists of bilingual chart parsing (BCP) of the training data, combining rules found in the chart using a *rule arithmetic* to propose new rules, and using EM to estimate rule probabilities.

In this paper, we study the behavior of the rule arithmetic on two different language pairs: German-English and Farsi-English. We also propose an additional method for constructing new rules directly from the bilingual chart, and compare it with the rule arithmetic.

The paper is structured as follows: In Sec. 1, we explain our main motivation, summarize previous work, and briefly introduce the formalism of hierarchical phrase-based translation. In Sec. 2, we describe the bilingual chart parsing and the EM algorithm. The rule arithmetic is introduced in Sec. 3. The new method for proposing new rules directly from the chart is described in Sec. 4. The experimental setup is described in Sec. 5. Results are thoroughly discussed in Sec. 6. Finally, we conclude in Sec. 7.

1.1 Related work

Many previous works use the EM algorithm to estimate probabilities of translation rules: Wu (1997) uses EM to directly estimate joint word alignment probabilities of Inversion Transduction Grammar (ITG). Marcu and Wong (2002) use EM to estimate joint phrasal translation model (JPTM). Birch et al. (2006) reduce its complexity by using only concepts that match the high-confidence GIZA++ alignments. Similarly, Cherry and Lin (2007) use ITG for pruning. May and Knight (2007) use EM algorithm to train tree-to-string rule probabilities, and use the Viterbi derivations to re-align the training data. Huang and Zhou (2009) use EM to estimate conditional rule probabilities $P(\alpha|\gamma)$ and $P(\gamma|\alpha)$ for Synchronous Context-free Grammar. Others try to overcome the deterministic nature of using bilingual alignments for rule extraction by sampling techniques (Blunsom et al., 2009; DeNero et al., 2008). Galley et al. (2006) define minimal rules for tree-to-string translation, merge them into composed rules (similarly to the rule arithmetic), and train weights by EM. While in their method, word alignments are used to define all rules, rule arithmetic proposes new rules independently of word alignments. Similarly, Liu and Gildea (2009) identify matching long sequences ("big templates") using word alignments and "liberate" matching small subtrees based on chart probabilities. Our method of proposing rules directly from the chart does not use word alignment at all.

1.2 Formally syntax-based models

Our baseline model follows the Chiang's hierarchical model (Chiang, 2007; Chiang, 2005; Zhou et al., 2008) based on Synchronous Context-free Grammar (SCFG). The rules have form

$$X \rightarrow \langle \gamma, \alpha, \sim \rangle, \tag{4}$$

where X is the only non-terminal in the grammar, γ and α are source and target strings with terminals and up to two non-terminals, $\sim$ is the correspondence between the non-terminals. Corresponding non-terminals have to be expanded at the same time.

2 Bilingual chart parsing and EM algorithm

In this section, we briefly overview the algorithm for bilingual chart parsing and EM estimation of SCFG rule features.

Let $\mathbf{e} = e_1^M$ and $\mathbf{f} = f_1^N$ of source and target sentences. For each sentence pair $\mathbf{e}, \mathbf{f}$, the 'E' step of the EM algorithm will use the bilingual chart parser to enumerate all possible derivations Φ, compute inside probabilities $\beta_{ijkl}(X)$ and outside probabilities $\alpha_{ijkl}(X)$, and finally calculate expected counts $c(r)$ how many times each rule r produced the corpus C.

The inside probabilities can be defined recursively and computed dynamically during the chart parsing:

$$\beta_{ijkl} = \sum_{\rho \in t_{ijkl}} P(\rho.r) \prod_{(i'j'k'l') \in \rho.bp} \beta_{i'j'k'l'}, \tag{5}$$

where t_{ijkl} represents the chart cell spanning (e_i^j, f_k^l), and the data structure ρ stores the rule $\rho.r$. If r has non-terminals, then $\rho.bp$ stores backpointers $\rho.bp_1$ and $\rho.bp_2$ to the cells representing their derivations.

The outside probabilities can be computed recursively by iterating the chart in top-down ordering. We start from the root cell $\alpha_{1,M,1,N} := 1$ and propagate the probability mass as

$$\alpha_{\rho.bp_1} += P(\rho.r)\alpha_{ijkl} \quad (6)$$

for rules with one non-terminal, and

$$\alpha_{\rho.bp_1} += P(\rho.r)\alpha_{ijkl}\beta_{\rho.bp_2}, \quad (7)$$
$$\alpha_{\rho.bp_2} += P(\rho.r)\alpha_{ijkl}\beta_{\rho.bp_1}, \quad (8)$$

for rules with two non-terminals. The top-down ordering ensures that each α_{ijkl} accumulates updates from all cells higher in the chart before its own outside probability is used.

The contributions to the rule expected counts are computed as

$$c(\rho.r) += \frac{P(\rho.r)\alpha_{ijkl}\prod_{i=1}^{\rho.n}\beta_{\rho.bp_i}}{\beta_{1,M,1,N}}. \quad (9)$$

Finally, rule probabilities $P(r)$ are obtained by normalizing expected counts in the 'M' step.

To improve the grammar coverage, the ruleset is extended by the following rules providing "backoff" parses and scoring for the SCFG rules:

(10) $\langle X_1, X_1 f\rangle$, $\langle X_1, f X_1\rangle$, $\langle X_1 e, X_1\rangle$, $\langle e X_1, X_1\rangle$,

(11) $\langle X_1 X_2, X_2 X_1\rangle$.

Rules (10) enable insertions and deletions, while rule (11) allows for aligning swapped constituents in addition to the standard glue rule.

3 Proposing new rules with rule arithmetic

The main idea of this work is to propose new rules independently of the bilingual word alignments. We parse each sentence pair using the baseline ruleset extended by the new rule types (10) and (11). Then we select the *most promising* rule usages and combine each two of them using the *rule arithmetic* to propose new rules. We put the new rules into a temporary pool, and parse and compute probabilities and expected counts again, this time we use rules from the baseline and from the temporary pool. Finally, we dump expected counts for proposed rules, and empty the temporary pool. This way we can try to propose many rules for each sentence pair, and to filter them later using accumulated expected counts from the EM.

The term *most promising* is purposefully vague — to cover all possible approaches to filtering rule usages. In our implementation, we are limited by space and time, and we have to prune the number of rules that we can combine. We use expected counts as the main scoring criterion. When computing the contributions to expected counts from particular rule usages as described by (9), we remember the n-best contributors, and use them as candidates after the expected counts for the given sentence pair have been estimated.

The *rule arithmetic* combines existing rules using *addition* operation to create new rules. The idea is shown in Example 12.

(12) Addition

$\langle 5,13,5,11,13,13\rangle$ $\langle 4,10,6,10,5,5\rangle$ $X \to \langle X_1$ zu X_2, to X_2 $X_1\rangle$
$\langle 5,11,6,11,0,0\rangle$ $\langle 6,10,7,10,0,0\rangle$ $X \to \langle$diese X_1, the $X_1\rangle$

1:	...	4	5	6	...	11	12	13	3	4	5	6	7	...	10
2:	...	0	-1	-1	...	-1	zu	-2	0	to	-2	-1	-1	...	-1
3:	...	0	diese	-3	...	-3	0	0	0	0	0	the	-3	...	-3
4:	...	0	diese	-3	...	-3	zu	-2	0	to	-2	the	-3	...	-3

5: $\langle 5,13,6,11,13,13\rangle$ $\langle 4,10,7,10,5,5\rangle$ $X \to \langle$diese X_1 zu X_2, to X_2 the $X_1\rangle$

First, create span projections for both source and target sides of both rules. Use symbol 0 for all unspanned positions, copy terminal symbols as they are, and use symbols -1, -2, -3, and -4 to transcribe X_1 and X_2 from the first rule, and X_1 and X_2 from the second rule. Repeat the non-terminal symbol on all spanned positions. In Example 12 line 1 shows the positions in the sentence, lines 2 and 3 show the rule span projections of the two rules.

Second, merge source span projections (line 4), record mappings of non-terminal symbols. We require that merged projections are *continuous*. We allow substituting non-terminal symbols by terminals, but we require that the whole span of the non-terminal is fully replaced. In other words, shortenings of non-terminal spans are not allowed.

Third, collect new rule. The merged rule usages (lines 5) are generalized into rules, so that they are not limited to the particular span for which they were originally proposed.

The rule arithmetic can combine all types of rules – phrase pairs, abstract rules, glues, swaps, insertions and deletions. However, we require that

at least one of the rules is either a phrase pair or an abstract rule.

4 Proposing directly from chart

One of the issues observed while proposing new rules with the rule arithmetic is the selection of the best candidates. The number of all candidates that can be combined depends on the length of the sentence pair and on the number of competing parsing hypotheses. Using a fixed size of the n-best can constitute a risk of selecting bad candidates from shorter sentences. On the other hand, the spans of the best candidates extracted from long sentences can be far from each other, so that most combinations are not valid rules (e.g., the combination of two discontinuous phrasal rules is not defined).

In our new approach we propose new rules directly from the bilingual chart, relying on the inside and outside probabilities computed after the parsing of the sentence pair. The method has two steps. In the first step we identify best matching parallel sequences; in the second step we propose "holes" for non-terminals.

4.1 Identifying best matching sequences

To identify the best matching sequences, we score all sequences (e_i^j, f_k^l) by a scoring function:

$$score_{ijkl} = \frac{\alpha_{ijkl}\beta_{ijkl}}{\beta_{1,M,1,N}} Lex(i,j,k,l), \quad (13)$$

where the lexical score is defined as:

$$Lex(i,j,k,l) = \sum_{j'=1}^{N} \prod_{i'=0}^{M} t(f_{j'}|e_{i'})\delta_{ijkli'j'} \quad (14)$$

The t is the lexical probability from the word-to-word translation table, and $\delta_{ijkli'j'}$ is defined as δ_{ins} if $i' \in \langle i,j \rangle$ and $j' \in \langle k,l \rangle$, and as δ_{out} if $i' \notin \langle i,j \rangle$ and $j' \notin \langle k,l \rangle$, and as 0 elsewhere. The purpose of this function is to score only the pairs of words that are both either from within the sequence or from outside the sequence. Usually $0 \leq \delta_{out} \leq \delta_{ins}$ to put more weight on words within the parallel sequence.

The scoring function is a combination of expected counts contribution of a sequence (e_i^j, f_k^l) estimated from the chart with the IBM Model 1 lexical score.

Since only the sequences spanned by filled chart cells can have non-zero expected counts, we can select the n-best matching sequences relatively efficiently.

4.2 Proposing non-terminal positions

Similar approach can be used to propose best positions for non-terminals. We score every combination of non-terminal positions. The expected counts can be estimated using Eq. 9. Since we are proposing new rules, the probability $P(r)$ used in that equation is not defined. Again, we can use Model 1 score instead, and use the following scoring function:

$$s_{ijkl}(bp_1, bp_2) = \frac{Lex(i,j,k,l,bp_1,bp_2)\alpha_{ijkl}\beta_{bp_1}\beta_{bp_2}}{\beta_{1,M,1,N}}, \quad (15)$$

$Lex(i,j,k,l,bp_1,bp_2)$ is defined as in Eq. 14. This time using $0 \leq \delta_{out} \leq \delta_{NT1} = \delta_{NT2} \leq \delta_{term}$, restricting the IBM Model 1 to score only word pairs that both belong either to the terminals of the proposed rule, or to the sequences spanned by the same non-terminal, or outside of the rule span. The scoring function for rules with one non-terminal is just a special case of 15.

Again, the candidates can be scored efficiently, taking into account only those combinations of non-terminal spans that correspond to filled cells in the chart.

The proposed method is again independent of bilingual alignment, but at the same time utilizes the information obtained from the bilingual chart parsing.

5 Experiments

We carried out experiments on two language pairs, German-English and Farsi-English.

The **German-English** data is a subset (297k sentence pairs) of the Europarl (Koehn, 2005) corpus. Since we are focused on speech-to-speech translation, the punctuation was removed, and the text was lowercased. The dev set and test set contain each 1k sentence pairs with one reference.

The word alignments were trained by GIZA++ toolkit (Och and Ney, 2000). Phrase pairs were

extracted using grow-diag-final (Koehn et al., 2007). The baseline ruleset was obtained as in (Chiang, 2007). The maximum phrase length for rule extraction was set to 10, the minimum required non-terminal span was 2.

Additional rules for insertion, deletion, and swap were added to improve the parsability of the data, and to help EM training and rule arithmetic. However, these rules are not used by the decoder, since they would degrade the performance.

New rules were proposed after the first iteration of EM[1], either by rule arithmetic or directly from the chart.

Only non-terminal rules proposed by the rule arithmetic from at least two different sentence pairs and ranked (by expected counts $c(r)$) in the top 100k were used. Figure 4 presents a sample of the new rules.

New rules were also proposed directly from the chart, using the approach in Sec. 4. 5% of best matching parallel sequences, and 5 best scoring rules were selected from each parallel sequence. Non-terminal rules from the 200k-best rank were added to the model. Figure 5 presents a sample of the new rules.

Finally, one more iteration of EM was used to adjust the probabilities of the new and baseline rules. These probabilities were used as features in the decoding.

The performance of rule arithmetic was also verified on **Farsi-English** translation. The training corpus contains conversational spoken data from the DARPA TransTac program extended by movie subtitles and online dictionaries downloaded from the web (297k sentence pairs). The punctuation was removed, and the text was lowercased. The dev set is 1,420 sentence pairs held out from the training data, with one reference. The test set provided by NIST contains 470 sentences with 4 references. The sentences are about 30% longer and more difficult.

The training pipeline was the same as for the German-English experiments. 122k new non-terminal rules were proposed using the rule arithmetic.

[1]Since our initial experiments did not show any significant gain from proposing rules after additional (lengthy) iterations of EM.

The feature weights were tuned on the dev set for each translation model separately. The translation quality was measured automatically by BLEU score (Papineni et al., 2001).

6 Discussion of results

The BLEU score results are shown in the Table 3. The cumulative gain of rule arithmetic and EM (RA + EM-i0) is 1 BLEU point for German-English translation and 2 BLEU points for Farsi-English. The cumulative gain of rules proposed from the chart (DC + EM-i0) is 0.2 BLEU points for German-English. For comparison of effects of various components of our method, we also show scores after the first five iterations of EM (EM-i0–EM-i4) without adding any new rules, just using EM-trained probabilities as feature weights, and also scores for new rules added into the baseline without adjusting their costs by EM (RA).

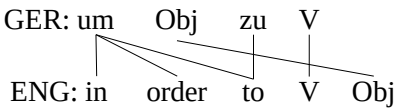
The qualities of proposed rules are discussed in this section.

6.1 German-English rules from rule arithmetic

The Figure 4 presents a sample of new rules proposed during this experiment. The table is divided into three parts, presenting rules from the top, middle, and bottom of the 100K list. The quality of the rules is high even in the middle part of the table, the tail part is worse.

We were surprised by seeing short rules consisting of frequent words. For example $\langle$um X_1, in order $X_1\rangle$. When looking into word-level alignments, we realized that these rules following the pattern 16 prevent the baseline approach from extracting the rule.

(16)

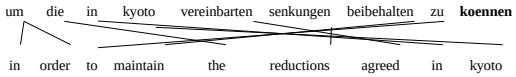


Similarly many other rules match the pattern of beginning of a subordinated clause, such as *that is why*, or insertions, such as *of course*, which both have to be strictly followed by VSO construction in German, in contrast to the SVO word order in English.

We also studied the cases of rule arithmetic correcting for systematic word alignment errors. For

example the new rule $\langle X_1$ zu koennen, to $X_1\rangle$ was learned from the sentence

(17)



The English translation often uses a different modality, thus the modal verb *koennen* is always aligned with null. Since unaligned words are usually not allowed at the edges of sub-phrases generalized into non-terminals (Chiang, 2007), this rule cannot be learned by the baseline.

We observe that many new proposed rules correspond to patterns with a non-terminal spanning one word. For example $\langle$um X_1 zu X_2, to X_2 $X_1\rangle$ corresponds to the same pattern 16, where X_2 spans one verb. The line *baseline min1* in the Table 3 shows 0.3 BLEU improvement of a model trained without the minimum non-terminal span requirement. However, this improvement comes at a cost of more than four times increased model size, as shown in Table 2. We observe that using the minimum span requirement while learning from bitext alignments combined with rule arithmetic that can learn the most reliable rules spanning one word yields better performance in speed, memory, and precision.

We can also study the new rules quantitatively. We want to know how the rules proposed by the rule arithmetic are used in decoding. We traced the translation of the 1,000 test set sentences to mark the rules that were used to generate the best scoring hypotheses.

The stats are presented in the Table 1. The chance that a new rules will be used in the test set decoding (0.86%) is more than 7 times higher than that of all rules (0.12%). Encouraging evidence is that while the rule arithmetic rules constitute only 1.87% of total rules, they present 9.17% of rules used in the decoding.

The Figure 1 lists the most frequently used new rules in the decoding. We can see many rules with 2 non-terminals that model complex verb forms ($\langle$wird X_1 haben,will have $X_1\rangle$), reordering in clauses ($\langle$um X_1 zu gewaehrleisten, to ensure $X_1\rangle$), or reordering of verbs from the second position in German to SVO in English ($\langle$heute X_1 wir X_2, today we X_1 $X_2\rangle$).

	RA Ger.	DC Ger.	RA Farsi
Sentences translated	1,000	1,000	417
|ALL| (all rules)	5.359,751	5.459,751	8.532,691
|NEW| (new rules)	100,000	200,000	121,784
$\frac{\text{\|NEW\|}}{\text{\|ALL\|}}$	1.87%	3.66%	1.43%
|hits ALL|	10,122	7,256	2,521
|glue|	2,910	271	267
|hits ALL unique|	6.303	6,433	2,058
$\frac{\text{\|hits ALL unique\|}}{\text{\|ALL\|}}$	0.12%	0.12%	0.02
|hits NEW|	928	1,541	125
|hits NEW unique|	858	1,504	110
$\frac{\text{\|hits NEW unique\|}}{\text{\|NEW\|}}$	0.86%	0.75 %	0.09
$\frac{\text{\|hits NEW\|}}{\text{\|hits ALL\|}}$	9.17%	21.23%	4.96%
|terminals from NEW|	4,385	7,825	407
$\frac{\text{\|terminals from NEW\|}}{\text{\|hits NEW\|}}$	4,73	5.08	3.26

Table 1: Rule hits for 1,000 test set.

Model	#phrases	#rules
Ger-Eng baseline	8.5M	5.3M
Ger-Eng baseline min1	8.5M	23.M

Table 2: Model sizes.

We also studied the correlation between the rank of the proposed rules (ranked by expected counts) and the hit rate during the decoding. The Figure 2 measures the hit rate for each of 1,000 best ranking rules, and should be read as follows: the rules ranking 0 to 999 were used 70 times, the hit rate decreases as the rank grows so that there were no hits for rules ranking 90k and more. The rank is a good indicator of the usefulness of new rules.

We hypothesize that the new rules are capable of combining partial solutions to form hypotheses with better word order, or better complex verb forms so that these hypotheses are better scored and are parts of the winning solutions more often.

6.2 German-English rules proposed directly from the chart

We also studied why the rules proposed directly from the bilingual chart yield smaller improvement than the rule arithmetic. The number of new rules used in the decoding (1,541) is even higher than that of the rule arithmetic, and it constitutes 21.23% of all cases. The two experiments were

#hits	Ger	Eng
5	X_1 stellt X_2 dar	X_1 is X_2
3	X_1 sowohl X_2 als auch	X_1 both X_2 and
3	X_1 ist es X_2	it is X_2 X_1
3	X_1 die X_2 ist	X_1 which is X_2
2	wird X_1 haben	will have X_1
2	wir X_1 damit X_2	we X_1 so that X_2
2	was X_1 hat X_2	what X_1 has X_2
2	was X_1 betrifft so	as regards X_1
2	und X_1 muessen wir X_2	and X_1 we must X_2
2	um X_1 zu gewaehrleisten	to ensure X_1
2	um X_1 zu X_2	to X_2 X_1
2	sowohl X_1 als auch	both X_1 and
2	sie X_1 auch X_2	they also X_1 X_2
2	in erster linie X_1	X_1 in the first instance
2	in X_1 an	in X_1
2	ich X_1 meine	i X_1
2	heute X_1 wir X_2	today we X_1 X_2
2	herr praesident X_1 und herren	mr president X_1 and gentlemen
2	gleich X_1	X_1 a moment
2	es muss X_1 werden	it must be X_1

Figure 1: Examples of the most frequently h rules during the decoding.

tuned separately, so that they used different glu rule weights. That is why we observe the differ ence in the number of glues (and the number c total rules) in the Table 1. We do not observe an significant correlation between the rank of the rul and the hit rate. The Figure 3 shows that the first 10k-ranked rules are hit several times, and then the hit rate stays flat.

We offer an explanation based on our observations of rules used for the decoding. The rules proposed directly from the chart contain a big portion of content words. These rules do not capture any important differences between the structures of the two languages that could not be handled by phrasal rules as well. For example, the rule ⟨die neuen vorschriften sollen X_1,the new rules are X_1⟩ is correct, but a combination of a baseline phrasal rule and glue will produce the same result.

We also see many rules with non-terminals spanning one word. For example, the sequence

(18) die europaeische kommission—the european commission

will produce the rule

(19) ⟨die X_1 kommission, the X_1 commission⟩.

Although the sequence and the rule are high scored by 13 and 15, we intuitively feel that gen-

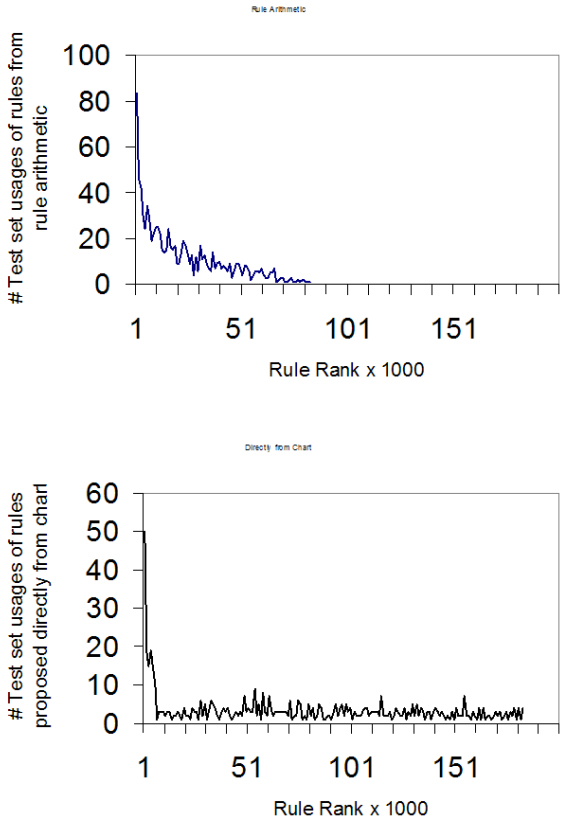


Figure 3: Usage of new rules (DC).

eralizing the word *european* is not very helpful in this context.

The rule arithmetic could propose the rule 19 as

(20) ⟨die X_1, the X_1⟩ + ⟨kommission, commission⟩,

but since the candidates for combination are selected as rules with the highest expected counts (Sec. 3), the rules 20 will most likely loose to the phrase pair 18 and will not be selected.

To conclude our comparison, we observe that both methods produce reliable rules that are often reused in decoding. Nevertheless, since the rule arithmetic combines the most successful rules from each parallel parse, the resulting rules enable structural transformations that could not be handled by baseline rules.

	German-English		Farsi-English	
Model	dev set	test set	dev set	test set
baseline	23.9	25.4	41.1	38.2
RA + EM-i0	24.8	**26.4**	41.8	**40.2**
DC + EM-i0	24.6	25.6		
EM-i0	24.4	26.1	40.8	39.1
EM-i1	24.4	25.8	41.3	38.5
EM-i2	24.4	25.9	41.4	38.2
EM-i3	24.4	26.0	41.3	39.3
EM-i4	24.4	26.0	41.6	39.6
RA	24.4	26.1	40.7	38.4
baseline min1	24.0	25.7		

Table 3: BLEU scores

um X_1	in order X_1
natuerlich X_1	of course X_1
deshalb X_1	this is why X_1
X_1 zu koennen	to X_1
X_1 ist	it is X_1
nach der tagesordnung folgt die X_1	the next item is the X_1
herr X_1 herr kommissar X_2	mr X_1 commissioner X_2
die X_1 der X_2	X_1 the X_2
im gegenteil X_1	on the contrary X_1
nach der tagesordnung folgt X_1	the next item is X_1
X_1 die X_2	the X_1 the X_2
die X_1 die	the X_1
ausserdem X_1	in addition X_1
daher X_1	that is why X_1
wir X_1 nicht X_2	we X_1 not X_2
die X_1 der X_2	the X_2 X_1
deshalb X_1	for this reason X_1
um X_1 zu X_2	to X_2 X_1
X_1 nicht X_2 werden	X_1 not be X_2

Figure 4: Sample rules (RA).

ausserdem X_1 wir	we X_1 also
die X_1 des kommissars	the commissioner 's X_1
den X_1 ratsvorsitz	the X_1 presidency
ich hoffe dass X_1	i would hope that X_1
X_1 ist zu X_2 geworden	X_1 has become X_2
die X_1 des vereinigten koenigreichs	the uk X_1
X_1 maij weggen X_2	X_1 maij weggen X_2
X_1 wir auf X_2 sind	X_1 we are on X_2
ich frage mich X_1	i wonder X_1

Figure 5: Sample rules (DC).

6.3 Farsi-English rules from the rule arithmetic

Although we have only limited resources to qualitatively analyze the Farsi-English experiments, we noticed that there are two major groups of new rules.

The first group corresponds to the fact that Farsi does not have definite article and allows pro-drop. We observe many new rules that could not be learned from word alignments, since some definite articles or pronouns in English were aligned to null (and unaligned words are not allowed at the edges of phrases). However, if the chart contains an insertion (of the determiner or pronoun) with a high expected count, the rule arithmetic may propose new rule by combining it with other rules.

The second group contains rules that help word reordering. We observe rules moving verbs from the S PP O V in Farsi into SVO in English as well as rules reordering wh-clauses.

Most of the rules traced during the test set decoding belong to the second group. Figure 1 shows that the number of new rules hit during the decoding is smaller compared to the German-English experiments. On the other hand, the rules have smaller number of terminals so that we assume that the positive effect of these rules comes from the reordering of non-terminals.

7 Conclusion

In this work, we studied two new methods for learning hierarchical MT rules: the rule arithmetic and proposing directly from the parse forest. We discussed systematic patterns where the rule arithmetic outperforms alignment-based approaches and verified its significant improvement on two different language pairs (German-English and Farsi-English). We also hypothesized why the second method – proposing rules directly from the chart – improves the baseline less than the rule arithmetic.

Acknowledgment

This work is partially supported by the DARPA TRANSTAC program under the contract number NBCH2030007. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of DARPA.

References

Birch, Alexandra, Chris Callison-Burch, Miles Osborne, and Philipp Koehn. 2006. Constraining the phrase-based, joint probability statistical translation model. In *Proceedings on WSMT'06*, pages 154–157.

Blunsom, Phil, Trevor Cohn, Chris Dyer, and Miles Osborne. 2009. A gibbs sampler for phrasal synchronous grammar induction. In *ACL '09*, pages 782–790.

Cherry, Colin. 2007. Inversion transduction grammar for joint phrasal translation modeling. In *NAACL-HLT'07/SSST'07*.

Chiang, David. 2005. A hierarchical phrase-based model for statistical machine translation. In *ACL'05*, pages 263–270.

Chiang, David. 2007. Hierarchical phrase-based translation. *Comput. Linguist.*, 33(2):201–228.

Cmejrek, Martin, Bowen Zhou, and Bing Xiang. 2009. Enriching SCFG rules directly from efficient bilingual chart parsing. In *IWSLT'09*, pages 136–143.

DeNero, John, Alexandre Bouchard-Côté, and Dan Klein. 2008. Sampling alignment structure under a bayesian translation model. In *EMNLP '08*, pages 314–323.

Galley, Michel, Jonathan Graehl, Kevin Knight, Daniel Marcu, Steve DeNeefe, Wei Wang, and Ignacio Thayer. 2006. Scalable inference and training of context-rich syntactic translation models. In *Proc. of ACL*, pages 961–968.

Huang, Songfang and Bowen Zhou. 2009. An EM algorithm for SCFG in formal syntax-based translation. In *Proc. IEEE ICASSP'09*, pages 4813–4816.

Koehn, Philipp, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *ACL*.

Koehn, Philipp. 2005. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of MT Summit*.

Liu, Ding and Daniel Gildea. 2009. Bayesian learning of phrasal tree-to-string templates. In *EMNLP '09*, pages 1308–1317.

Marcu, Daniel and W Wong. 2002. A phrase-based, joint probability model for statistical machine translation. In *Proceedings of EMNLP'02*.

May, Jonathan and Kevin Knight. 2007. Syntactic realignment models for machine translation. In *Proceedings of EMNLP-CoNLL'07*, pages 360–368.

Och, F. J. and H. Ney. 2000. Improved statistical alignment models. In *Proc. of ACL*, pages 440–447, Hong Kong, China, October.

Papineni, K., S. Roukos, T. Ward, and W. Zhu. 2001. Bleu: a method for automatic evaluation of machine translation. Technical Report RC22176, IBM T. J. Watson Research Center.

Wu, Dekai. 1997. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational Linguistics*, 23(3):377–403.

Zhou, Bowen, Bing Xiang, Xiaodan Zhu, and Yuqing Gao. 2008. Prior derivation models for formally syntax-based translation using linguistically syntactic parsing and tree kernels. In *Proceedings of the ACL'08: HLT SSST-2*, pages 19–27.

Unsupervised cleansing of noisy text

Danish Contractor
IBM India Software Labs
dcontrac@in.ibm.com

Tanveer A. Faruquie
IBM Research India
ftanveer@in.ibm.com

L. Venkata Subramaniam
IBM Research India
lvsubram@in.ibm.com

Abstract

In this paper we look at the problem of cleansing noisy text using a statistical machine translation model. Noisy text is produced in informal communications such as Short Message Service (SMS), Twitter and chat. A typical Statistical Machine Translation system is trained on parallel text comprising noisy and clean sentences. In this paper we propose an unsupervised method for the translation of noisy text to clean text. Our method has two steps. For a given noisy sentence, a weighted list of possible clean tokens for each noisy token are obtained. The clean sentence is then obtained by maximizing the product of the weighted lists and the language model scores.

1 Introduction

Noisy unstructured text data is found in informal settings such as Short Message Service (SMS), online chat, email, social message boards, newsgroup postings, blogs, wikis and web pages. Such text may contain spelling errors, abbreviations, non-standard terminology, missing punctuation, misleading case information, as well as false starts, repetitions, and special characters.

We define noise in text as any kind of difference between the surface form of a coded representation of the text and the correct text. The SMS "u kno whn is d last train of delhi metro" is noisy because several of the words are not spelled correctly and there are grammar mistakes. Obviously the person who wrote this message intended to write exactly what is there in the SMS. But still it is considered noisy because the message is coded using non-standard spellings and grammar.

Current statistical machine translation (SMT) systems rely on large parallel and monolingual training corpora to produce high quality translations (Brown et al., 1993). Most of the large parallel corpora available comprise newswire data that include well formed sentences. Even when web sources are used to train a SMT system, noisy portions of the corpora are eliminated (Imamura et al., 2003) (Imamura and Sumita, 2002) (Khadivi and Ney, 2005). This is because it is known that noise in parallel corpora results in incorrect training of models thus degrading the performance.

We are not aware of sufficiently large parallel datasets comprising noisy and clean sentences. In fact, even dictionaries comprising of noisy to clean mappings in one language are very limited in size.

With the increase in noisy text data generated in various social communication media, cleansing of such text has become necessary. The lack of noisy parallel datasets means that this problem cannot be tackled in the traditional SMT way, where translation models are learned based on the parallel dataset. Consider the problem of translating a noisy English sentence e to a clean English sentence h. SMT imagines that e was originally conceived in clean English which when transmitted over the noisy channel got corrupted and became a noisy English sentence. The objective of SMT is to recover the original clean sentence.

Coling 2008: Poster Volume, pages 189–196,
Beijing, August 2010

The goal of this paper is to analyze how noise can be tackled. We present techniques to translate noisy text sentences e to clean text sentences h. We show that it is possible to clean noisy text in an unsupervised fashion by incorporating steps to construct ranked lists of possible clean English tokens and then searching for the best clean sentence. Of course as we will show for a given noisy sentence, several clean sentences are possible. We exploit the statistical machine learning paradigm to let the decoder pick the best alternative from these possible clean options to give the final translation for a given noisy sentence.

The rest of the paper is organized as follows. In section 2 we state our contributions and give an overview of our approach. In Section 3 we describe the theory behind clean noisy text using MT. In Section 4 we explain how we use a weighing function and a plain text dictionary of clean tokens to guess possible clean English language tokens. Section 5 describes our system along with our results. We have given an analysis of the kind of noise present in our data set in section 5.2

2 Our Approach

In this paper we describe an unsupervised method to clean noisy text. We formulate the text cleansing problem in the machine translation framework using translation model 1 (Brown et al., 1993). We clean the text using a pseudo-translation model of clean and noisy words along with a language model trained using a large monolingual corpus. We use a decoder to search for the best clean sentence for a noisy sentence using these models.

We generate scores for the pseudo translation model using a weighing function for each token in an SMS and use these scores along with language model probabilities to hypothesize the best clean sentence for a given noisy SMS. Our approach can be summarized in the following steps:

- Tokenize noisy SMS S into n tokens s_1, s_2 ... s_n. For each SMS token s_i create a weighted list based on a weighing function. These lists along with their scores corresponds to the translation probabilities of the SMT translation model.
- Use the lists generated in the step above along with clean text language model scores, in a decoder to hypothesize the best clean sentence
- At the end of the search choose the highest scoring sentence as the clean translation of the noisy sentence

In the above approach we do not learn the translation model but emulate the translation model during decoding by analyzing the noise of the tokens in the input sentence.

3 Noisy sentence translation

Statistical Translation models were invented by Brown, et al (Brown et al., 1993) and are based on the source-channel paradigm of communication theory. Consider the problem of translating a noisy sentence e to a clean sentence h. We imagine that e was originally conceived cleanly which when transmitted over the noisy communication channel got corrupted and became a noisy sentence. The goal is to get back the original clean sentence from the noisy sentence. This can be expressed mathematically as

$$\hat{h} = arg\max_{h} Pr(h|e)$$

By Bayes' Theorem

$$\hat{h} = arg\max_{h} Pr(e|h)Pr(h)$$

Conceptually, the probability distribution $P(e|h)$ is a table which associates a probability score with every possible pair of clean and noisy sentences (e, h). Every noisy sentence e is a candidate translation of a given clean sentence h. The goodness of the translation $h \Rightarrow e$ is given by the probability score of the pair (e, h). Similarly, $Pr(h)$ is a table which associates a probability score with every possible clean sentence h and measures how well formed the sentence h is.

It is impractical to construct these tables exactly by examining individual sentences (and sentence pairs) since the number of conceivable sentences in any language is countably infinite. Therefore, the challenge in Statistical Machine Translation is to construct approximations to the probability

distributions $P(e|h)$ and $Pr(h)$ that give an acceptable quality of translation. In the next section we describe a model which is used to approximate $P(e|h)$.

3.1 IBM Translation Model 2

IBM translation model 2 is a generative model, i.e., it describes how a noisy sentence e could be stochastically generated given a clean sentence h. It works as follows:

- Given a clean sentence h of length l, choose the length (m) for the noisy sentence from a distribution $\epsilon(m|l)$.
- For each position $j = 1, 2, \ldots m$ in the noisy string, choose a position a_j in the clean string from a distribution $a(a_j|j, l, m)$. The mapping $\mathbf{a} = (a_1, a_2, \ldots, a_m)$ is known as alignment between the noisy sentence e and the clean sentence h. An alignment between e and h tells which word of e is the corrupted version of the corresponding word of h.
- For each $j = 1, 2, \ldots m$ in the noisy string, choose an noisy word e_j according to the distribution $t(e_j|h_{a_j})$.

It follows from the generative model that probability of generating $e = e_1 e_2 \ldots e_m$ given $h = h_1 h_2 \ldots h_l$ with alignment $\mathbf{a} = (a_1, a_2, \ldots, a_m)$ is

$$Pr(e, a|h) = \epsilon(m|l) \prod_{j=1}^{m} t(e_j|h_{a_j}) a(a_j|j, m, l).$$

It can be easily seen that a sentence e could be produced from h employing many alignments and therefore, the probability of generating e given h is the sum of the probabilities of generating e given h under all possible alignments a, i.e., $Pr(e|h) = \sum_a Pr(e, a|h)$. Therefore,

$$Pr(e|h) =$$

$$\epsilon(m|l) \sum_{a_1=0}^{l} .. \sum_{a_m=0}^{l} \prod_{j=1}^{m} t(e_j|h_{a_j}) a(a_j|j, m, l).$$

The above expression can be rewritten as follows:

$$Pr(e|h) = \epsilon(m|l) \prod_{j=1}^{m} \sum_{i=0}^{l} t(e_j|h_i) a(i|j, m, l).$$

Typical statistical machine translation systems use large parallel corpora to learn the translation probabilities (Brown et al., 1993). Traditionally such corpora have consisted of news articles and other well written articles. Therefore in theory $P(e|h)$ should be constructed by examining sentence pairs of clean and noisy sentences. There exists some work to remove noise from SMS (Choudhury et al., 2007) (Byun et al., 2008) (Aw et al., 2006) (Neef et al., 2007) (Kobus et al., 2008). However, all of these techniques require an aligned corpus of SMS and conventional language for training.

Aligned parallel corpora for noisy sentence is difficult to obtain. This lack of data for a language and the domain dependence of noise makes it impractical to construct corpus from which $P(e|h)$ can be learnt automatically. This leads to difficulty in learning $P(e|h)$. Fortunately the alignment between clean and noisy sentences are monotonic in nature hence we assume a uniform distribution for $a(i|j, m, l)$ held fixed at $(l+1)^{-1}$. This is equivalent to model 1 of IBM translation model. The translation models $t(e_j|h_{a_j})$ can be thought of as a ranked list of noisy words given a clean word. In section 4.2 we show how this ranked list can be constructed in an unsupervised fashion.

3.2 Language Model

The problem of estimating the sentence formation distribution $Pr(h)$ is known as the language modeling problem. The language modeling problem is well studied in literature particularly in the context of speech recognition. Typically, the probability of a n-word sentence $h = h_1 h_2 \ldots h_n$ is modeled as $Pr(h) = Pr(h_1|H_1)Pr(h_2|H_2) \ldots Pr(h_n|H_n)$, where H_i is the *history* of the ith word h_i. One of the most popular language models is the n-gram model (Brown et al., 1993) where the history of a word consists o f the word and the previous $n-1$ words in the sentence, i.e., $H_i = h_i h_{i-1} \ldots h_{i-n+1}$. In our application we use a smoothed trigram model.

3.3 Decoding

The problem of searching for a sentence h which minimizes the product of translation model prob-

ability and the language model probability is known as the decoding problem. The decoding problem has been proved to be NP-complete even when the translation model is IBM model 1 and the language model is bi-gram (K Knight., 1999). Effective suboptimal search schemes have been proposed (F. Jelinek, 1969), (C. Tillman et al., 1997).

4 Pseudo Translation Model

In order to be able to exploit the SMT paradigm we first construct a pseudo translation model. The first step in this direction is to create noisy token to clean token mapping. In order to process the noisy input we first have to map noisy tokens in noisy sentence, S^e, to the possible correct lexical representations. We use a similarity measure to map the noisy tokens to their clean lexical representations .

4.1 Similarity Measure

For a term $t_e \in \mathcal{D}^e$, where $\mathcal{D}^e$ is a dictionary of possible clean tokens, and token s_i of the noisy input S^e, the similarity measure $\gamma(t_e, s_i)$ between them is

$$\gamma(t_e, s_i) = \begin{cases} \frac{LCSRatio(t_e,s_i)}{EditDistance_{SMS}(t_e,s_i)} & \text{if } t_e \text{ and } s_i \text{ share same starting character} \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where $LCSRatio(t_e, s_i) = \frac{length(LCS(t_e,s_i))}{length(t_e)}$ and $LCS(t_e, s_i)$ is the *Longest common subsequence* between t_e and s_i. The intuition behind this measure is that people typically type the first few characters of a word in an SMS correctly. This way we limit the possible variants for a particular noisy token.

The *Longest Common Subsequence Ratio* (LCSRatio) (Melamed et al., 1999) of two strings is the ratio of the length of their LCS and the length of the longer string. Since in the SMS scenario, the dictionary term will always be longer than the SMS token, the denominator of LCSR is taken as the length of the dictionary term.

The $EditDistance_{SMS}$ (Figure 1) compares the Consonant Skeletons (Prochasson et al., 2007) of the dictionary term and the SMS token. If the Levenshtein distance between consonant skeletons is small then $\gamma(t_e, s_i)$ will be high. The intuition behind using $EditDistance_{SMS}$ can be explained through an example. Consider an SMS token "gud" whose most likely correct form is "good". The two dictionary terms "good" and "guided" have the same LCSRatio of 0.5 w.r.t "gud", but the $EditDistance_{SMS}$ of "good" is 1 which is less than that of "guided", which has $EditDistance_{SMS}$ of 2 w.r.t "gud". As a result the similarity measure between "gud" and "good" will be higher than that of "gud" and "guided". Higher the *LCSRatio* and lower the $EditDistance_{SMS}$, higher will be the similarity measure. Hence, for a given SMS token "byk", the similarity measure of word "bike" is higher than that of "break".

In the next section we show how we use this similarity measure to construct ranked lists. Ranked lists of clean tokens have also been used in FAQ retrieval based on noisy queries (Kothari et al., 2009).

Procedure $EditDistance_{SMS}(t_e, s_i)$
Begin
  return $LevenshteinDistance(CS(s_i), CS(t_e)) + 1$
End

Procedure *CS (t): // Consonant Skeleton Generation*
Begin
  Step 1. remove consecutive repeated characters in t
    // $(fall \rightarrow fal)$
  Step 2. remove all vowels in t
    //$(painting \rightarrow pntng, threat \rightarrow thrt)$
  return t
End

Figure 1: $EditDistance_{SMS}$

4.2 List Creation

For a given noisy input string S^e, we tokenize it on white space and replace any occurrence of digits to their string based form (e.g. 4get, 2day) to get a series of n tokens $s_1, s_2, \ldots, s_n$. A list L_i^e is created for each token s_i using terms in a dic-

hv u cmplted ure prj rprt
d ddline fr sbmission of d rprt hs bn xtnded
i wil be lte by 20 mns
d docs shd rech u in 2 days
thnk u for cmg 2 d prty

Figure 2: Sample SMS queries

tionary D^e consisting of clean english words. A term t_e from D^e is included in L_i^e if it satisfies the threshold condition

$$\gamma(t_e, s_i) > \phi \quad (2)$$

Heuristics are applied to boost scores of some words based on positional properties of characters in noisy and clean tokens. The scores of the following types of tokens are boosted:

1. Tokens that are a substring of a dictionary words from the first character.

2. Tokens having the same first and last character as a dictionary word.

3. Token that are dictionary words themselves (clean text).

The threshold value ϕ is determined experimentally. Thus we select only the top scoring possible clean language tokens to construct the sentence.

Once the list are constructed the similarity measure along with the language model scores is used by the decoding algorithm to find the best possible English sentence. It is to be noted that these lists are constructed at decoding time since they depend on the noisy surface forms of words in the input sentence.

5 Experiments

To evaluate our system we used a set of 800 noisy English SMSes sourced from the publicly available National University of Singapore SMS corpus[1] and a collection of SMSes available from the Indian Institute of Technology, Kharagpur. The SMSes are a collection of day-to-day SMS exchanges between different users. We manually

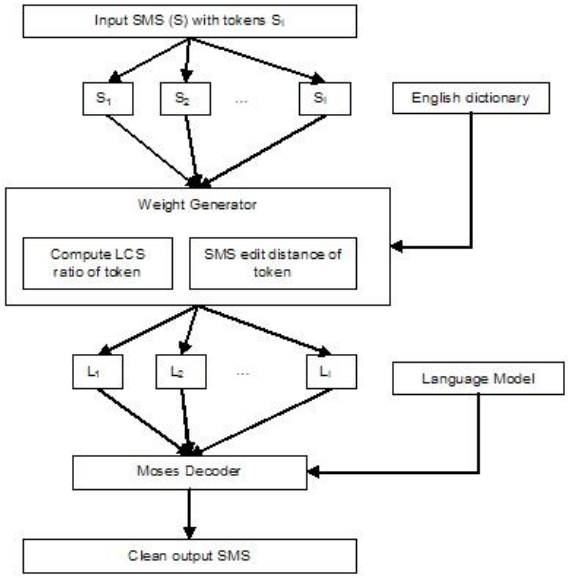


Figure 3: System implementation

	BLEU scores	1-gram	2-gram	3-gram	4-gram
Noisy text	40.96	63.7	45.1	34.5	28.3
Cleaned text	53.90	77.5	58.7	47.4	39.5

Table 1: BLEU scores

generated a cleaned english version of our test set to use as a reference.

The noisy SMS tokens were used to generate clean text candidates as described in section 4.2. The dictionary D^e used for our experiments was a plain text list of 25,000 English words. We created a tri-gram language model using a collection of 100,000 clean text documents. The documents were a collection of articles on news, sporting events, literature, history etc. For decoding we used Moses[2], which is an open source decoder for SMT (Hoang et al., 2008), (Koehn et al., 2007). The noisy SMS along with clean candidate token lists, for each SMS token and language model probabilities were used by Moses to hypothesize the best clean english output for a given noisy SMS. The language model and translation models weights used by Moses during the decoding phase, were adjusted manually after some experimentation.

We used BLEU (Bilingual evaluation understudy) and Word error rate (WER) to evaluate the performance of our system. BLEU is used to

[1] http://wing.comp.nus.edu.sg/downloads/smsCorpus

[2] http://www.statmt.org/moses/

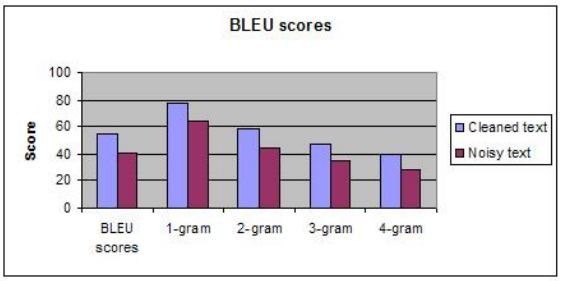


Figure 4: Comparison of BLEU scores

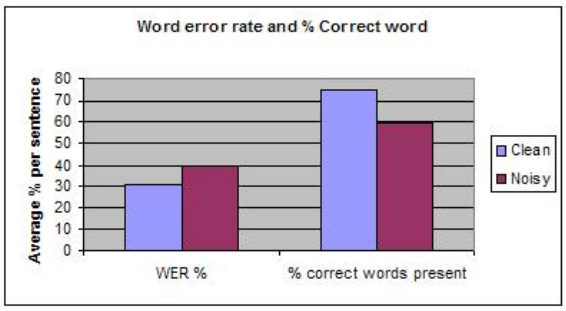


Figure 5: Word error rates

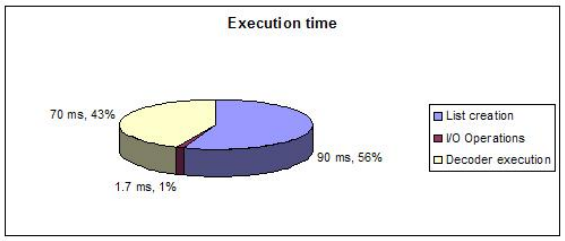


Figure 6: Execution time slices

establish similarity between a system translated and human generated reference text. A noisy SMS ideally has only one possible clean translation and all human evaluators are likely to provide the same translation. Thus, BLEU which makes use of n-gram comparisons between reference and system generated text, is very useful to measure the accuracy of our system. As shown in Fig 4 , our system reported significantly higher BLEU scores than unprocessed noisy text.

The word error rate is defined as

$$WER = \frac{S + D + I}{N} \quad (3)$$

where S is the number of substitutions, D is the number of the deletions, I is the number of the insertions and N is the number of words in the reference The WER can be thought of as an execution of the Levenstein Edit distance algorithm at the token level instead of character level.

Fig 5 shows a comparison of the WER. Sentences generated from our system had 10 % lower WER as compared to the unprocessed noisy sentences. In addition, the sentences generated by our system match a higher number of tokens (words) with the reference sentences, as compared to the noisy sentences.

5.1 System performance

Unlike standard MT system when $P(e|h)$ is precomputed during the training time, list generation in our system is dynamic because it depends on the noisy words present in the input sentence. In this section we evaluate the computation time for list generation along with the decoding time for finding the best list. We used an Intel Core 2 Duo 2.2 GHz processor with 3 GB DDR2 RAM to implement our system. As shown in Fig 6 the additional computation involving list creation etc takes up 56% (90 milliseconds) of total translation time. 43% of the total execution time is taken by the decoder, while I/O operations take only 1% of the total execution time. The decoder execution time slices reported above exclude the time taken to load the language model. Moses took approximately 10 seconds to load our language model.

5.2 Measuring noise level in SMS queries

The noise in the collected SMS corpus can be categorized as follows

1. Removal of characters : The commonly observed patterns include deletion of vowels (as in “msg” for “message”), deletion of repeated character (as in ”happy” for “hapy”) and truncation (as in “tue” for “tuesday”)

Type of Noise	% of Total Noisy Tokens
Deletion of Characters	48%
Phonetic Substitution	33%
Abbreviations	5%
Dialectical Usage	4%
Deletion of Words	1.2%

Table 2: Measure of Types of SMS Noise

	Clean (Reference) text	Noisy text	Output text
Perplexity	19.61	34.56	21.77

Table 3: Perplexity for Reference, Noisy Cleaned SMS

2. Phonetic substitution: For example, “2” for “to” or “too”, “lyf”’ for “life”, “lite” for “light” etc.

3. Abbreviation: Some frequently used abbreviations are “tb” for “text back”, “lol” for “laughs out loud”, “AFAICT” for “as far as i can tell” etc.

4. Dialectal and informal usage: Often multiple words are combined into a single token following certain dialectal conventions. For example, “gonna” is used for “going to”, “aint” is used for “are not”, etc.

5. Deletion of words: Function words (e.g. articles) and pronouns are commonly deleted. “I am reading the book” for example may be typed as “readin bk”.

Table 2 lists statistics on these noise types from 101 SMSes selected at random from our data set. The average length of these SMSes was 13 words. Out of the total number of words in the SMSes, 52% were non standard words. Table 2 lists the statistics for the types of noise present in these non standard words.

Measuring character level perplexity can be another way of estimating noise in the SMS language.The perplexity of a LM on a corpus gives an indication of the average number of bits needed per n-gram to encode the corpus. Noise results in the introduction of many previously unseen n-grams in the corpus. Higher number of bits are needed to encode these improbable n-grams which results in increased perplexity.

We built a character-level language model (LM) using a document collection (vocabulary size is 20K) and computed the perplexity of the language model on the noisy and the cleaned SMS test-set and the SMS reference data.

From Table 3 we can see the difference in perplexity for noisy and clean SMS data. Large perplexity values for the SMS dataset indicates a high level of noise. The perplexity evaluation indicates that our method is able to remove noise from the input queries as given by the perplexity and is close to the human correct reference corpus whose perplexity is 19.61.

6 Conclusion

We have presented an inexpensive, unsupervised method to clean noisy text. It does not require the use of a noisy to clean language parallel corpus for training. We show how a simple weighing function based on observed heuristics and a vocabulary file can be used to shortlist clean tokens. These tokens and their weights are used along with language model scores, by a decoder to select the best clean language sentence.

References

Monojit Choudhury, Rahul Saraf, Vijit Jain, Animesh Mukherjee, Sudeshna Sarkar, Anupam Basu. 2007. Investigation and modeling of the structure of texting language. *International Journal on Document Analysis and Recognition*.

Jeunghyun Byun, Seung-Wook Lee, Young-In Song, Hae-Chang Rim. 2008. Two Phase Model for SMS Text Messages Refinement. *In Proceedings of AAAI Workshop on Enhanced Messaging*.

Aiti Aw, Min Zhang, Juan Xiao, and Jian Su. 2006. A phrase-based statistical model for SMS text normalization. *In Proceedings of COLING-ACL.*

Guimier de Neef, Emilie, Arnaud Debeurme, and Jungyeul Park. 2007. TILT correcteur de SMS : Evaluation et bilan quantitatif. *In Actes de TALN*, Toulouse, France.

Catherine Kobus, Francois Yvon and Geraldine Damnati. 2008. Normalizing SMS: Are two metaphors better than one? *In Proceedings of COLING*, Manchester.

Sreangsu Acharya, Sumit Negi, L Venkata Subramaniam, Shourya Roy. 2009. Language independent unsupervised learning of short message service dialect. *International Journal on Document Analysis and Recognition*.

Philipp Koehn, Hieu Hoang, Alexandra Birch Mayne, Christopher Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, Evan Herbst 2007. Moses: Open source toolkit for statistical machine

translation. *In Proceedings of ACL, Demonstration Session* .

Peter F. Brown, Vincent J.Della Pietra, Stephen A. Della Pietra, Robert. L. Mercer 1993. The Mathematics of Statistical Machine Translation: Parameter Estimation *Computational Linguistics*.

I. D. Melamed. 1999. Bitext maps and alignment via pattern recognition. *Computational Linguistics.*

E. Prochasson, C. Viard-Gaudin, and E. Morin. 2007. Language models for handwritten short message services. *In Proceedings of ICDAR.*

S. Khadivi and H. Ney. 2005. Automatic filtering of bilingual corpora for statistical machine translation. *In Proceedings of NLDB*, pages 263–274, 2005.

K. Imamura and E. Sumita. 2002. Bilingual corpus cleaning focusing on translation literality. In *In Proceedings of ICSLP*.

K. Imamura, E. Sumita, and Y. Matsumoto. 2003. Automatic construction of machine translation knowledge using translation literalness. In *In Proceedings of EACL.*

K. Knight, 1999. Decoding complexity in word replacement translation models. *Computational Linguistics*.

F. Jelinek, 1969. A fast sequential decoding algorithm using a stack. *IBM Journal of Research and Development*.

C. Tillman, S. Vogel, H. Ney, and A. Zubiaga. 1997. A DP-based search using monotone alignments in statistical translation. *In Proceedings of ACL.*

Hieu Hoang, Philipp Koehn. 2008. Design of the Moses decoder for statistical machine translation. *In Proceedings of ACL Workshop on Software Engineering, Testing, and Quality Assurance for Natural Language Processing*.

Govind Kothari, Sumit Negi, Tanveer A. Faruquie, Venkatesan T. Chakraverthy, L. Venkata Subramaniam. 2009. SMS based interface for FAQ retrieval, In *In Proceedings of ACL-IJCNLP*

Improving Reordering with Linguistically Informed Bilingual n-grams

Josep Maria Crego
LIMSI-CNRS
jmcrego@limsi.fr

François Yvon
LIMSI-CNRS & Univ. Paris Sud
yvon@limsi.fr

Abstract

We present a new reordering model estimated as a standard n-gram language model with units built from morpho-syntactic information of the source and target languages. It can be seen as a model that translates the morpho-syntactic structure of the input sentence, in contrast to standard translation models which take care of the surface word forms. We take advantage from the fact that such units are less sparse than standard translation units to increase the size of bilingual context that is considered during the translation process, thus effectively accounting for mid-range reorderings. Empirical results on French-English and German-English translation tasks show that our model achieves higher translation accuracy levels than those obtained with the widely used lexicalized reordering model.

1 Introduction

Word ordering is one of the major issues in statistical machine translation (SMT), due to the many word order peculiarities of each language. It is widely accepted that there is a need for structural information to account for such differences. Structural information, such as Part-of-speech (POS) tags, chunks or constituency/dependency parse trees, offers a greater potential to learn generalizations about relationships between languages than models based on word surface forms, because such "surfacist" models fail to infer generalizations from the training data.

The word ordering problem is typically decomposed in a number of related problems which can be further explained by a variety of linguistic phenomena. Accordingly, we can sort out the reordering problems into three categories based on the kind of linguistic units involved and/or the typical distortion distance they imply. Roughly speaking, we face *short-range* reorderings when single words are reordered within a relatively small window distance. It consist of the easiest case as typically, the use of phrases (in the sense of translation units of the phrase-based approach to SMT) is believed to adequately perform such reorderings. *Mid-range* reorderings involve reorderings between two or more phrases (translation units) which are closely positioned, typically within a window of about 6 words. Many alternatives have been proposed to tackle mid-range reorderings through the introduction of linguistic information in MT systems. To the best of our knowledge, the authors of (Xia and McCord, 2004) were the first to address this problem in the statistical MT paradigm. They automatically build a set of linguistically grounded rewrite rules, aimed at reordering the source sentence so as to match the word order of the target side. Similarly, (Collins, et al 2005) and (Popovic and Ney, 2006) reorder the source sentence using a small set of hand-crafted rules for German-English translation. (Crego and Mariño, 2007) show that the ordering problem can be more accurately solved by building a source-sentence word lattice containing the most promising reordering hypotheses, allowing the decoder to decide for the best word order hypothesis. Word lattices are built by means of rewrite rules operating on POS tags; such rules are automatically extracted from the training bi-text. (Zhang, et al 2007) introduce shallow parse (chunk) information to reorder the source sentence, aiming at extending the scope of their rewrite rules, encoding reordering hypotheses in the form of a confusion network that is then passed to the decoder. These studies tackle mid-range reorderings by predicting more or less accurate reordering hypotheses. However, none

Coling 2008: Poster Volume, pages 197–205,
Beijing, August 2010

of them introduce a reordering model to be used in decoding time. Nowadays, most of SMT systems implement the well known *lexicalized reordering* model (Tillman, 2004). Basically, for each translation unit it estimates the probability of being translated *monotone*, *swapped* or placed *discontiguous* with respect to its previous translation unit. Integrated within the *Moses* (Koehn, et al 2007) decoder, the model achieves state-of-the-art results for many translation tasks. One of the main reasons that explains the success of the model is that it considers information of the source- and target-side surface forms, while the above mentionned approaches attempt to hypothesize reorderings relying only on the information contained on the source-side words.

Finally, *long-range* reorderings imply reorderings in the structure of the sentence. Such reorderings are necessary to model the translation for pairs like Arabic-English, as English typically follows the SVO order, while Arabic sentences have different structures. Even if several attempts exist which follow the above idea of making the ordering of the source sentence similar to the target sentence before decoding (Niehues and Kolss, 2009), long-range reorderings are typically better addressed by syntax-based and hierarchical (Chiang, 2007) models. In (Zollmann et al., 2008), an interesting comparison between phrase-based, hierarchical and syntax-augmented models is carried out, concluding that hierarchical and syntax-based models slightly outperform phrase-based models under large data conditions and for sufficiently non-monotonic language pairs.

Encouraged by the work reported in (Hoang and Koehn, 2009), we tackle the mid-range reordering problem in SMT by introducing a n-gram language model of bilingual units built from POS information. The rationale behind such a model is double: on the one hand we aim at introducing morpho-syntactic information into the reordering model, as we believe it plays an important role for predicting systematic word ordering differences between language pairs; at the same time that it drastically reduces the sparseness problem of standard translation units built from surface forms. On the other hand, n-gram language modeling is a robust approach, that enables to account for arbitrary large sequences of units. Hence, the proposed model takes care of the translation adequacy of the structural information present in translation hypotheses, here introduced in the form of POS tags. We also show how the new model compares to a widely used *lexicalized reordering* model, which we have also implemented in our particular bilingual n-gram approach to SMT, as well as to the widely known *Moses* SMT decoder, a state-of-the-art decoder performing lexicalized reordering.

The remaining of this paper is as follows. In Section 2 we briefly describe the bilingual n-gram SMT system. Section 3 details the bilingual n-gram reordering model, the main contribution of this paper, and introduces additional well known reordering models. In Section 4, we analyze the reordering needs of the language pairs considered in this work and we carry out evaluation experiments. Finally, we conclude and outline further work in Section 5.

2 Bilingual n-gram SMT

Our SMT system defines a translation hypothesis t given a source sentence s, as the sentence which maximizes a linear combination of feature functions:

$$\hat{t}_1^I = \arg\max_{t_1^I} \left\{ \sum_{m=1}^{M} \lambda_m h_m(s_1^J, t_1^I) \right\} \quad (1)$$

where λ_m is the weight associated with the feature $h_m(s,t)$. The main feature is the log-score of the translation model based on bilingual n-grams. This model constitutes a language model of a particular *bi-language* composed of bilingual units which are typically referred to as *tuples* (Mariño et al., 2006). In this way, the translation model probabilities at the sentence level are approximated by using n-grams of tuples:

$$p(s_1^J, t_1^I) = \prod_{k=1}^{K} p((s,t)_k | (s,t)_{k-1} \ldots (s,t)_{k-n+1})$$

where s refers to source t to target and $(s,t)_k$ to the k^{th} tuple of the given bilingual sentence pairs, s_1^J and t_1^I. It is important to notice that, since both languages are linked up in tuples, the context

information provided by this translation model is bilingual. As for any standard n-gram language model, our translation model is estimated over a training corpus composed of sentences of the language being modeled, in this case, sentences of the *bi-language* previously introduced. Translation units consist of the core elements of any SMT system. In our case, tuples are extracted from a word aligned corpus in such a way that a unique segmentation of the bilingual corpus is achieved, allowing to estimate the n-gram model. Figure 1 presents a simple example illustrating the unique tuple segmentation for a given word-aligned pair of sentences (top).

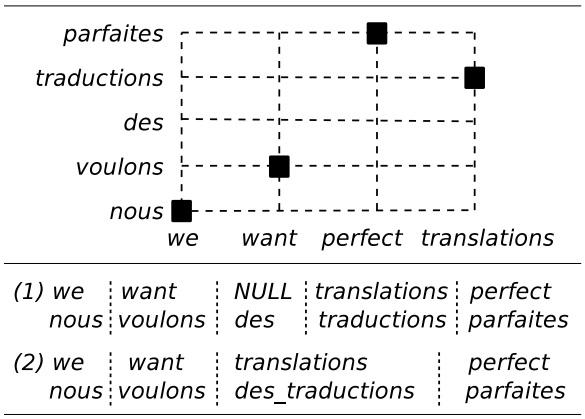


Figure 1: *Tuple extraction from an aligned sentence pair.*

The resulting sequence of tuples *(1)* is further refined to avoid *NULL* words in source side of the tuples *(2)*. Once the whole bilingual training data is segmented into tuples, n-gram language model probabilities can be estimated. Notice from the example that the English source words *perfect* and *translations* have been reordered in the final tuple segmentation, while the French target words are kept in their original order. During decoding, sentences to be translated are encoded in the form of word lattices containing the most promising reordering hypotheses, so as to reproduce the word order modifications introduced during the tuple extraction process. Hence, at decoding time, only those reordering hypotheses encoded in the word lattice are examined. Reordering hypotheses are introduced following a set of reordering rules automatically learned from the bi-text corpus word alignments.

Following on the previous example, the rule *perfect translations* $\rightsquigarrow$ *translations perfect* produces the swap of the English words that is observed for the French and English pair. Typically, POS information is used to increase the generalization power of such rules. Hence, rewrite rules are built using POS instead of surface word forms. See (Crego and Mariño, 2007) for details on tuples extraction and reordering rules.

3 Reordering Models

In this section, we detail three different reordering models implemented in our SMT system. As previously outlined, the purpose of reordering models is to accurately learn generalizations for the word order modifications introduced on the source side during the tuple extraction process.

3.1 Source n-gram Language Model

We employ a n-gram language model estimated over the source words of the training corpus after being reordered in the tuple extraction process. Therefore, the model scores a given source-side reordering hypothesis according to the reorderings performed in the training sentences.

POS tags are used instead of surface forms in order to improve generalization and to reduce sparseness. The model is estimated as any standard n-gram language model, described by the following equation:

$$p(s_1^J, t_1^I) = \prod_{j=1}^{J} p(s_j^t | s_{j-1}^t, \ldots, s_{j-n+1}^t) \quad (2)$$

where s_j^t relates to the POS tag used for the j^{th} source word.

The main drawback of this model is the lack of knowledge of the hypotheses on the target-side. The probability assigned to a sequence of source words is only conditioned to the sequence of source words.

3.2 Lexicalized Reordering Model

A broadly used reordering model for phrase-based systems is lexicalized reordering (Tillman, 2004). It introduces a probability distribution for each phrase pair that indicates the likelihood of being

translated *monotone*, *swapped* or placed *discontiguous* to its previous phrase. The ordering of the next phrase with respect to the current phrase is typically also modeled. In our implementation, we modified the three orientation types and consider: a *consecutive* type, where the original monotone and swap orientations are lumped together, a *forward* type, specifying discontiguous forward orientation, and a *backward* type, specifying discontiguous backward orientation. Empirical results showed that in our case, the new orientations slightly outperform the original ones. This may be explained by the fact that the model is applied over tuples instead of phrases.

Counts of these three types are updated for each unit collected during the training process. Given these counts, we can learn probability distributions of the form $p_r(orientation|(st))$ where $orientation \in \{c, f, b\}$ (consecutive, forward and backward) and (st) is a translation unit. Counts are typically smoothed for the estimation of the probability distribution. A major weakness of the lexicalized reordering model is due to the fact that it does not considers phrase neighboring, *i.e.* a single probability is learned for each phrase pair without considering its context. An additional concern is the problem of sparse data: translation units may occur only a few times in the training data, making it hard to estimate reliable probability distributions.

3.3 Linguistically Informed Bilingual n-gram Language Model

The bilingual n-gram LM is estimated as a standard n-gram LM over translation units built from POS tags represented as:

$$p(s_1^J, t_1^I) = \prod_{k=1}^{K} p((st)_k^t | (st)_{k-1}^t \dots (st)_{k-n+1}^t)$$

where $(st)_k^t$ relates to the k^{th} translation unit, $(st)_k$, built from POS tags instead of words.

This model aims at alleviating the drawbacks of the previous two reordering models. On the one hand it takes into account bilingual information to model reordering. On the other hand it considers the phrase neighboring when estimating the reordering probability of a given translation unit. Figure 2 shows the sequence of translation units built from POS tags, used in our previous example.

PP	*VBP*	*NNS*	*JJ*
PRO:PER	*VER:pres*	*PRP:det_NOM*	*ADJ*

Figure 2: *Sequence of POS-tagged units used to estimate the bilingual n-gram LM.*

POS-tagged units used in our model are expected to be much less sparse than those built from surface forms, allowing to estimate higher order language models. Therefore, larger bilingual context are introduced in the translation process. This model can also be seen as a translation model of the sentence structure. It models the adequacy of translating sequences of source POS tags into target POS tags.

Note that the model is not limited to using POS information. Rather, many other information sources could be used (supertags, additional morphology features, *etc.*), allowing to model different translation properties. However, we must take into account that the degree of sparsity of the model units, which is directly related to the information they contain, affects the level of bilingual context finally introduced in the translation process. Since more informed units may yield more accurate predictions, more informed units may also force the model to fall to lower n-grams. Hence, the degree of accuracy and generalization power of the model units must be carefully balanced to allow good reordering predictions for contexts as large as possible.

As any standard language model, smoothing is needed. Empirical results showed that Kneser-Ney smoothing (Kneser and Ney, 1995) achieved the best performance among other options (measured in terms of translation accuracy).

3.4 Decoding Issues

A straightforward implementation of the three models is carried out by extending the log-linear combination of equation (1) with the new features. Note that no additional decoding complexity is introduced in the baseline decoding implementation. Considering the bilingual n-gram language model, the decoder must know the POS tags for

each tuple. However, each tuple may be tagged differently, as words with same surface form may have different POS tags.

We have implemented two solutions for this situation. Firstly, we assume that each tuple has a single POS-tagged version. Accordingly, we select a single POS-tagged version out of the multiple choices (the most frequent). Secondly, all POS-tagged versions of each tuple are allowed. The second choice implies using more accurate POS-tagged tuples to model reordering, however, it overpopulates the search space with spurious hypotheses, as multiple identical units (with different POS tags) are considered.

Our first empirical findings showed no differences in translation accuracy for both configurations. Hence, in the remaining of this paper we only consider the first solution (a single POS-tagged version of each tuple). The training corpus composed of tagged units out of which our new model is estimated is accordingly modified to contain only those tagged units considered in decoding. Note that most of the ambiguity present in word tagging is resolved by the fact that translation units may contain multiple source and target side words.

4 Evaluation Framework

In this section, we perform evaluation experiments of our novel reordering model. First, we give details of the corpora and baseline system employed in our experiments and analyze the reordering needs of the translation tasks, French-English and German-English (in both directions). Finally, we evaluate the performance of our model and contrast results with other reordering models and translation systems.

4.1 Corpora

We have used the fifth version of the *EPPS* and the *News Commentary* corpora made available in the context of the *Fifth ACL Workshop on Statistical Machine Translation*. Table 1 presents the basic statistics for the training and test data sets. Our test sets correspond to *news-test2008* and *newstest2009* file sets, hereinafter referred to as *Tune* and *Test* respectively.

French, German and English Part-of-speech tags are computed by means of the *TreeTagger* [1] toolkit. Additional German tags are obtained using the *RFTagger* [2] toolkit, which annotates text with fine-grained part-of-speech tags (Schmid and Laws, 2008) with a vocabulary of more than 700 tags containing rich morpho-syntactic information (gender, number, case, tense, *etc.*).

Lang.	*Sent.*	*Words*	*Voc.*	*OOV*	*Refs*
Train					
French	1.75 M	52.4 M	137 k	–	–
English	1.75 M	47.4 M	138 k	–	–
Tune					
French	2,051	55.3 k	8,957	1,282	1
English	2,051	49.2 k	8,359	1,344	1
Test					
French	2,525	72.8 k	10,832	1,749	1
English	2,525	65.1 k	9,568	1,724	1
Train					
German	1,61 M	42.2 M	381 k	–	–
English	1,61 M	44.2 M	137 k	–	–
Tune					
German	2,051	47,8 k	10,994	2,153	1
English	2,051	49,2 k	8,359	1,491	1
Test					
German	2,525	62,8 k	12,856	2,704	1
English	2,525	65,1 k	9,568	1,810	1

Table 1: *Statistics for the training, tune and test data sets.*

4.2 System Details

After preprocessing the corpora with standard tokenization tools, word-to-word alignments are performed in both directions, source-to-target and target-to-source. In our system implementation, the *GIZA++* toolkit[3] is used to compute the word alignments. Then, the *grow-diag-final-and* (Koehn et al., 2005) heuristic is used to obtain the alignments from which tuples are extracted.

In addition to the tuple n-gram translation model, our SMT system implements six additional feature functions which are linearly com-

[1] www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger
[2] www.ims.uni-stuttgart.de/projekte/corplex/RFTagger
[3] http://www.fjoch.com/GIZA++.html

bined following a discriminative modeling framework (Och and Ney, 2002): a *target-language model* which provides information about the target language structure and fluency; two *lexicon models*, which constitute complementary translation models computed for each given tuple; a 'weak' distance-based *distortion model*; and finally a *word-bonus model* and a *tuple-bonus model* which are used in order to compensate for the system preference for short translations.

All language models used in this work are estimated using the *SRI language modeling* toolkit[4]. According to our experience, Kneser-Ney smoothing (Kneser and Ney, 1995) and interpolation of lower and higher n-grams options are used as they typically achieve the best performance. Optimization work is carried out by means of the widely used *MERT* toolkit[5] which has been slightly modified to perform optimizations embedding our decoder. The *BLEU* (Papineni et al., 2002) score is used as objective function for MERT and to evaluate test performance.

4.3 Reordering in German-English and French-English Translation

Two factors are found to greatly impact the overall translation performance: the morphological mismatch between languages, and their reordering needs. The vocabulary size is strongly influenced by the number of word forms for number, case, tense, mood, *etc.*, while reordering needs refer to the difference in their syntactic structure. In this work, we are primarily interested on the reordering needs of each language pair. Figure 3 displays a quantitative analysis of the reordering needs for the language pairs under study.

Figure 3 displays the ($\%$) distribution of the reordered sequences, according to their size, observed for the training bi-texts of both translation tasks. Word alignments are used to determine reorderings. A reordering sequence can also be seen as the sequence of words implied in a reordering rule. Hence, we used the reordering rules extracted from the training corpus to account for reordering sequences. Coming back to the example of Figure 1, a single reordering sequence is found, which considers the source words *perfect translations*.

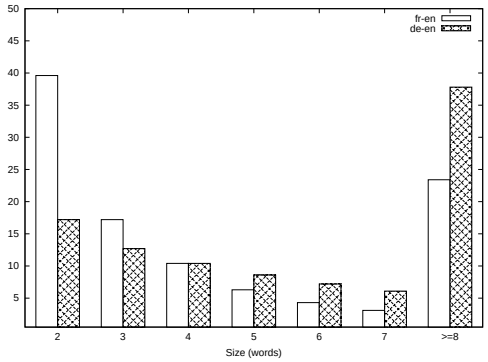


Figure 3: *Size (in words) of reorderings (%) observed in training bi-texts.*

As can be seen, the French-English and German-English pairs follow a different distribution of reorderings according to their size. A lower number of short-range reorderings are observed for the German-English task while a higher number of long-range reorderings. Considering mid-range reorderings (from 5 to 7 words), the French-English pair shows a lower percentage ($\sim 14\%$) than the German-English ($\sim 22\%$). A similar performance is expected when considering the opposite translation directions. Note that reorderings are extracted from word-alignments, an automatic process which is far notoriously error-prone. The above statistics must be accordingly considered.

4.4 Results

Translation accuracy (BLEU) results are given in table 2 for the same *baseline* system performing different reordering models: source 6-gram LM (**sLM**); lexicalized reordering (**lex**); bilingual 6-gram LM (**bLM**) assuming a single POS-tagged version of each tuple. In the case of the German-English translation task we also report results for the bilingual 5-gram LM built from POS tags obtained from *RFTagger* containing a richer vocabulary tag set (**b$^+$LM**). For comparison purposes, we also show the scores obtained by the **Moses** phrase-based system performing lexicalized reordering. Models of both systems are built sharing the same training data and word alignments.

[4]http://www.speech.sri.com/projects/srilm/

[5]http://www.statmt.org/moses/

The worst results are obtained by the **sLM** model. The fact that it only considers source-language information results clearly relevant to accurately model reordering. A very similar performance is shown by our bilingual n-gram system and Moses under lexicalized reordering (**bLM** and **Moses**), slightly lower results are obtained by the n-gram system under French-English translation.

Config	*Fr⇝En*	*En⇝Fr*	*De⇝En*	*En⇝De*
sLM	22.32	21.97	17.11	12.23
lex	22.46	22.09	17.31	12.38
bLM	**23.03**	**22.32**	17.37	12.58
b$^+$LM	–	–	**17.57**	**12.92**
Moses	22.81	**22.33**	17.22	12.45

Table 2: *Translation accuracy (BLEU) results.*

When moving from **lex** to **bLM**, our system increases its accuracy results for both tasks and translation directions. In this case, results are slightly higher than those obtained by Moses (same results for English-to-French). Finally, results for translations performed with the bilingual n-gram reordering model built from rich German POS tags (**b$^+$LM**) achieve the highest accuracy results for both directions of the German-English task. Even though results are consistent for all translation tasks and directions they fall within the statistical confidence margin. Add ± 2.36 to French-English results and ± 1.25 to German-English results for a 95% confidence level. Very similar results were obtained when estimating our model for orders from 5 to 7.

In order to better understand the impact of the proposed reordering model, we have measured the accuracy of the reordering task. Hence, isolating the reordering problem from the more general translation problem. We use BLEU to account the n-gram matching between the sequence of source words aligned to the 1-best translation hypothesis, *i.e.* the permutation of the source words output by the decoder, and the permutation of source words that monotonizes the word alignments with respect to the target reference. Note that in order to obtain the word alignments of the test sets we re-aligned the entire corpus after including the test set. Table 3 shows the BLEU results of the reordering task. Bigram, trigram and 4gram precision scores are also given.

Pair	*Config*	*BLEU (2g/3g/4g)*
Fr⇝En	*lex*	71.69 $(75.0/63.4/55.6)$
	bLM	71.98 $(75.3/63.7/56.0)$
En⇝Fr	*lex*	72.92 $(75.5/65.0/57.6)$
	bLM	73.25 $(75.8/65.4/58.1)$
De⇝En	*lex*	62.12 $(67.3/52.1/42.5)$
	b$^+$LM	63.29 $(68.3/53.5/44.0)$
En⇝De	*lex*	62.72 $(67.9/52.8/43.1)$
	b$^+$LM	63.36 $(68.6/53.6/43.8)$

Table 3: *Reordering accuracy (BLEU) results.*

As can be seen, the bilingual n-gram reordering model shows higher results for both translation tasks and directions than lexicalized reordering, specially for German-English translation. Our model also obtains higher values of n-gram precision for all values of n.

Next, we validate the introduction of additional bilingual context in the translation process. Figure 4 shows the average size of the translation unit n-grams used for the test set according to different models (German-English), the surface form 3-gram language model (main translation model), and the new reordering model when built from the reduced POS tagset (POS) and using the rich POS tagset (POS$^+$).

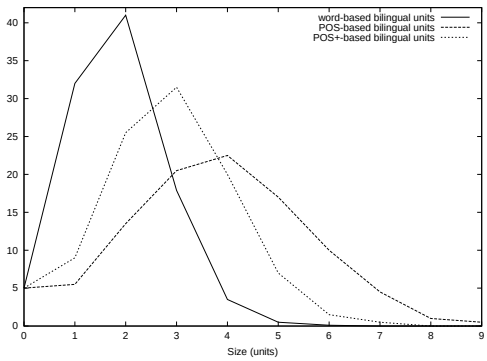


Figure 4: *Size of translation unit n-grams (%) seen in test for different n-gram models.*

As expected, translation units built from the reduced POS tagset are less sparse, enabling us to

introduce larger n-grams in the translation process. However, the fact that they achieve lower translation accuracy scores (see Table 2) indicates that the probabilities associated to these large n-grams are less accurate. It can also be seen that the model built from the rich POS tagset uses a higher number of large n-grams than the language model built from surface forms.

The availability of mid-range n-grams validates the introduction of additional bilingual context achieved by the new model, leading to effectively modeling mid-range reorderings. Notice additionally that considering the language model built from surface forms, only a few 4-grams of the test set are seen in the training set, which explains the small reduction in performance observed when translating with a bilingual 4-gram language model (internal results). Similarly, the results shown in Figure 4 validates the choice of using bilingual 5-grams for b^+LM and 6-grams for bLM.

Finally, we evaluate the mismatch between the reorderings collected on the training data, and those output by the decoder. Table 4 shows the percentage of reordered sequences found for the 1-best translation hypothesis of the test set according to their size. The French-to-English and German-to-English tasks are considered.

Pair	Config	2	3	4	5	6	7	≥ 8
Fr⇝En	*lex*	58	23	10	5	2	1	1
	bLM	57	23	11	4	2.5	1.5	1
De⇝En	*lex*	33	24	22	14	5	1.5	0.5
	b$^+$LM	35	25	19	13	5	2.5	0.5

Table 4: *Size (%) of the reordered sequences observed when translating the test set.*

Very similar distributions are observed for both reordering models. In parallel, distributions are also comparable to those presented in Figure 3 for reorderings collected from the training bi-text, with the exception of long-range and very short-range reorderings. This may be explained by the fact that system models, in special the distortion penalty model, typically prefer monotonic translations, while the system lacks a model to support large-range reorderings.

5 Conclusions and Further Work

We have presented a new reordering model based on bilingual n-grams with units built from linguistic information, aiming at modeling the structural adequacy of translations. We compared our new reordering model to the widely used lexicalized reordering model when implemented in our bilingual n-gram system as well as using *Moses*, a state-of-the-art phrase-based SMT system.

Our model obtained slightly higher translation accuracy (BLEU) results. We also analysed the quality of the reorderings output by our system when performing the new reordering model, which also outperformed the quality of those output by the system performing lexicalized reordering. The back-off procedure used by standard language models allows to dynamically adapt the scope of the context used. Therefore, in the case of our reordering model, back-off allows to consider always as much bilingual context (n-grams) as possible. The new model was straightforward implemented in our bilingual n-gram system by extending the log-linear combination implemented by our decoder. No additional decoding complexity was introduced in the baseline decoding implementation.

Finally, we showed that mid-range reorderings are present in French-English and German-English translations and that our reordering model effectively tackles such reorderings. However, we saw that long-range reorderings, also present in these tasks, are yet to be addressed.

We plan to further investigate the use of different structural information, such as supertags, and tags conveying different levels of morphology information (gender, number, tense, mood, *etc.*) for different language pairs.

Acknowledgments

This work has been partially funded by OSEO under the Quaero program.

References

F. Xia and M. McCord. Improving a Statistical MT System with Automatically Learned Rewrite Patterns. In *Proc. of the COLING 2004*, 508–514, Geneva, Switzerland, August 2004.

D. Chiang. Hierarchical phrase-based translation. *Computational Linguistics*, 33(2):201–228, June 2007.

H. Hoang and Ph. Koehn. Improving Mid-Range Re-Ordering Using Templates of Factors. In *Proc. of the EACL 2009*, 372–379, Athens, Greece, March 2009.

J. M. Crego and J. B. Mariño. Improving statistical MT by coupling reordering and decoding. In *Machine Translation*, 20(3):199—215, July 2007.

Mariño, José and Banchs, Rafael E. and Crego, Josep Maria and de Gispert, Adria and Lambert, Patrick and Fonollosa, J.A.R. and Costa-jussà, Marta N-gram Based Machine Translation. In *Computational Linguistics*, 32(4):527–549, 2006

Ch. Tillman. A Unigram Orientation Model for Statistical Machine Translation. In *Proc. of the HLT-NAACL 2004*, 101–104, Boston, MA, USA, May 2004.

M. Collins, Ph. Koehn and I. Kucerova. Clause Restructuring for Statistical Machine Translation. In Proc. of the ACL 2005, 531–540, Ann Arbor, MI, USA, June 2005.

Ph. Koehn, H. Hoang, A. Birch, Ch. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, Ch. Moran, R. Zens, Ch. Dyer, O. Bojar, A. Constantin and E. Herbst. Moses: Open Source Toolkit for Statistical Machine Translation. In Proc. of the ACL 2007, demonstration session, prague, Czech Republic, June 2007.

Y. Zhang, R. Zens and H. Ney Improved Chunk-level Reordering for Statistical Machine Translation. In Proc. of the IWSLT 2007, 21–28, Trento, Italy, October 2007.

H. Schmid and F. Laws. Estimation of Conditional Probabilities with Decision Trees and an Application to Fine-Grained POS Tagging. In Proc. of the COLING 2008, 777–784, Manchester, UK, August 2008.

F.J. Och and H. Ney. Improved statistical alignment models. In *Proc. of the ACL 2000*, 440–447, Hong Kong, China, October 2000.

Ph. Koehn, A. Axelrod, A. Birch, Ch. Callison-Burch, M. Osborne and D. Talbot. Edinburgh System Description for the 2005 IWSLT Speech Translation Evaluation. In *Proc of the IWSLT 2005*, Pittsburgh, PA, October 2005.

F. J. Och and H. Ney. Discriminative Training and Maximum Entropy Models for Statistical Machine Translation. In *Proc. of the ACL 2002*. 295–302, Philadelphia, PA, July 2002.

A. Stolcke. SRLIM: an extensible language modeling toolkit. *Proc. of the INTERSPEECH 2002*. 901–904, Denver, CO, September 2008.

K. Papineni, S. Roukos, T. Ward, and W.J. Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proc. of the ACL 2002*, 311–318, Philadelphia, PA, July 2002.

R. Kneser and H. Ney. Improved backing-off for m-gram language modeling. In *Proc. of the ICASSP 1995*. 181–184, Detroit, MI, May 1995.

A. Zollmann, A. Venugopal, F. J. Och and J. Ponte. A Systematic Comparison of Phrase-Based, Hierarchical and Syntax-Augmented Statistical MT. In *Proc. of the COLING 2008*. 1145–1152, Manchester, UK, August 2008.

M. Popovic and H. Ney. POS-based Word Reorderings for Statistical Machine Translation. In *Proc. of the LREC 2006*. 1278–1283, Genoa, Italy, May 2006.

J. Niehues and M. Kolss. A POS-Based Model for Long-Range Reorderings in SMT. In *Proc. of the WMT 2009*. 206–214, Athens Greece, March 2009.

Comparing Sanskrit Texts for Critical Editions *

Marc Csernel
Projet AXIS: Inria-Rocquencourt
& Universite Paris-Dauphine
Marc.Csernel@inria.fr

Tristan Cazenave
LAMSADE
Universite Paris-Dauphine,
cazenave@lamsade.dauphine.fr

Abstract

Traditionally Sanskrit is written without blank, sentences can make thousands of characters without any separation. A critical edition takes into account all the different known versions of the same text in order to show the differences between any two distinct versions, in term of words missing, changed or omitted. This paper describes the Sanskrit characteristics that make text comparisons different from other languages, and will present different methods of comparison of Sanskrit texts which can be used for the elaboration of computer assisted critical edition of Sanskrit texts. It describes two sets of methods used to obtain the alignments needed. The first set is using the L.C.S., the second one the global alignment algorithm. One of the methods of the second set uses a classical technique in the field of artificial intelligence, the A* algorithm to obtain the suitable alignment. We conclude by comparing our different results in term of adequacy as well as complexity.

1 Introduction

A critical edition is an edition that takes into account all the different known versions of the same text. If the text is mainly known through a great number of manuscripts that include non trivial differences, the critical edition often looks rather daunting for readers unfamiliar with the subject: the edition is then formed mainly by footnotes that enlighten the differences between manuscripts, while the main text (that of the edition) is rather short, sometimes a few lines on a page. The differences between the texts are usually described in term of words (sometime sentences) missing, added or changed in a specific manuscript. This reminds us the edit distance but in term of words instead of characters. The text of the edition is established by the editor according to his own knowledge of the text. It can be a particular manuscript or a "mean" text built according to some specific criteria. Building a critical edition by comparing texts two by two, especially manuscript ones, is a task which is certainly long and, sometimes, tedious. This is why, for a long time, computer programs have been helping philologists in their work (see O'Hara (1993) or Monroy (2002) for example), but most of them are dedicated to texts written in Latin (sometimes Greek) scripts.

In this paper we will focus on the problems involved by a critical edition of manuscripts written in Sanskrit. Our approach will be illustrated by texts that are extracted from manuscripts of the "Banaras gloss", *kāśikāvṛtti*.

The Banaras gloss was written around the 7th century A.D., and is one of the most famous commentary on the Pāṇini's grammar, which is known as the first **generative** grammar ever written, and was written around the fifth century B.C. as a set of rules. These rules cannot be understood without the explanation provided by a commentary such as the *kāśikāvṛtti*. This collection was chosen, because it is one of the largest collection of Sanskrit manuscripts (about hundred different ones) of the same text actually known.

* This work is supported by the EEC FP7 project IDEAS

Coling 2008: Poster Volume, pages 206–213,
Beijing, August 2010

In what follows we will first describe the characteristics of Sanskrit that matter for text comparison algorithms, we will then show that such a comparison requires the use of a lemmatized text as the main text. The use of a lemmatized text induces the need of a lexical preprocessing. Once the lexical preprocessing is achieved, we can proceed to the comparison, where we develop two kinds of approach, one based on the LCS, which was used to solved this problem, the other one related to sequence alignment. In both cases the results are compared in terms of adequacy as well as complexity. We then conclude and examine the perspective of further work.

2 How to compare Sanskrit manuscripts

One of the main characteristics of Sanskrit is that it is not linked to a specific script. But here we will provide all our examples using the *Devanāgarī* script, which is nowadays the most used. The script has a 48 letters alphabet. Due to the long English presence in India, a tradition of writing Sanskrit with the Latin alphabet (a transliteration) has been established for a long time. These transliteration schemes were originally carried out to be used with traditional printing. It was adapted for computers by Frans Velthuis (Velthuis, 1991), more specifically to be used with TEX. According to the Velthuis transliteration scheme, each Sanskrit letter is written using one, two or three Latin characters; notice that according to most transliteration schemes, upper case and lower case Roman characters have a very different meaning.

In ancient manuscripts, Sanskrit is written without spaces, and this is an important graphical specificity, because it increases greatly the complexity of text comparison algorithms. On the other hand, each critical edition deals with the notion of word. Since electronic Sanskrit lexicons such as the one built by Huet (2006; 2004) do not cope with grammatical texts, we must find a way to identify each Sanskrit word within a character string, without the help of either a lexicon or of spaces to separate the words.

The reader interested in a deeper approach of the Sanskrit characteristics which matters for a computer comparison can look in Csernel and Patte (2009).

The solution comes from the lemmatization of one of the two texts of the comparison: the text of the edition. The lemmatized text is prepared **by hand** by the editor. We call it a *padapāṭha*, according to a mode of recitation where syllables are separated. From this lemmatized text, we will build the text of the edition, that we call a *saṃhitapāṭha*, according to a mode of recitation where the text is said continuously. The transformation of the *padapāṭha* into the *saṃhitapāṭha* is not straightforward because of the existence of *sandhi* rules.

What is called *sandhi* — from the Sanskrit: liaison — is a set of phonetic rules which apply to the morpheme junctions inside a word or to the junction of words in a sentence. These rules are perfectly codified in Pāṇini's grammar. Roughly speaking the Sanskrit reflects (via the *sandhi*) in the writing the liaison(s) which are made by a human speaker. A text with separators (such as spaces) between words, can look rather different (the letter string can change greatly) from a text where no separator is found (see the example of *padapāṭha* on next page).

The processing is done in three steps, but only two of them will be considered in this paper:

- **First step**: The *padapāṭha* is transformed into a virtual *saṃhitapāṭha* in order to make feasible a comparison with a manuscript. The transformation consists in removing all the separations between words and then in applying the *sandhi*. This virtual *saṃhitapāṭha* which will form the text of the edition, is compared with each manuscript. As a sub product of this lexical treatment, the places where the separation between words occur will be kept into a table which will be used in further treatments.

- **Second step:** An alignment of a manuscript and the virtual *saṃhitapāṭha*. We describe three different methods to obtain these alignments. The aim is to identify, as precisely as possible, the words in the manuscript, using the *padapāṭha* as a pattern. Once the words of the manuscript have been determined, we can see through the alignment those which have been added, modified or suppressed.

- **Third step:**: Display the results in a comprehensive way for the editor.

The comparison is done paragraph by paragraph, according to the paragraphs made in the *padapāṭha* during its elaboration by the editor. Each of the obtained alignments, together with the lemmatized text (i.e. *padapāṭha*), suggests an identification of the words of the manuscript.

3 The lexical preprocessing

The goal of this step is to transform both the *padapāṭha* and the manuscript in order to make them comparable. This treatment will mainly consist in transforming the *padapāṭha* into a *saṃhitapāṭha* by applying the *sandhi*.

At the end of the lexical treatment the texts are transmitted to the comparison module in an internal encoding.

This allows us to ensure the comparison whatever the text encoding.

An example of *padapāṭha*:

```
vi^ud^panna_ruupa_siddhi+v.rtti+iya.mkaa"sikaa_naama
```

We can see that words are separated by three different lemmatization signs: +, _, ^ which indicate respectively the presence of an inflected item, the component of a compound word, the presence of a prefix.

The previous *padapāṭha* becomes the following *saṃhitapāṭha*:

```
vyutpannaruupasiddhirv.rttiriya.mkaa"sikaanaama
```

after the transformation induced by the lexical pre-processing, the bold letters represent the letters (and the lemmatization signs) which have been transformed.

Notice that we were induced (for homogeneity reasons) to remove all the spaces from the manuscript before the comparison process. Thus no word of the manuscript can appear separately during that process.

The *sandhi* are perfectly determined by the Sanskrit grammar (see for example Renou (1996)). They induce a special kind of difficulties due to the fact that their construction can be, in certain cases, a two-step process. During the first step, a *sandhi* induces the introduction of a new letter (or a letter sequence). This new letter can induce, in the second step, the construction of another *sandhi*.

```
1d0                 Word 1 'tasmai' is :
< tasmai            - Missing
4c3,5               Word 2 '"srii' is :
< gurave            - Followed by
---                 Added word(s)
> gane             'ga.ne"saaya'
> "                 Word 3 'gurave' is :
> saaya             - Missing
```

`Ediff` with spaces L.C.S. based results without space

Table 1: different comparisons

4 The first trials

The very first trials on Sanskrit critical edition were conducted by Csernel and Patte (2009). Their first idea was to use `diff` (Myers (1986)) in order to obtain the differences between two Sanskrit sequences.

But they find the result quite disappointing. The classical `diff` command line provided no useful information at all.

They obtained a slightly better result with `Emacs ediff`, as shown in Table 1, left column: we can see which words are different. But as soon as they wanted to compare the same sequences without blank, they could not get a better result using `ediff` than using `diff`. This is why they started to implement an L.C.S. (Hirschberg, 1975) based algorithm. Its results appear in the right column of Table 1.

4.1 The L.C.S based algorithm

The L.C.S matrix associated with the previous result can be seen on figure 1 on next page.

On this figure the vertical text represents the *saṃhitapāṭha*, the horizontal text is associated with a manuscript. The horizontal bold dark lines have been provided by the *padapāṭha*, before it has been transformed into the *saṃhitapāṭha*.

The rectangles indicate how the correspondences have been done between the *saṃhitapāṭha* and the manuscript. One corresponds to a word missing (`tasmai`), two correspond to a word present in both strings: the words `s"rii` and `nama.h`, the last one corresponds to a word with a more ambiguous status, we can say either that

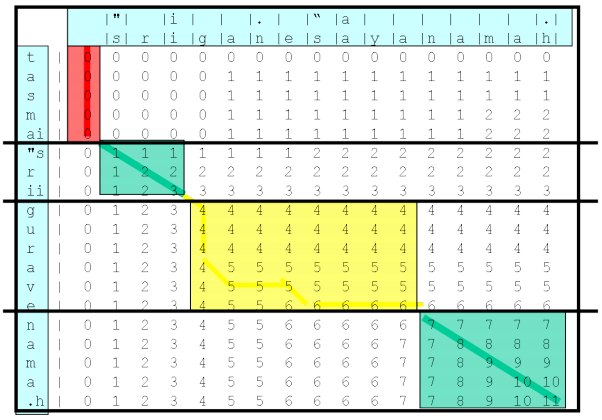


Figure 1: The L.C.S. Matrix

the word has been replaced or that one word is missing and another word has been added. We can see below the result in term of alignment where the double "|" represents a separation between two words.

t	a	s	m	ai		"s	r	ii		g	u	r	a	v	e	-	-	-	-		n	a	m	a	.h
-	-	-	-	-		"s	r	ii		g	-	-	a	.n	e	"s	aa	y	a		n	a	m	a	.h

the corresponding alignment

If the result appears quite obvious within this example, it is not always so easy, particularly when different paths within the matrix can lead to different alignments providing different results.

This induced them to put a lot of post treatments to improve their results, and, at the end, the method looked rather complicated. This is why we were induced to produce an aligment method based on the edit distance.

5 Alignment based on edit distance

We used two different methods to get the alignments formed by the matrix: the first one, based on the common sense, is the subject of this section. The second one, based on the IDA* algorithm is the subject of the next one.

The idea is to get anyone of the alignments between the *saṃhitapāṭha* and the *manuscript*, from the distance matrix, and then apply some simple transformations to get the right one.

The first goal is to minimize the number of incomplete words which appear in the alignment (mostly in the *manuscript*). The second goal is to improve the compactness of each letter sequence by moving in the same word the letters apart from the gaps.

In the following we consider that the distance matrix has been built from the top left to the bottom right, and that the alignment is built by keeping a path from the bottom right till the top left of the matrix.

In such case, if some words are missing in the *manuscript*, some letters can be misaligned (not with the proper word), but this misalignment can be easily corrected by shifting the orphan letters till the correct matching word.

5.1 Shifting the orphan letters

We will call an orphan letter a letter belonging to an incomplete word of the *manuscript* (generally) and being isolated. To obtain a proper alignment these letters must fit with the words to which they belong.

The sequence Seq 1 below gives a good example. The upper line of the table represents the *padapāṭha*, the second one the *manuscript*. In this table, the words `pratyaahaaraa` and `rtha.h` are missing in the *manuscript*. Consequently the letters `a.h` are misplaced, with the word `rtha.h`. The goal is to shift them to the right place with the word `upade"sa.h`. The result after shifting the letters appears in the sequence Seq 2 .

u	p	a	d	e	"s	a	.h	p	r	a	t	y	aa	h	aa	r	aa	r	th	a	.h
u	p	a	d	e	"s	-	-	-	-	-	-	-	-	-	-	-	-	-	-	a	.h

Seq 1

u	p	a	d	e	"s	a	.h	p	r	a	t	y	aa	h	aa	r	aa	r	th	a	.h
u	p	a	d	e	"s	a	.h	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Seq 2

On the second example (Seq 3 & 4) we see on the left side of the table that the letter `a` must just be shifted from the beginning of `asiddhy` to the end of `saavarny` giving Seq 4.

s	aa	v	a	r	.n	y	a	p	r	a	s	y	d	dh	y
s	aa	v	a	r	.n	y	-	-	-	a	s	y	d	dh	y

Seq 3: the orphan letter

s	aa	v	a	r	.n	y	a	p	r	a	s	y	d	dh	y
s	aa	v	a	r	.n	y	a	-	-	-	s	y	d	dh	y

Seq 4: once shifted

But another kind of possible shift is the one linked to the presence of supplementary letters within the *manuscript* such as in Seq 5. The letters `a` and `nam` of the *padapāṭha* are shifted to the right end of the sequence `prayoj` such as shown in Seq 6.

p	r	a	y	o	j	-	-	-	-	a	-	-	-	-	n	a	m
p	r	a	y	o	j	a	n	a	m	s	a	.m	j	"n	aa	n	a	m

Seq 5: before shifting

	p	r	a	y	o	j	a	n	a	m	-	-	-	-	-	-	-	-
	p	r	a	y	o	j	a	n	a	m	s	a	.m	j	"n	aa	n	a	m

Seq 6: once shifted

5.2 The results

The results of the program are first displayed as a text file. They do not come directly from the alignment but from a further treatment, which eliminates some of the useless differences discovered, and transform the other ones into something more convenient for a human reader.

```
Paragraph 3 is Missing in File Asb2
 Word 11 'saara' is:
 - Substituted with 'saadhu' in Man. aa
 Word 17 'viv.rta' is:
 - Followed by Added word(s) 'grantha"saa'
in Manuscript A3
 Word 21 'viudpanna' is:
 - Substituted with 'vyutpannaa' in Man. A3
(P3) Word 32 'k.rtyam' is:
 - Substituted with 'karyam' in
Manuscript A3
 - Substituted with 'kaaryam' in
Manuscripts aa, am4, ba2
```

Such a result, if not fully perfect, has been validated as a correct base for further ameliorations.

6 Using A* for critical edition

In this section we explain the application of A* (Hart et al., 1968; Ikeda and Imai, 1994) to critical edition. We start defining a position for the problem, then we explain the cost function we have used and the admissible heuristic. We end with the search algorithm.

6.1 Positions

A position is a couple of indexes (`x,y`) that represents a position in the dynamic programming matrix. The starting position is at the bottom right of the matrix. The goal position is at the upper left of the matrix (0,0). There are at most three successors of a position: the upper position (`x,y-1`), the position on the left (`x-1,y`) and the position at the upper left (`x-1,y-1`).

Moving to the position at the upper left means aligning two characters in the sequences. Moving up means aligning a gap in the horizontal sequence with a letter in the vertical sequence. Moving to the left means aligning a gap in the vertical sequence with a letter in the horizontal sequence.

6.2 A cost function for the critical edition

It appeared at the end of the first trials of Csernel and Patte (2009) that we can consider the most important criteria concerning the text alignment to be an alignment concerning as few words as possible, and as a secondary criteria the highest possible compactness.

It can be formalized by a cost function which will contain

- the edit distance between the two strings.
- the number of sequences of gaps.
- the number of words in the *manuscript* containing at least a gap.

6.3 The admissible heuristic

We can observe that the edit distance contained in the dynamic programming matrix is always smaller than the score function we want to minimize since the score function is the edit distance increased by the number of gap sequences and the number of words containing gaps.

At any node in the tree, the minimum cost path that goes through that node will be greater than the cost of the path to the node (the `g` value) increased by the edit distance.

The edit distance contained in the dynamic programming matrix is an admissible heuristic for our problem.

6.4 The search algorithm

The search algorithm is the adaptation of IDA* (Korf, 1985) to the critical edition problem. It takes 7 parameters: `g` the cost of the path to the node, `y` and `x` the coordinates of the current position in the matrix, and four booleans that tell if a gap has already been seen in the same word of the *padapāṭha*, if a gap has already been seen in the same word of the *manuscript*, if the previous move is a gap in the *manuscript* or a move in the *padapāṭha*.

The search is successful if it has reached the upper left of the matrix (`x = 0` and `y = 0`, lines 3 and 4 of the pseudo code), and it fails if the minimal cost of the path going through the current node is greater than the threshold (lines 5-6). The search is also stopped if the position has already been searched during the same iteration, with the same threshold and a less or equal `g` (lines 7-8).

In other cases recursive calls are performed (lines 15, 22, 36 and 43).

The first case deals with the insertion of a gap in the *padapāṭha* (possible if `x` is strictly positive, lines 11-16). If this is the first gap in the word we do not add anything to the cost, since we don't care about the number of words containing gaps in the *padapāṭha*, if the previous move is not a gap in the *padapāṭha* then we add one to the cost (line 14) and the recursive call is made with a cost of $g + deltag + 1$ since inserting a gap also costs one.

The second case deals with alignment of the same letters (lines 17-23). In that case the recursive call is performed with the same `g` since it costs zero to align the same letters and that no gap is inserted.

The third case deals with the insertion of a gap in the *manuscript* (possible if y is strictly positive, lines 24-37). Then the cost is increased by one for the first gap in the word (line 28), by one for the first gap of a sequence of gaps (line 32), and by one since a gap is inserted.

The fourth case deals with the alignment of two different letters and increases the cost by one since aligning two different letters costs one and no gap is inserted (lines 38-45).

The pseudo code for the search algorithm is:

```
1  bool search (g, y, x, gapAlreadySeen,
                gapInMat,
2               previousIsGapInMat,
                previousIsGapInPad)
3    if y=0 and x=0
4     return true

5    if g + h(y,x) > threshold
6      return false

7    if position already searched with smaller g
8      return false

9    newSeen = gapAlreadySeen
10   newSeenMat = gapInMat

11   if x > 0
12     deltag = 0
13     if not previousIsGapInPad
         // cost of a sequence of gaps
         // in the Padapatha
14       deltag = deltag + 1
15     if search (g+deltag+1, y, x-1,
             true, gapInMat, false, true)
16       return true

17   if y > 0 and x > 0
18     if alignment of the same letters
19       if new word in the Padapatha
20         newSeen = false
21         newSeenMat = false
22       if search (g, y-1 , x-1, newSeen,
                  newSeenMat, false, false)
23         return true

24   if y > 0
25     deltag = 0;
26     if not gapInMat
27       // cost of each word containing
         // gaps in the Matrikapatha
28       deltag = 1
29       newSeenMat = true
30     if not previousIsGapInMat
31       // cost of a sequence of gaps in
         // the Matrikapatha
32       deltag = deltag + 1
33     if new word in the Padapatha
34       newSeen = false;
35       newSeenMat = false;
36     if search (g+deltag+1, y-1, x,
         newSeen, newSeenMat, true, false)
37       return true;

38   if y>0 and x>0
39     if alignment of different letters
40       if new word in the Padapatha
41         newSeen = false
42         newSeenMat = false
43       if search (g+1, y-1 , x-1, newSeen,
                    newSeenMat, false, false)
44         return true

45   return false
```

The search function is bounded by a threshold on the cost of the path. In order to find the shortest path, an iterative loop progressively increasing the cost is used.

7 Experiments and Conclusions

We have tested on our Sanskrit texts three different methods to align them: one based upon the L.C.S., the two other ones based on the edit distance. We have tested them on a set of 43 different manuscripts of a short text, the introduction of the *kāśikāvṛtti*: the *pratyāhārasūtraḥ*. A critical edition of this text exists (Bhate et al., 2009), and we have not seen obvious differences with our results.

The size of the *padapāṭha* related to this text is approximately 9500 characters. The time needed for the treatment is approximately 29 seconds for the L.C.S based one, 22 for the second method (with the shifts) and 185 seconds for the third one based on the IDA*algorithm (all mesured on a Pentium 4 (3.2mgz)).

The comparison between the first method and

the two others cannot be absolute, because the first one displays its results under a more synthetic form, and cannot display only the alignments. This form takes a little more time to be proceeded but less time to be written.

Comparing the different methods:

- The first trial (L.C.S.) was a very useful one, because it allows displaying significant results to Sanskrit philologists, and opens the possibility of further research. But it is too complicated compared with other approaches, and the different steps needed, though useful, do not provide the opportunity to make easily further improvements.

- The second approach gives the best results in term of time. It is conceptually quite simple, and not too difficult to implement in term of programming. And it gives place, because it has been simple to implement, for further improvements.

- What can we say then about the IDA* method, which is by far the longest to make the computation? That it is unmistakably not the best choice as a production method when computation time is a preoccupation (but the time overhead has nothing definitive), but it is for sure, for the person "who knows" the most flexible, and the easiest way to implement alignment methods, and to check an hypothesis. Using A* would probabbly be faster as the branching factor is small.

The use of edit distance based methods has been, by the simplifications and the ameliorations it provide for the comparison of the Sanskrit text a great improvement. Both methods will allow us to consider different coefficients for replacing the letters in the edit distance matrix and leads to further simplification of the pre-processing. The IDA* (or other A*) method, opens wide the doors for further experiments. Among these experiments one of the most interesting will consist in the modelling of an interaction between the information provided by the annotations contained in each manuscript (especially the presence of missing parts of the text) and the alignment.

It is difficult to provide a numerical evaluation of the different results, first because they are not provided under the same form, the first method is provided as a human readable text and the two other ones as sequence alignments, secondly because it is difficult (and we did not find it) to provide a criterion which differs from the function we optimize in the A* algorithm. Otherwise even if the differences between the two methods are rather tiny, the A* algorithm which optimizes by construction the criterion will be considered always as slightly better.

Another possible improvement is related to the fact that in Sanskrit, the order of the words is not necessary meaningful. Two sentences with the words appearing in two different orders can have the same meaning.

But there is a problem that none of these methods can solve, the problem induced by the absence of a word which has been used to build a *sandhi*. Once it disappeared the *sandhi* disappeared too, and a new *sandhi* can appear, then it looks like a real change of the text, but these modifications are perfectly justified in term of Sanskrit grammar and should not be notified in the critical edition. For example if we look at the following sequence:

"s	aa	s	t	r	a	p	r	a	v	.r	t	t	y	a	r	th	a	.h
"s	aa	s	t	r	aa	-	-	-	-	-	-	-	-	-	r	th	a	.h

- the word `"saastra` has been changed in `"saastr`**`aa`** (with a long `a` at the end).

- the word `prav.rtty` has disappeared.

- the word `artha.h` has been changed to `rtha.h`

In fact only the second point is valid. If we put the words `"saastra` and `artha.h` one after another in a Sanskrit text we get `"saastr`**`aa`**`rtha.h`. The two short `a` at the junction of the two words become a long `aa` (in bold) because of a *sandhi* rule. We have (until now) no precise idea on the way to solve this kind of problem, but we have the deep feeling that the answer will not be straightforward.

On the other hand we believe that the problems induced by the comparison of Sanskrit texts for

the construction of a critical edition, is an interesting family of problems. We hope that the solutions of these problems can be applied to other languages, and perhaps that it will also benefit to some other problems.

References

Bhate, Saroja, Pascale Haag, and Vincenzo Vergiani. 2009. The critical edition. In Haag, Pascale and Vincenzo Vergiani, editors, *Studies in the kāśikāvṛtti The section on Pratyāhāras*. Societa Editrice Fiorentina.

Csernel, Marc and François Patte. 2009. Critical edition of sanskrit texts. In *Sanskrit Computational Linguistics*, volume 5402 of *Lecture Notes in Computer Science*, pages 358–379.

Hart, P., N. Nilsson, and B. Raphael. 1968. A formal basis for the heuristic determination of minimum cost paths. *IEEE Trans. Syst. Sci. Cybernet.*, 4(2):100–107.

Hirschberg, D.S. 1975. A linear space algorithm for computing maximal common subsequences. *CACM*, 18(6):341–343.

Huet, Gerard. 2004. Design of a lexical database for sanskrit. In *COLING Workshop on Electronic Dictionaries*, pages 8–14, Geneva.

Huet, Gerard. 2006. Héritage du sanskrit: Dictionnaire français-sanskrit. http://sanskrit.inria.fr/Dico.pd.

Ikeda, T. and T. Imai. 1994. Fast A* algorithms for multiple sequence alignment. In *Genome Informatics Workshop 94*, pages 90–99.

Korf, R. E. 1985. Depth-first iterative-deepening: an optimal admissible tree search. *Artificial Intelligence*, 27(1):97–109.

Monroy, C. et al. 2002. Visualization of variants in textual collations to analyse the evolution of literary works in the cervantes project. In *Proceedings of the 6th European Conference, ECDL 2002*, pages 638–53, Rome, Italy.

Myers, E.W. 1986. An O(ND) difference algorithm and its variations. *Algorithmica*, 1(2):251–266.

O'Hara, R.J. Robinson, P.M.W. 1993. Computer-assisted methods of stemmatic analysis. In Blake, Norman and Peter Robinson, editors, *Occasional Papers of the Canterbury Tales Project*, volume 1, pages 53–74. Office for Humanities Communication, Oxford University.

Renou, Louis. 1996. *Grammaire sanskrite: phonétique, composition, dérivation, le nom, le verbe, la phrase*. Maisonneuve, Paris. (réimpression).

Velthuis, F., 1991. *Devanāgarī for TEX, Version 1.2, User Manual*. http://www.ctan.org/tex-archive/language/devanagari/velthuis/.

Hybrid Decoding: Decoding with Partial Hypotheses Combination over Multiple SMT Systems*

Lei Cui[†], **Dongdong Zhang**[‡], **Mu Li**[‡], **Ming Zhou**[‡], and **Tiejun Zhao**[†]

[†]School of Computer Science and Technology
Harbin Institute of Technology
{cuilei,tjzhao}@mtlab.hit.edu.cn
[‡]Microsoft Research Asia
{dozhang,muli,mingzhou}@microsoft.com

Abstract

In this paper, we present *hybrid decoding* — a novel statistical machine translation (SMT) decoding paradigm using multiple SMT systems. In our work, in addition to component SMT systems, system combination method is also employed in generating partial translation hypotheses throughout the decoding process, in which smaller hypotheses generated by each component decoder and hypotheses combination are used in the following decoding steps to generate larger hypotheses. Experimental results on NIST evaluation data sets for Chinese-to-English machine translation (MT) task show that our method can not only achieve significant improvements over individual decoders, but also bring substantial gains compared with a state-of-the-art word-level system combination method.

1 Introduction

In recent years, system combination for SMT has been known to be quite effective with translation consensus information built from multiple SMT systems. The combination approaches can be classified into two types. One is the combination with each system's outputs, which can be seen as full hypotheses combination. The other is the partial hypotheses (PHS) combination during the decoding phase.

A lot of impressive work has been done to improve the performance of the SMT systems by utilizing consensus statistics which come from single system or multiple systems. For example, Minimum Bayes Risk (MBR) (Kumar and Byrne, 2004) decoding over n-best list finds a translation that has lowest expected loss with all the other hypotheses, and it shows that improvement over the Maximum a Posteriori (MAP) decoding. Several word-based methods (Rosti et al., 2007a; Sim et al., 2007) have also been proposed. Usually, these methods take n-best list from different SMT systems as inputs, and construct a confusion network for second-pass decoding. There are also a lot of research work to advance the confusion network construction by finding better alignment between the skeleton and the other hypotheses (He et al., 2008; Ayan et al., 2008). Typically, all the approaches above only use full hypotheses but have no access to the PHS information.

Moreover, some dedicated efforts have been tried by manipulating PHS between multiple MT systems. Collaborative decoding (co-decoding) (Li et al., 2009) leverages translation consensus by exchanging partial translation results and reranking both full and partial hypotheses explored in decoding. However, no new PHS are generated compared to the individual decoding but only the ranking is affected. Liu et al. (2009) proposes joint decoding, a method that integrates multiple translation models in one decoder. Although joint decoding is able to generate new translations compared to single decoder, it has to use the PHS existed in one of its component decoder at each step. Different from their work, we propose a new perspective which leverages outputs from local word-level combination. This will potentially bring much benefit of performance since word-

*This work has been done while the first author was visiting Microsoft Research Asia.

Coling 2008: Poster Volume, pages 214–222,
Beijing, August 2010

level combination can produce more promising PHS.

The word-level system combination method is employed to generate partial translation hypotheses in our hybrid decoding framework. In this sense, full hypotheses word-level combination (FH-Comb) method(Rosti et al., 2007a; Sim et al., 2007; He et al., 2008; Ayan et al., 2008) can be considered as a special case of hybrid decoding, where their combinations are only performed on the largest hypotheses. Similar with FH-Comb, hybrid decoding also uses word alignment information. However, challenge exists in hybrid decoding as word alignment needs to be carefully conducted through the decoding process. Obviously, document-level word alignment methods such as GIZA++(Och and Ney, 2000) are quite time consuming and unpractical to be embedded into hybrid decoding. We propose a heuristic method that can conduct word alignment of partial hypotheses based on word alignment information of phrase pairs learnt automatically from the model training process. In this way, more PHS are generated and the search space is enlarged substantially, which brings better translation results.

The rest of the paper is organized as follows: Section 2 gives a formal description of hybrid decoding, including framework overview, word-level PHS combination and parameter estimation. We conduct experiments with different settings and make comparison between our method and baseline, as well as a state-of-the-art word-level system combination method in Section 3. Experimental results discussion is presented in Section 4. Section 5 concludes the paper.

2 Hybrid Decoding

2.1 Overview

Different system combination methods (Li et al., 2009; Liu et al., 2009) offer different frameworks to coordinate multiple SMT decoders. Hybrid decoding provides a new scheme to organize multiple decoders to work synchronously. As the decoding algorithms may differ in multiple decoders[1], hybrid decoding has some difficulty in integrating different decoding algorithms. Without loss of generality, we assume that bottom-up CKY-based decoding is adopted in each individual decoder, which is the same as co-decoding (Li et al., 2009) and joint decoding (Liu et al., 2009). Hybrid decoding collects n-best PHS of a source span[2] from multiple decoders, then results from word-level PHS combination of that span are given back to each decoder, mixed with the original PHS. After that, we re-rank the hybrid list and continue the decoding. In an example with two decoders, parts of the whole decoding process are illustrated in Figure 1 and can be summarized as follows:

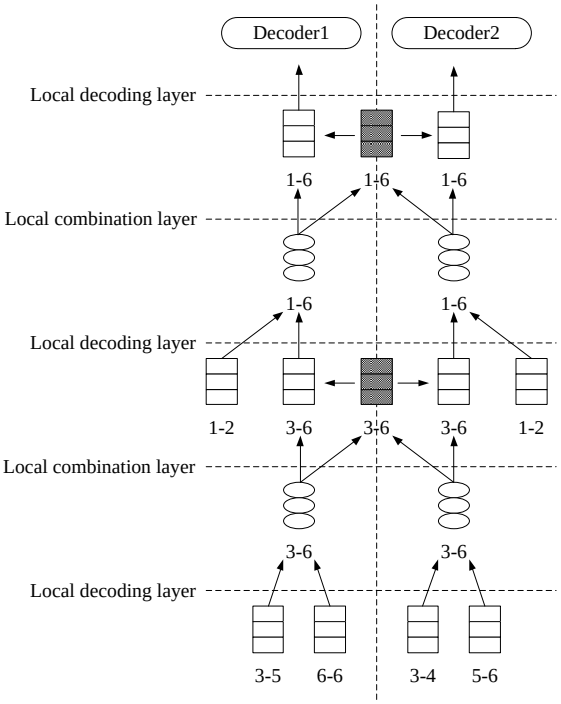


Figure 1: Hybrid decoding with two decoders, where the string "s-e" means the source span starts from position s and ends at position e. The blank rectangles represent the n-best partial translations of each decoder, and the shaded rectangles illustrate the n-best local combination outputs. The ovals denote bottom-up CKY-based decoding results.

[1]In the SMT area, some decoders use left-right decoding to generate the hypothesis and "Pharaoh" (Koehn et al., 2003) is one of them, while others adopt bottom-up decoding which is represented by "Hiero" (Chiang, 2007).

[2]The word "span" is used to represent translation unit in CKY-based decoders, which denotes one or more consecutive words in the source sentence.

1. *Individual decoding*. Each individual decoder should maintain the n-best PHS of each span from the bottom. After all the individual decoders finish translating the same span, they feed their own partial translations into a public container which can be used for word-level PHS combination, then get back the partial combination outputs for step 3.

2. *Local word-level combination*. After fed with PHS from multiple decoders, a confusion network is built and word-level combination for PHS is conducted. The obtained new partial translations are given back to each individual decoder to continue the decoding.

3. *Mix new PHS with the original ones*. The span in each individual decoder will receive the corresponding new PHS from the local combination outputs. The feature space of the new PHS is not exactly the same with that of the original ones. It has to be mapped in some way then the mixed hypotheses are reranked.

In the following sub-sections, we first present the background of word-level combination for PHS, then introduce hybrid decoding algorithm in detail, as well as the feature definition and parameter estimation.

2.2 Word-Level Combination for Partial Hypotheses

Most word-level system combination methods are based on confusion network decoding. In confusion network construction, one hypothesis has to be selected as the skeleton which determines the word order of the combination results. Other hypotheses are aligned against the skeleton. Either votes or some word confidence scores are assigned to each word in the network.

Most of the research on confusion network construction focuses on seeking better word alignment between the skeleton and the other hypotheses. So far, several word alignment procedures are used for SMT system combination, which mainly are GIZA++ alignments (Matusov et al., 2006), TER alignments (Sim et al., 2007) and IHMM alignments (He et al., 2008). Similar with general word-level system combination method, word-level PHS combination also uses word alignment information. However, in hybrid decoding, it is quite time-consuming and impractical to conduct word alignment like GIZA++ for each span. Fortunately, unit hypotheses word alignment can be obtained from the model training process, which is shown in Figure 2. We devise a heuristic approach for PHS alignment that leverages the translation derivations from the sub-phrases. The derivation information ultimately comes from the phrase table in phrase-based systems (Koehn et al., 2003; Xiong et al., 2006) or the rule table in syntactic-based systems (Chiang, 2007; Liu et al., 2007; Galley et al., 2006).

政治||| political ||| 0-0
政治 经济||| political and economic ||| 0-0 1-2
经济||| economic ||| 0-0
经济 利益||| economic interests ||| 0-0 1-1
政治 [X_1] ||| political and [X_1] ||| 0-0 1-2

Figure 2: The example of translation alignment from phrase-table and rule-table

The derivation is built in a phrase-based system as follows. For example, we have two phrase translations “我们 的 ||| our ||| 0-0 1-0” and “经济 利益 ||| economic interests ||| 0-0 1-1”, where string “m-n” means the m^{th} word in the source phrase is aligned to the n^{th} word in the target phrase. When combining the two phrases for generating “我们 的 经济 利益”, we obtain the translation hypothesis as “our economic interests” and also integrate the alignment fragment to get “0-0 1-0 2-1 3-2”. The case is similar in syntactic-based system for non-terminal substitution, which we will not discuss further here.

Next, we introduce the skeleton-to-hypothesis word alignment algorithm in detail. With the translation derivations, the skeleton-to-hypothesis ($sk2hy$) word alignment can be performed based on the source-to-skeleton ($so2sk$) and source-to-hypothesis ($so2hy$) word alignment as they share the same source sentence. The basic idea is to construct the $sk2hy$ word alignment with the *minimum correspondence subsets* (MCS). A MCS is defined as a triple $< SK, HY, SO >$ where the

SK is the subset of skeleton words, HY is the subset of the hypothesis words, and SO is the minimum source word set that all target words in both SK and HY are aligned to. Figure 3 shows the algorithm for skeleton-to-hypothesis alignment. Most of the pseudo-code is self-explained except for some subroutines, which are listed in Table 1.

```
1:  procedure SKEHYPALIGN(so2sk, so2hy)
2:      repeat
3:          Fetch out a source word to SO
4:          SO_1 = SO_2 = SO
5:          repeat
6:              SO=UNION(SO_1, SO_2)
7:              SK=GETALIGN(SO, so2sk)
8:              HY=GETALIGN(SO, so2hy)
9:              SO_1=GETALIGN(SK, so2sk)
10:             SO_2=GETALIGN(HY, so2hy)
11:         until |SO_1| == |SO_2| == |SO|
12:         sim_max = -infinity
13:         for all sk ∈ SK do
14:             for all hy ∈ HY do
15:                 sim =SIM(sk, hy)
16:                 if sim ≥ sim_max then
17:                     sim_max = sim
18:                     sk_max = sk
19:                     hy_max = hy
20:                 end if
21:             end for
22:         end for
23:         ADDALIGN(sk_max, hy_max)
24:     until all the source words are fetched out
25: end procedure
```

Figure 3: Algorithm for skeleton-to-hypothesis alignment

Subroutines	Description
UNION(A,B)	the union of set A and set B
GETALIGN(S,$align$)	get the words aligned to S based on $align$
SIM(w_1,w_2)	similarity between w_1 and w_2, we use edit distance here
ADDALIGN(w_1,w_2)	align w_1 with w_2

Table 1: Description for subroutines

Due to the variety of the word order in n-best outputs, skeleton selection becomes essential in confusion network construction. The simplest way is to use the top-1 PHS from any individual decoder with the best performance under some criteria. However, this cannot always lead to better performance on some evaluation metrics (Rosti et al., 2007a). An alternative would be MBR method with some loss function such as TER (Snover et al., 2006) or BLEU (Papineni et al., 2002). We show the experimental results of two skeleton selection methods for PHS combination in Section 3.

2.3 Hybrid Decoding Model

For a given source sentence f, any individual decoder in hybrid decoding finds the best translation e^* among the possible translation hypotheses $\Phi(f)$ in terms of a ranking function F:

$$e^* = \text{argmax}_{e \in \Phi(f)} F(e) \quad (1)$$

Suppose we have n individual decoders. The ranking function F_n of the n^{th} decoder can be written as:

$$F_n(e) = \sum_{i=1}^{m} \lambda_{n,i} h_{n,i}(f, e) \quad (2)$$

where each $h_{n,i}(f, e)$ is a feature function of the n^{th} decoder, and $\lambda_{n,i}$ is the corresponding feature weight. m is the number of features in each decoder.

The final result of hybrid decoder is the top-1 translation from the confusion network, which is constructed on multiple decoders with the last layer's output of CKY-based decoding.

2.4 Hybrid Decoding Algorithm

The hybrid decoder acts as a control unit which controls the synchronization of multiple individual decoders. The algorithm is fully demonstrated in Figure 4. The hybrid decoder pushes the same span f_i^j to different decoders and gets back the n-best PHS (lines 2-6). When the span's length is too small, both word alignment and partial combination results are not accurate. We predefine a fixed threshold δ which is used for determining the start-up of combination (line 7). When the length condition holds, the n-best PHS of each individual

decoder are stored in container G (lines 8). Confusion network is constructed and new PHS can be extracted from it and are further mixed and sorted with the original ones (lines 11-15).

```
1:  procedure HYBRIDDECODING(f_1^n, D)
2:      for l ← 1...n do
3:          for all i, j s.t. j − i = l do
4:              G ← ∅
5:              for all d ∈ D do
6:                  nbest =DECODING(d, i, j)
7:                  if j − i ≥ δ then
8:                      ADD(G, nbest)
9:                  end if
10:             end for
11:             cn =CONNETBUILD(G)
12:             nbest′ =GETPARHYP(cn)
13:             for all d ∈ D do
14:                 MIXSORT(nbest_d, nbest′)
15:             end for
16:         end for
17:     end for
18: end procedure
```

Figure 4: Hybrid decoding algorithm

2.5 Hybrid Decoding Features

Next we present the PHS word-level combination feature functions for hybrid decoding. Following (Rosti et al., 2007b), four features are utilized to model the PHS as:

Word Confidence Feature $h_{wc}(e)$
The word confidence feature is computed as $h_{wc}(e) = \sum_{i=1}^{n} \mu_i c_{iw}$, where n is the number of the systems, μ_i is the system confidence of system i, and c_{iw} is the word confidence of word w in system i.

Word Penalty Feature $h_{wp}(e)$
Word penalty feature is the number of words in the partial hypothesis (PH).

Null Penalty Feature $h_{np}(e)$
For null penalty feature, we mean the number of NULL links along the PH when extracted from the confusion network.

Language Model Feature $h_{lm}(e)$
Different from the above three combination features, which can be obtained during the confusion network construction or hypotheses extraction, the language model feature cannot be summed up on the fly. Instead, it must be re-computed when building each new PH.

2.6 Feature Space Mapping

The features used in hybrid decoding can be classified into two categories: features for individual decoders (FID) and features for PHS word-level combination (FComb), and they are independent. When mixing the new PHS with the original ones of individual decoders, FComb space has to be mapped to a FID space. However, several features in FID are not defined in FComb, such as source to target (S2T) phrase probability, target to source (T2S) phrase probability, S2T lexical probability, T2S lexical probability and other model specific features. A mapping function H needs to be defined as follows:

$$F_{fid} = H(F_{fcomb}) \tag{3}$$

where F_{fcomb} denotes the feature vector from FComb space, while F_{fid} is the feature vector from FID space.

An easy mapping function is implemented with an intuitive motivation: PHS combination results are better than the ones in individual decoder and we prefer not to disorder the original search space. Thus, the undefined feature values of PHS from FComb space are assigned by corresponding feature values of the top-1 PH in original decoder. Experiments show that our method is not only practical but also quite effective.

2.7 Parameter Estimation

Minimum Error Rate Training (MERT) (Och, 2003) algorithm is adopted to estimate feature weights for hybrid decoding. As hybrid decoder makes use of PHS from both individual decoders and combination results as a whole, we devise a new feature vector representation. The feature vectors from FID space and FComb space are simply concatenated to form a longer vector without overlapping. The weights are tuned simultaneously in order to reach a relatively global optima.

3 Experiment

3.1 Data and Metric

We conducted our experiments on the test data of NIST 2005 and NIST 2006 Chinese-to-English machine translation tasks. The NIST 2003 test data is used as the development data to tune the parameters. Statistics of the data sets are shown in Table 2. Translation performances are measured with case-insensitive BLEU4 score (Papineni et al., 2002). Statistical significance test is performed using the bootstrap re-sampling method proposed by Koehn (2004).

The bilingual training corpora we used are listed in Table 3, which contains 498K sentence pairs, 12.1M Chinese words and 13.8M English words after pre-processing. Word alignment is performed by GIZA++ (Och and Ney, 2000) in both directions with the default setting, and the intersect-diag-grow method is used to refine the symmetric word alignment.

Data Set	# Sentences
NIST 2003(dev)	919
NIST 2005(test)	1,082
NIST 2006(test)	1,664

Table 2: Statistics of test/dev data sets

LDC ID	Description
LDC2003E07	Ch/En Treebank Par Corpus
LDC2003E14	FBIS Multilanguage Texts
LDC2005T06	Ch News Translation Text Part 1
LDC2005T10	Ch/En News Magazine Par Text
LDC2005E83	GALE Y1 Q1 Release - Translations
LDC2006E26	GALE Y1 - En/Ch Par Financial News
LDC2006E34	GALE Y1 Q2 Release - Translations V2.0
LDC2006E85	GALE Y1 Q3 Release - Translations
LDC2006E92	GALE Y1 Q4 Release - Translations

Table 3: Training corpora for Chinese-English translation

The language model used for hybrid decoding and all the baseline systems is a 5-gram model trained with the Xinhua portion of LDC English Gigaword Version 3.0 plus the English part of bilingual training data.

3.2 Implementation

We use two baseline systems. The first one (SYS1) is re-implementation of Hiero, a hierarchical phrase-based system (Chiang, 2007) based on Synchronous Context Free Grammar (SCFG). Phrasal translation rules and hierarchical translation rules with nonterminals are extracted from all the bilingual sentence pairs. The second one (SYS2) is a phrase-based system (Xiong et al., 2006) based on Bracketing Transduction Grammar (Wu, 1997) with a lexicalized reordering model (Zhang et al., 2007) under maximum entropy framework, where the phrasal translation rules are exactly the same with that of SYS1. The lexicalized reordering model is trained using the MaxEnt toolkit (Zhang, 2006) where the training instances are extracted from subset of the training corpora, which contains LDC2003E07, LDC2003E14, LDC2005T06, LDC2005T10. Both systems use the bottom-up CKY-based decoding with cube-pruning (Chiang, 2007) and the beam size is set to 10 for decoding efficiency.

For hybrid decoder, we set δ to be $sentence.length - 3$, meaning that the PHS of individual decoders only perform local combination in the last three layers. The reason why we adopt this setting is because we find that starting local combination on short spans hurts the performance badly on test data. Experimental results are shown in the next section.

3.3 Translation Results

We present the overall results of hybrid decoding over two baseline systems on both test sets. We also implement an IHMM-based word-level system combination method (He et al., 2008) to make comparison with hybrid decoding system, and the n-best candidates used for IHMM-based word-level system combination is set to 10. Parameters for all the systems are tuned on NIST 2003 test set. The results are shown in Table 4.

In Table 4, we find that the hybrid decoding performs significantly better than SYS1 and SY2 on both test sets. Besides, compared to IHMM word-level system combination method, hybrid decoding also brings substantial gains with 0.63% and 0.92% points respectively.

	NIST 2005	NIST 2006
SYS1	0.3745	0.3346
SYS2	0.3699	0.3296
IHMM Word-Comb	0.3821^{*}	0.3421^{*}
Hybrid	0.3884^{*+}	0.3513^{*+}

Table 4: Hybrid decoding results on test sets, *:significantly better than SYS1 and SYS2 with $\boldsymbol{p}<0.01$, +:significantly better than IHMM Word-Comb with $\boldsymbol{p}<0.01$

We also try different layers for determining the start-up of local word-level PHS combination. Figure 5 gives the intuitive BLEU results.

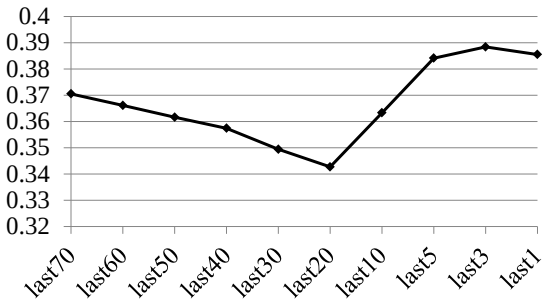


Figure 5: Performance of hybrid decoding with different start-up settings on NIST 2005 test set, where the "lastM" means to conduct local word-level PHS combination in the last M layers from the perspective of CKY-based decoding.

As shown in Figure 5, the performance drops drastically if we start to conduct word-level PHS combination too early. After considering about efficiency and performance, we determine to do that in the last three layers.

We then investigate the effects on hybrid decoding with different beam sizes, and compare the trend with two baseline systems and IHMM-based word-level system combination method as well. The results are illustrated in Figure 6.

From what we see in Figure 6, the performance of each system is monotonically increasing as the beam size becomes larger. Hybrid decoding performs consistently better than IHMM Word-Comb when the beam size is small, and the largest improvement (+0.63% points) is obtained when the beam size is set to 10. However, as the beam size

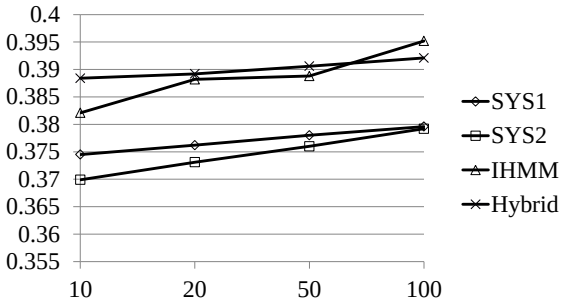


Figure 6: Performance of hybrid decoding with different beam sizes on NIST 2005 test set

increases, the performance gap is getting narrow. One intuitive observation is that hybrid decoding performs slightly worse than IHMM Word-Comb when the beam size is set to 100. One possible reason for this phenomenon is that, the alignment noise may be introduced to hybrid decoding since we have to generate monolingual alignments with many poor translation derivations.

The confusion network for PHS of each system can be built independently. We would like to evaluate the performance of single system hybrid decoding. Table 5 gives the results on both Hiero and BTG decoders.

	NIST 2005		NIST 2006	
	SYS1	SYS2	SYS1	SYS2
baseline	0.3745	0.3699	0.3346	0.3296
self-comb	0.3770	0.3758^{*}	0.3358	0.3355^{*}

Table 5: Hybrid decoding for single system, *:significantly better than baseline with $\boldsymbol{p}<0.05$

Table 5 shows that BTG decoder (SYS2) has more potential for so-called "self-boosting". The self-combination of BTG decoder improves the performance substantially over the baseline. However, we did not observe any significant improvement for Hiero decoder (SYS1).

Finally, we examine the impacts of skeleton selection for PHS in hybrid decoding. The results in Table 6 demonstrate that, compared to the top-1 selection method, translation performance can be improved significantly with MBR-based skeleton selection method. It strongly suggests that choosing the skeleton with more consistent word order

will lead to better translation results.

	NIST 2005	NIST 2006
Top-1	0.3817	0.3415
MBR	0.3884*	0.3513*

Table 6: Skeleton selection in hybrid decoding, *:significantly better than top-1 skeleton selection method with $p<0.01$

4 Discussion

System combination methods have been widely used in SMT to improve the performance. For example, in (Rosti et al., 2007a), several combination methods have been proposed to make use of different kinds of consensus information. In (He et al., 2008), better word alignment method is adopted to advance the word-level system combination. Our method is different from these methods in the sense that we do not exclusively rely on the n-best full hypotheses from each individual decoder, but emphasize the importance of word-level combination for PHS. Thus, it enlarges the search space and is more prone to find better translations. Experimental results have shown the effectiveness of our method.

The idea of multiple systems collaborative decoding (Li et al., 2009) works well on re-ranking the outputs of each system using n-gram agreement statistics. However, no new translation results are generated compared to individual decoding. Our method takes advantage of confusion network to give PHS which cannot be seen before.

Although (Liu et al., 2009) also work on PHS, we explore the cooperation of multiple systems from a new perspective. They use translation derivations from different decoders jointly as a bridge to connect different models. Different from their work, we devise a heuristic method to obtain word alignment information from the derivation of each decoder, which can be embedded for word-level PHS combination easily and efficiently.

5 Conclusion and Future Work

In this paper, we propose a new SMT decoding framework named hybrid decoding, in which multiple decoders work synchronously to conduct local decoding and local word-level PHS combination in turn. We also devise a heuristic method to obtain word alignment information directly from the translation derivations, which is both intuitive and efficient. Experimental results show that with hybrid decoding the overall performance can be improved significantly over both the individual baseline decoder and the state-of-the-art system combination method.

In the future, we will involve more individual SMT decoders into hybrid decoding. In addition, we would like to keep on this work in two directions. On the one hand, start-up threshold of PHS combination will be explored in detail to find its underlying impact on hybrid decoding. On the other hand, we will try to employ a more theoretically sound approach to conduct the feature space mapping from the feature space of confusion network to that of individual decoders.

References

Ayan, Necip Fazil, Jing Zheng, and Wen Wang. 2008. *Improving alignments for better confusion networks for combining machine translation systems*. In *Proceedings of the 22nd International Conference on Computational Linguistics*, pages 33-40

Chiang, David. 2005. *A hierarchical phrase-based model for statistical machine translation*. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 263-270

Chiang, David. 2007. *Hierarchical phrase-based translation*. *Computational Linguistics*, 33(2): pages 201-228

Galley, Michel, Jonathan Graehl, Kevin Knight, Daniel Marcu, Steve DeNeefe, Wei Wang, and Ignacio Thayer. 2006. *Scalable inference and training of context-rich syntactic translation models*. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 961-968

He, Xiaodong, Mei Yang, Jianfeng Gao, Patrick Nguyen, and Robert Moore. 2008. *Indirect-hmm-based hypothesis for combining outputs from machine translation systems*. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 98-107

Koehn, Phillip, Franz J. Och, and Daniel Marcu. 2003. *Statistical phrase-based translation*. In *Proceed-*

ings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics, pages 48-54

Koehn, Phillip. 2004. *Statistical significance tests for machine translation evaluation*. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 388-395

Kumar, Shankar and William Byrne. 2004. *Minimum bayes-risk decoding for statistical machine translation*. In *Proceedings of the 2004 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 169-176

Li, Mu, Nan Duan, Dongdong Zhang, Chi-Ho Li, and Ming Zhou. 2009. *Collaborative decoding: partial hypothesis re-ranking using translation consensus between decoders*. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 585-592

Liu, Yang, Yun Huang, Qun Liu, and Shouxun Lin. 2007. *Forest-to-string statistical translation rules*. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 704-711

Liu, Yang, Haitao Mi, Yang Feng, and Qun Liu. 2009. *Joint decoding with multiple translation models*. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 576-584

Matusov, Evgeny, Nicola Ueffing, and Hermann Ney. 2006. *Computing consensus translation from multiple machine translation systems using enhanced hypotheses alignment*. In *11th Conference of the European Chapter of the Association for Computational Linguistics*, pages 33-40

Och, Franz Josef. and Hermann Ney. 2000. *Improved statistical alignment models*. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 440-447

Och, Franz Josef. 2003. *Minimum Error Rate Training in Statistical Machine Translation*. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 160-167

Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. *BLEU: a method for automatic evaluation of machine translation*. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311-318

Rosti, Antti-Veikko, Necip Fazil Ayan, Bing Xiang, Spyros Matsoukas, Richard Schwartz, and Bonnie Dorr. 2007. *Combining outputs from multiple machine translation systems*. In *Proceedings of the 2007 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 228-235

Rosti, Antti-Veikko, Spyros Matsoukas, and Richard Schwartz. 2007. *Improved word-level system combination for machine translation*. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 312-319

Sim, K.C., W. Byrne, M. Gales, H. Sahbi, and P. Woodland. 2007. *Consensus network decoding for statistical machine translation combinnation*. In *32nd IEEE International Conference on Acoustics, Speech, and Signal Processing*

Snover, Matthew, Bonnie Dorr, Richard Schwartz, Linnea Micciula, and John Makhoul. 2006. *A study of translation edit rate with targeted human annotation*. In *the 7th conference of the Association for Machine Translation in the Americas*, pages 223-231

Wu, Dekai. 1997. *Stochastic inversion transduction grammars and bilingual parsing of parallel corpora*. *Computational Linguistics*, 23(3): pages 377-404

Xiong, Deyi, Qun Liu, and Shouxun Lin. 2006. *Maximum entropy based phrase reordering model for statistical machine translation*. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 521-528

Zhang, Dongdong, Mu Li, Chi-Ho Li, Ming Zhou. 2007. *Phrase Reordering Model Integrating Syntactic Knowledge for SMT*. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 533-540

Zhang, Le. 2006. *Maximum entropy modeling toolkit for python and c++*. available at `http://homepages.inf.ed.ac.uk/lzhang10/maxent_toolkit.html`.

Global Ranking via Data Fusion

Hong-Jie Dai[1,2] Po-Ting Lai[3] Richard Tzong-Han Tsai[3*] Wen-Lian Hsu[1,2*]

[1]Department of Computer Science, National Tsing-Hua University,
[2]Institute of Information Science, Academia Sinica,
[3]Department of Computer Science & Engineering, Yuan Ze University
hongjie@iis.sinica.edu.tw
s951416@mail.yzu.edu.tw
thtsai@saturn.yzu.edu.tw
hsu@iis.sinica.edu.tw

Abstract

Global ranking, a new information retrieval (IR) technology, uses a ranking model for cases in which there exist relationships between the objects to be ranked. In the ranking task, the ranking model is defined as a function of the properties of the objects as well as the relations between the objects. Existing global ranking approaches address the problem by "learning to rank". In this paper, we propose a global ranking framework that solves the problem via data fusion. The idea is to take each retrieved document as a pseudo-IR system. Each document generates a pseudo-ranked list by a global function. The data fusion algorithm is then adapted to generate the final ranked list. Taking a biomedical information extraction task, namely, interactor normalization task (INT), as an example, we explain how the problem can be formulated as a global ranking problem, and demonstrate how the proposed fusion-based framework outperforms baseline methods. By using the proposed framework, we improve the performance of the top 1 INT system by 3.2% using the official evaluation metric of the BioCreAtIvE challenge. In addition, by employing the standard ranking quality measure, NDCG, we demonstrate that the proposed framework can be cascaded with different local ranking models and improve their ranking results.

1 Introduction

Information Retrieval (IR) involves finding documents that are relevant to a given query in a large corpus. The task is usually formulated as a ranking problem. When a user submits a query, the IR system retrieves all documents that contain at least one query term, calculates a ranking score for each of the documents using a ranking model, and sorts the documents according to the ranking scores. The scores represent the relevance, importance, and/or diversity of the retrieved documents. Thus, the quality of a search engine can be determined by the accuracy of the ranking results.

Recently, a machine learning technology called *learning to rank* has been applied extensively to the task. Several state-of-the-art machine learning-based ranking algorithms have been proposed, e.g., RankSVM and RankNet. These algorithms differ substantially in terms of the ranking models and optimization techniques employed, but most of them can be regarded as "local ranking" approaches in the sense that each model is defined on a single document without considering the possible relations to other documents to be ranked. In many applications, this is only a loose approximation as there is always relational information among documents. For example, in some cases, users may prefer that two similar documents have similar relevance scores; even

* Corresponding author

Coling 2008: Poster Volume, pages 223–231,
Beijing, August 2010

though one of the documents is not as relevant to the given query as the other; this problem is similar to Pseudo Relevance Feedback (Kwok, 1984). In other cases, web pages from the same site form a sitemap hierarchy in which a parent document should be ranked higher than its child documents (referred to as Topic Distillation at TREC (Chowdhury, 2007)). To utilize all available information, more advanced ranking algorithms define a ranking model as a function of all the documents to be ranked, i.e., a global ranking model (Qin et al., 2008a; Qin et al., 2008b).

Unlike conventional ranking and learning to rank models, such as BM25 and RankSVM, whose ranking functions are defined on a query and document pair, global ranking models utilize both content information and relation information. Qin et al. (2008) proposed the first supervised learning framework for the global ranking problem. They formulated the problem as an optimization problem that involves finding an objective function to minimize the trade-off between local consistence and global consistence and implemented it on SVM. Subsequently, they defined the global ranking problem formally in (Qin et al., 2008) and solved it by employing continuous conditional random fields (CRF).

In this paper, we propose a new framework for the global ranking problem. The major difference between our work and that of Qin et al. (2008a; 2008b) is that we do not compile a feature vector of relational information directly to construct a new machine-learned ranking model for global ranking. Instead, we use the ranking results generated by the original ranking model and then employ an algorithm with the relational information to transform the global ranking problem into a data fusion problem; that is also known as a rank aggregate problem. The proposed framework is flexible and can be cascaded with conventional ranking models or learning to rank models.

The remainder of this paper is organized as follows. In Section 2, we present a formal definition of global ranking. In Section 3, we describe the proposed framework and consider three fusion algorithms that can be used with our framework. We also explain how the algorithms can be adapted to solve the global ranking problem. In Section 4, we introduce a biomedical text mining task called the interactor normalization task (INT) (Krallinger et al., 2009) and show why it should be formulated as a global ranking problem. In Section 5, we report extensive experiments conducted on the INT dataset released by BioCreAtIvE (Krallinger et al., 2009). Section 6 contains some concluding remarks.

2 Global Ranking Problem

The global ranking problem was first defined formally by Qin et al. (2008). In this paper, we propose a new global ranking framework based on their definition. Although we developed the framework independently, we adopt Qin et al.'s terminology.

Let q denote a query. In addition, let $x^{(q)} = \left\{x_1^{(q)}, x_2^{(q)}, \dots, x_{n^{(q)}}^{(q)}\right\}$ denote the documents retrieved by q, and let $y^{(q)} = \left\{y_1^{(q)}, y_2^{(q)}, \dots, y_{n^{(q)}}^{(q)}\right\}$ denote the ranking scores assigned to the documents. Here, $n^{(q)}$ represents the number of documents retrieved by q. Note that the numbers of documents varies according to different queries. We assume that $y^{(q)}$ is determined by a ranking model.

If a ranking model is defined on a single document, i.e., in the form of

$$y_i^{(q)} = f\left(x_i^{(q)}\right), \mathrm{i} = 1, \dots, n^{(q)}, \quad (1)$$

it is called a "local ranking" model.

Let $\left\{g_k\left(y_i^{(q)}, y_j^{(q)}, x^{(q)}\right)\right\}_{k=1}^{K}$ be a set of real-value functions defined on $y_i^{(q)}$, $y_j^{(q)}$, and $x^{(q)}$ $(i, j = 1, \dots, n^{(q)}, i \neq j)$. The functions

$$g\left(y_i, y_j, x\right) \quad (2)$$

represents the relations between documents. Equation 2 is defined according to the requirements of different tasks. For example, for the Pseudo Relevance Feedback problem, Qin et al. (2008) defined Equation 2 as the similarities between any two documents in their CRF-based model.

If a ranking model takes all the documents as its input and exploits both local and global information (Equation 2) in the documents, i.e., in the form of

$$y^{(q)} = F\left(x^{(q)}\right),$$

it is called a "global ranking" approach.

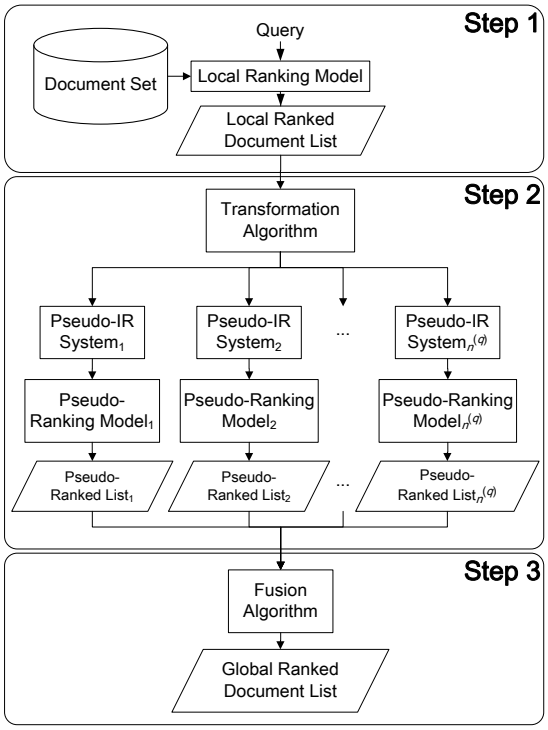


Figure 1. The Proposed Framework for Global Ranking.

3 Fusion-based Global Ranking Framework

It is usually difficult to develop a global ranking algorithm that can fully utilize all the local and global information in documents to produce a document rank and also consider the score ranks. One example of a global ranking algorithm that satisfied these criteria is the one proposed in (Qin et al., 2008) in which the modified CRF algorithm handles context (local) features and relational (global) features in the documents. Without solving a ranking problem directly, however, the modified CRF algorithm is more like a regression algorithm since it optimizes the CRF parameters in a maximum likelihood estimate without considering the score ranks. With respect to the ranking feature, in this section, we describe our framework based on the idea of data fusion for solving the global ranking problem.

3.1 Framework Description

The flow chart of the proposed framework is illustrated in Figure 1. The first step is the same as that of the traditional local ranking model. Given a query, the local ranking model $y_i^{(q)}$ defined in Equation 1 is used to calculate the ranking score for each document, and return a document list sorted according to the local scores.

The second step transforms the global ranking problem into a data fusion problem. Our idea is to take each retrieved document as a pseudo-IR system, and the pseudo-ranking model, $y'^{(q)}_i$, used by each system is the function defined in Equation 2. For each pseudo-IR system, $x_j^{(q)}$, the pseudo-ranking model for a document $x_i^{(q)}$ is defined as follows:

$$y'^{(q)}_i = f\left(x_i^{(q)}\right) = g\left(y_i^{(q)}, y_j^{(q)}, x^{(q)}\right), \quad i = 1, \ldots, n^{(q)}. \tag{3}$$

There are totally $n^{(q)}$ pseudo-IR systems, which generate $n^{(q)}$ pseudo-ranked lists. As a result, the global ranking problem is transformed into a data fusion problem, that is to aggregate the pseudo-ranked lists. Figure 2 shows the steps of the transformation algorithm.

The final step is to adapt fusion algorithms to aggregate the pseudo-ranked lists. A canonical data fusion task is called *meta-search* (Aslam and Montague, 2001; Fox and Shaw, 1994; Lee, 1997; Nuray and Can, 2006), which aggregates Web search query results from several engines into a more accurate ranking. The origin of research on data fusion can be traced back to (Borda, 1781). In recent years, the process has been used in many new applications, including aggregating data from microarray experiments to discover cancer-related genes (Pihura et al., 2008), integration of results from multiple mRNA studies (Lin and Ding, 2008), and similarity searches across datasets and information merging (Adler et al., 2009; Zhao et al., 2010).

Liu et al. (2007) classified data fusion technologies into two categories: order-based fusion and score-based fusion. In the first category, the orders of the entities in individual ranking lists are used by the fusion algorithm. In the second category, the entities in individual ranking lists are assigned scores and the fusion algorithm uses the scores. In the following sub-sections, we adapt three fusion algorithms

```
function transform (x^(q): the documents retrieved
with query q)
{generate pseudo-ranked lists for x^(q)}
# a dictionary that maps the pseudo-IR systems to
# their corresponding pseudo-ranked lists
1. pseudoRankedLists = {}
2. for x_i^(q) in x^(q):
     # a dictionary that maps the relation score (real
     # value) to a list of documents.
3.   relation = {}
     for x_j^(q) in x^(q):
4.     relation[g(y_i^(q), y_j^(q), x^(q))].append(x_j^(q))
     # relation.keys() returns all keys stored in the
     # dictionary relation. The key of relation is the
     # relation score.
5.   Sort relation.keys() in decreasing order
     # a dictionary that maps a new rank to a list of
     # documents.
6.   pseudoRankedList = {}
7.   newRank = 0
     for score in sorted relation.keys():
       # relation[score] returns the document list
       # corresponding to the given score
       for doc in relation[score]:
8.       pseudoRankedList[1+newRank]
                          .append(doc)
9.     newRank = newRank + 1
10.  pseudoRankedLists [x_i^(q)] = pseudoRankedList
   return pseudoRankedLists
```

Figure 2. The Dependent Ranked List Generation Algorithm (represented using python syntax).

for the proposed framework. The first is the Borda-fuse model (Aslam and Montague, 2001), an order-based fusion approach based on an optimal voting procedure. The second is a linear combination (LC) model (Vogt and Cottrell, 1999), which is a score-based fusion approach.

3.2 Borda-fuse

The Borda-fuse model (Aslam and Montague, 2001) is based on a political election strategy called the Borda Count. For our framework, the rationale behind the strategy is as follows. Each pseudo-IR system $x_j^{(q)}$ is an analogy for a voter; and each voter ranks a fixed set of $n^{(q)}$ documents in order of preference (Equation 3). For each voter, the top ranked document is given $n^{(q)}$ points, the second ranked document is given $n^{(q)}$-1 points, and so on. If some documents left unranked by the voter, the remaining points are divided equally among the unranked documents. The documents are ranked in descending order of the total points.

In our framework, we implement two Borda-fuse-based models. The first is the modified Borda-fuse (MBF) model. In MBF, the number of points given for a voter's first and subsequent preferences is determined by the number of documents they have actually ranked, rather than the total number of ranked. Because the ranking model, $y'^{(q)}_i$, used by the pseudo-IR system may only retrieve m documents where m is smaller than $n^{(q)}$, we penalize systems that do not rank a full document set by reducing the number of points their vote distributes among the documents. In other words, if there are ten documents, but the pseudo-IR system only retrieves five, then the first document will only receive 5 points; the second will receive 4 points, and so on.

The second is the weighted Borda-fuse (WBF) model. The original Borda-fuse model reflects a democratic election in which each voter has equal weight. However, in many cases, we prefer some voters because they are more reliable. We employ a simple weighting scheme that multiplies the points assigned to a document determined by system $x_j^{(q)}$ by a weight $w_{x_j^{(q)}}$.

3.3 LC Model

The LC model has been used by many IR researchers with varying degrees of success (Bartell et al., 1994; Knaus et al., 1995; Vogt and Cottrell, 1999; Vogt and Cottrell, 1998). In our framework, it is defined as follows. Given a query q, a document $x_i^{(q)}$, the weights $\mathbf{w} = (w_1, w_2, w_3, \ldots, w_{n^{(q)}})$ for $n^{(q)}$ individual pseudo-IR systems, and jth pseudo-IR system's ranking score $y_j{}'^{(q)}_i$, the LC model calculates the ranking score ρ of $x_i^{(q)}$ against all pseudo-IR systems as follows:

$$\rho\left(\mathbf{w}, x_i^{(q)}\right) = \sum_{j=1}^{n^{(q)}} w_i y_j{}'^{(q)}_i \qquad (4)$$

This score is then used to rank the documents. For example, for two pseudo-IR systems, this reduces to:

$$\rho\left(w_1, w_2, x_i^{(q)}\right) = w_1 y_1{}'^{(q)}_i + w_2 y_2{}'^{(q)}_i$$

Compared with MBF, Equation 4 requires both relevance scores and training data to de-

termine the weight w_i given to each pseudo-IR system.

4 Case Study

In this section, we describe the task examined in our study. We also explain how we formulate the task as a global ranking problem. The experiments results are detailed in Section 5.

4.1 Interactor Normalization Task

The interactor normalization task (INT) is a complicated text mining task that involves the following steps: (1) It recognizes gene mentions in a full text article. (2) It maps the recognized gene mentions to corresponding unique database identifiers which is similar to the word sense disambiguation task in computational linguistics. (3) It generates a ranked list of the identifiers according to their importance in the article and their probability of playing the interactor role in protein-protein interactions (PPIs). Such ranked lists are useful for human curators and can speed up PPI database curation.

Dai et al. (2010) won first place in the BioCreAtIvE II.5 INT challenge (Mardis et al., 2009) by using a SVM-based local ranking model in which they treat gene mentions' identifiers in an article as the document set, and the query is a constant string "interactor". Based on their feature sets and evaluation results, we can find that their local ranking model tends to rank focus genes higher (Dai et al., 2010). However, the primary objective of INT is to generate a ranked list of interaction gene identifiers. According to (Jenssen et al., 2001), co-mentioned genes are usually related in some way. For example, if two gene mentions frequently occur alongside each other in the same sentence in an article, they probably have an association and influence each other's rank. Take a low-ranked interactor that is only mentioned twice in an article as an example. If both mentions are next to the highest-ranked interactor in the article, then the low-ranked interactor's rank should be boosted significantly. Therefore, the ranking task for each article can be formulated as a global ranking problem; the global ranking algorithm should consider both the local information from Dai et al.'s model and the global information from the associations among identifiers.

4.2 Global Ranking in INT

Let q be a constant "interactor." The identifier set generated by an INT system for a full-text article is analogous to the document set $x^{(q)} = \left\{x_1^{(q)}, x_2^{(q)}, \ldots, x_{n^{(q)}}^{(q)}\right\}$. Here $n^{(q)}$ denotes the number of identifiers. Note that the number of identifiers varies for different articles. Let $y^{(q)} = \left\{y_1^{(q)}, y_2^{(q)}, \ldots, y_{n^{(q)}}^{(q)}\right\}$ denote the ranking scores assigned to the identifiers given by a local ranking model. In this study, we used the INT system and SVM-based local ranking model released by (Dai et al., 2010) to generate the identifier set and ranking scores.

To obtain the global information, we consider the co-occurrence of identifiers and employ mutual information (MI) to measure the association between two identifiers as follows:

$$g(y_i, y_j, x) = \mathrm{MI}(x_i, x_j) = P(x_i, x_j) / \left(P(x_i) \times P(x_j)\right).$$

In the above formula, the identifier probabilities $P(x_i)$ and $P(x_j)$ are estimated by counting the number of occurrences in an article normalized by N, i.e., the number of sentences containing identifiers. The joint probability, $P(x_i, x_j)$, is estimated by the number of times x_i co-occurs with x_j in a window of k words normalized by N. Note that, in practice, other advanced approaches can be used to calculate the association score.

For the proposed framework, each identifier $x_i^{(q)}$ is a pseudo-IR system with MI as its pseudo-ranking model $y'^{(q)}_i$. The identifiers that co-occur with $x_i^{(q)}$ become candidates on $x_i^{(q)}$'s pseudo-ranked list.

5 Experiments

In the following sub-sections, we introduce the dataset used in the experiments, describe the evaluation methods, report the results of the experiments conducted to compare the performance of different methods, and discuss the efficiency of the proposed global ranking framework.

5.1 Dataset

We used the BioCreAtIvE II.5 Elsevier corpus released by BioCreAtIvE II.5 challenge in the experiments. The corpus contains 1,190 full-text journal articles selected mainly from FEBS Letters. Following the same format as the BioCreAtIvE II.5 INT challenge, we used articles published in 2008 (61 articles) as our training set and articles published in 2007 or earlier (61 articles) as our test set.

5.2 A Fusion-based Global Ranking Framework for INT

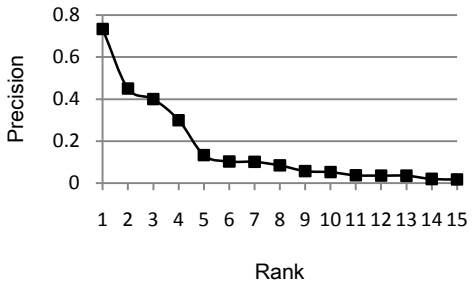


Figure 3. Precision of Different Ranks.

Before applying the proposed framework, we preprocess the articles in the dataset to identify all gene mentions, and map them to their corresponding identifiers. After preprocessing, each full-text article is associated with a list of identifiers (Step 1 in Figure 1). The transform and fusion algorithm is then applied on each article (Steps 2 and 3 in Figure 1).

To apply the WBF and LC models, we need to determine the weight assigned to each pseudo-IR system. To obtain the weight, we calculate the precision of each rank of the ranked lists generated by Dai et al.'s INT system. Figure 3 shows the precision of ranks 1 to 15 calculated by applying three-fold cross validation on the INT training set. We observe that the precision declines as the rank increases, which implies that the higher ranks predicted by their SVM-based local ranking model are more reliable than the lower ranks.

5.3 Evaluation Metrics

Our evaluations focus on two comparisons: the first compares the ranking of the proposed framework with the original local ranking model by using the area under the curve of the interpolated precision/recall (iP/R) curve. This is the evaluation metric used in the BioCreAtIvE II.5 challenge and is a common way to depict the degradation of precision as one traverses the retrieved results by plotting interpolated precision numbers against percentage recall. The area under the iP/R function f_{pr} is defined as follows:

$$Area\_iPR(f_{pr}) = \sum_{j=1}^{n}\left(p_{i_j} \times (r_j - r_{j-1})\right),$$

$$p_i(r) = \max_{r' \geq r} p(r')$$

where n is the total number of correct identifiers and p_i is the highest interpolated precision for the correct identifier j at r_j, the recall for that hit. The interpolated precision p_i is calculated for each recall r by taking the highest precision at r or any $r' \geq r$.

In the second comparison, we use a standard quality measure in IR to estimate the ranking performance of local ranking models and the proposed framework. We adopt Normalized Discounted Cumulative Gain (NDCG) to measure the performance. The NDCG score of a ranking is computed based on DCG (Discounted Cumulative Gain) as follows:

$$\mathrm{DCG}(r) = g(1) + \sum_{i=2}^{r} \frac{g(i)}{\log_2(i)},$$

where r is the rank position, and $g(i) \in \{0,1\}$ is the relevance grade of the ith identifier in the ranked result set. In our experiment, $g(i) = 1$ corresponds to an interaction identifier, and $g(i) = 0$ corresponds to other identifiers. NDCG is then computed as follows:

$$\mathrm{NDCG}(r) = \frac{\mathrm{DCG}(r)}{\mathrm{IDCG}(r)},$$

where IDCG denotes the results of a perfect ranking. The NDCG values for all articles are averaged to obtain the average performance of the proposed framework.

5.4 INT Test Set Performance

Figure 4 shows the *Area_iPR* scores of four configurations. In the baseline configuration (Local/Rank1), the SVM-based local ranking model released by Dai et al. is employed. In the configuration Global+LC, Global+MBF, and Global+WBF, the proposed global ranking framework is cascaded with the local ranking model and with three data fusion models: the LC model, the modified Borda-fuse (MBF) model, and the weighted Borda-fuse model. The figure also shows the *Area_iPR* scores of

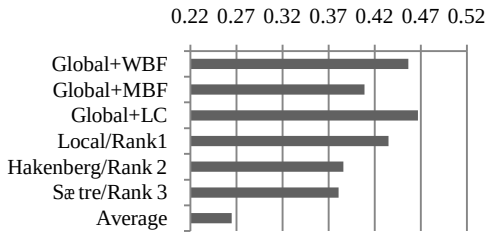


Figure 4. The *Area_iPR* Results of Different Ranking Models

the top three teams and the average *Area_iPR* score of all BioCreAtIvE II.5 INT participants (Average).

The results show that under the global ranking framework, *Area_iPR* performance is improved in addition to Global+MBF. The highest *Area_iPR* (Global+LC: 46.7%) is 3.2% higher than the Rank 1 score in the BioCreAtIvE II.5 INT challenge. According to our analysis, before global ranking, identifiers whose feature values rarely appear in the training set are often ranked incorrectly because their feature values are under-estimated by the ranking model. However, if the identifiers co-occur with higher-ranked identifiers whose feature values appear frequently, the proposed framework is very likely to increase their ranks. This results in an improved *Area_iPR* score.

5.5 Global Ranking Performance

Based on	Global Ranking	NDCG1	NDCG3	NDCG5
Local Ranking /Rank1	Global+LC	+0.908	+1.323	-0.003
	Global+MBF	-3.279	-1.034	-0.020
	Global+WBF	-0.016	+3.630	+2.071
Freq	Global+LC_f	+1.639	+3.152	+2.817
	Global+MBF_f	-6.860	-4.275	-4.839
	Global+WBF_f	+2.549	+2.390	+3.043

Table 1. The NDCG Gain (%) of Different Ranking Models.

To illustrate the effectiveness of the proposed global ranking framework and assess its performance when it is cascaded with other conventional ranking models, we implement a simple term frequency-based ranking function, which is based on the identifier frequency in an article as another local ranking model. If two or more identifiers have the same frequency, two heuristic rules are employed sequentially to rank them: (1) the identifier with the highest frequency in the Results section of the article, and (2) the identifier mentioned first in the article.

Table 1 shows the NDCG percentage gain of different ranking models. It compares the ranked list generated by our global ranking framework and by the local ranking models. We observe that (1) irrespective of whether the local ranking model is a conventional model or a learning to rank model, Global+LC and Global+WBF models achieve NDCG gains over the original rankings of the local ranking models; (2) the results show that our global ranking framework can improve the performance by only exploiting MI analysis. However, it is expected that employing more advanced relation extraction methods to determine the global information (Equation 3) would yield more reliable pseudo-ranked lists and lead to a further improvement in the final ranking; and (3) similar to the results in Section 5.4, the performance of Global+MBF does not improve. Global+MBF has a negative NDCG gain and the *Area_iPR* decreases by 2.61%. We believe this is due to MBF gives equal weight to each pseudo-IR system. As mentioned in Section 4.1, the document set in INT is comprised of the identifiers of the gene mentions derived by Dai et al.'s system. Unfortunately, there must be incorrect identifiers (the errors may be due to their gene mention recognition or identifier mapping processes). As in the meta-search, the best performance is often achieved by weighting the input systems unequally. Reasonable weights allow the algorithm to concentrate on good feedback from pseudo-IR systems and ignore poor feedback. As shown by the average precision results in Figure 3, the identifiers (corresponding to the pseudo-IR systems in our framework) in the higher ranks are more reliable; however, MBF cannot use this information, which leads to a negative NDCG gain and a lower *Area_iPR* score.

6 Conclusion

We have presented a new global ranking framework based on data fusion technology. Our approach solves the global ranking problem in three stages: the first stage ranks the document set by the original local ranking model; the second stage transforms the prob-

lem into a data fusion task by using global information, and the final stage adapts fusion algorithms to solve the ranking problem. The framework is flexible and it can be combined with other mature ranking models and fusion algorithms. We also show how the BioCreAtIvE INT can be formulated as a global ranking problem and solved by the proposed framework. Experiments on the INT dataset demonstrate the effectiveness of the proposed framework and its superior performance over other ranking models.

In our future work, we will address the following issues: (1) the use of advanced data fusion algorithms in the proposed framework; (2) assessing the performance of the proposed framework on other tasks, such as Pseudo Relevance Feedback and Topic Distillation; and (3) design an advanced supervised learning relation extraction algorithm to replace MI in INT to evaluate the system performance.

References

Adler, P., R. Kolde, M. Kull, A. Tkachenko, H. Peterson, J. Reimand and J. Vilo (2009). *Mining for coexpression across hundreds of datasets using novel rank aggregation and visualization methods*. *Genome Biology* 10(R139).

Aslam, J. A. and M. Montague (2001). *Models for metasearch*. *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, New Orleans, Louisiana, United States, ACM.

Bartell, B. T., G. W. Cottrell and R. K. Belew (1994). *Automatic combination of multiple ranked retrieval systems*. *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, Dublin, Ireland Springer-Verlag New York, Inc.

Borda, J. (1781). *Mémoire sur les élections au scrutin. Histoire del'Acad´emie Royale des Sciences* 2: 13.

Chowdhury, G. (2007). *TREC: Experiment and Evaluation in Information Retrieval*. *Online Information Review* 31(5): 462.

Dai, H.-J., P.-T. Lai and R. T.-H. Tsai (2010). *Multi-stage gene normalization and SVM-based ranking for protein interactor extraction in full-text articles*. *IEEE TRANSACTIONS ON COMPUTATIONAL BIOLOGY AND BIOINFORMATICS* 14 May. 2010. IEEE computer Society Digital Library. IEEE Computer Society.

Fox, E. A. and J. A. Shaw (1994). *Combination of Multiple Searches*. *1994*, Proceedings of the Second Text REtrieval Conference (TREC 2)

Jenssen, T.-K., A. Lagreid, J. Komorowski and E. Hovig (2001). *A literature network of human genes for high-throughput analysis of gene expression*. *Nature Genetics* 28(1): 21-28.

Knaus, D., E. Mittendorf and P. Schäuble (1995). *Improving a basic retrieval method by links and passage level evidence*. *NIST Special Publication 500-225: Overview of the Third Text REtrieval Conference (TREC-3)*.

Krallinger, M., F. Leitner and A. Valencia (2009). *The BioCreative II.5 challenge overview*. *Proceedings of the BioCreative II.5 Workshop 2009 on Digital Annotations*, Madrid, Spain.

Kwok, K. L. (1984). *A document-document similarity measure based on cited titles and probability theory, and its application to relevance feedback retrieval*. *SIGIR'84*.

Lee, J. H. (1997). *Analyses of multiple evidence combination*. *Proceedings of the 20th annual international ACM SIGIR conference on Research and development in information retrieval*, Philadelphia, Pennsylvania, United States, ACM.

Lin, S. and J. Ding (2008). *Integration of Ranked Lists via Cross Entropy Monte Carlo with Applications to mRNA and microRNA Studies*. *Biometrics* 65(1): 9-18.

Liu, Y.-T., T.-Y. Liu, T. Qin, Z.-M. Ma and H. Li (2007). *Supervised rank aggregation*. *Proceedings of the 16th international conference on World Wide Web*, Banff, Alberta, Canada, ACM.

Mardis, S., F. Leitner and L. Hirschman (2009). *BioCreative II.5: Evaluation and ensemble system performance*. *Proceedings of the BioCreative II.5 Workshop 2009 on Digital Annotations*, Madrid, Spain.

Nuray, R. and F. Can (2006). *Automatic ranking of information retrieval systems using data fusion*. *Inf. Process. Manage.* 42(3): 595-614.

Pihura, V., S. Dattaa and S. Datta (2008). *Finding common genes in multiple cancer types through meta–analysis of microarray experiments: A*

rank aggregation approach Genomics 92(6): 400-403

Qin, T., T.-Y. Liu, X.-D. Zhang, D.-S. Wang and H. Li (2008). *Global Ranking Using Continuous Conditional Random Fields*. *Proceedings of the Twenty-Second Annual Conference on Neural Information Processing Systems (NIPS 2008)*, Vancouver, Canada.

Qin, T., T. Liu, X. Zhang, D. Wang, W. Xiong and H. Li (2008). *Learning to rank relational objects and its application to web search*, ACM.

Vogt, C. and G. Cottrell (1999). *Fusion via a linear combination of scores*. *Information Retrieval* 1(3): 151-173.

Vogt, C. C. and G. W. Cottrell (1998). *Predicting the performance of linearly combined IR systems*. *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, Melbourne, Australia ACM.

Zhao, Z., J. Wang, S. Sharma, N. Agarwal, H. Liu and Y. Chang (2010). *An Integrative Approach to Identifying Biologically Relevant Genes*. *Proceedings of SIAM International Conference on Data Mining (SDM)*.

Topic-Based Bengali Opinion Summarization

Amitava Das
Department of Computer Science and Engineering
Jadavpur University
amitava.santu@gmail.com

Sivaji Bandyopadhyay
Department of Computer Science and Engineering
Jadavpur University
sivaji_cse_ju@yahoo.com

Abstract

In this paper the development of an opinion summarization system that works on Bengali News corpus has been described. The system identifies the sentiment information in each document, aggregates them and represents the summary information in text. The present sys-tem follows a topic-sentiment model for sentiment identification and aggregation. Topic-sentiment model is designed as discourse level theme identification and the topic-sentiment aggregation is achieved by theme clustering (k-means) and Document level Theme Relational Graph representation. The Document Level Theme Relational Graph is finally used for candidate summary sentence selection by standard page rank algorithms used in Information Retrieval (IR). As Bengali is a resource constrained language, the building of annotated gold standard corpus and acquisition of linguistics tools for lexico-syntactic, syntactic and discourse level features extraction are described in this paper. The reported accuracy of the Theme detection technique is 83.60% (precision), 76.44% (recall) and 79.85% (F-measure). The summarization system has been evaluated with Precision of 72.15%, Recall of 67.32% and F-measure of 69.65%.

1 Introduction

The Web has become a rich source of various opinions in the form of product reviews, travel advice, social issue discussions, consumer complaints, movie review, stock market predictions, real estate market predictions, etc. Present computational systems need to extend the power of understanding the sentiment/opinion expressed in an electronic text to act properly in the society rather than dealing with the topic of a document. The topic-document model of information retrieval has been studied for a long time and systems are available publicly since last decade. On the contrary Opinion Mining/Sentiment Analysis is still an unsolved research problem. Although a few systems like Twitter Sentiment Analysis Tool[1], TweetFeel[2] are available in World Wide Web since last few years still more research efforts are necessary to match the user satisfaction level and social need.

Researchers have taken multiple approaches towards the problem of Opinion Summarization like Topic-sentiment model, Textual summaries at single document or multiple document perspective and graphical summaries or visualization. The works on opinion tracking systems have explicitly incorporated temporal dimension. The topic-sentiment model is well established for opinion retrieval.

The concept of reputation system was first introduced in (Resnick et al., 2000). Reputation systems for both buyers and sellers are needed to earn each other's trust in online interactions.

Ku et al., (2005) selects representative words from a document set to identify the main concepts in the document set. A term is considered to represent a topic if it appears frequently across documents or in each document. Different methodologies have been used to assign weights to each word both at document level and paragraph level. The precision and recall values of the system have been reported as 0.56 and 0.85.

[1] http://twittersentiment.appspot.com/
[2] http://www.tweetfeel.com/

Coling 2008: Poster Volume, pages 232–240,
Beijing, August 2010

Zhou et al. (2006) have proposed the architecture for generative summary from blogosphere. Typical multi-document summarization (MDS) systems focus on content selection followed by synthesis by removing redundancy across multiple input documents. The online discussion summarization system (Zhou et al., 2006) work on an online discussion corpus involving multiple participants and discussion topics are passed back and forth by various participants. MDS systems are insufficient in representing this aspect of the interactions. Due to the complex structure of the dialogue, similar subtopic structure identification in the participant-written dialogues is essential. Maximum Entropy Model (MEMM) and Support Vector Machine (SVM) have been used with a number of relevant features.

Carenini et al. (2006) present and compare two approaches to the task of multi document opinion summarization on evaluative texts. The first is a sentence extraction based approach while the second one is a natural language generation-based approach. Relevant extracted features are categorized in two types: User Defined Features (UDF) and Crude Features (CF) as described in (Hu and Liu, 2004).

The summary generation technique uses the aggregation of the extracted features, CF and UDF. Opinion aggregation has been done by the two relevant features: opinion strength and polarity. A new opinion distribution function feature has been introduced to capture the overall opinion distributed in corpus.

Kawai et al. (2007) developed a news portal site called Fair News Reader (FNR) that recommends news articles with different sentiments for a user in each of the topics in which the user is interested. FNR can detect various sentiments of news articles and determine the sentimental preferences of a user based on the sentiments of previously read articles by the user. News articles crawled from various news sites are stored in a database. The contents are integrated as needed and the summary is presented on one page. A sentiment vector on the basis of word lattice model has been generated for every document. A user sentiment model has been proposed based on user sentiment state. The user sentiment state model works on the browsing history of the user. The intersection of the documents under User Vector and Sentiment Vector are the results.

2 Resource Organization

Resource acquisition is one of the most challenging obstacles to work with resource constrained languages like Bengali. Bengali is the fifth popular language in the World, second in India and the national language in Bangladesh. Extensive NLP research activities in Bengali have started recently but resources like annotated corpus, various linguistic tools are still unavailable for Bengali in the required measure. The manual annotation of gold standard corpus and acquisition of various tools used in the feature extraction for Bengali are described in this section.

2.1 Gold Standard Data Acquisition

2.1.1 Corpus

For the present task a Bengali news corpus has been developed from the archive of a leading Bengali news paper available on the Web (http://www.anandabazar.com/). A portion of the corpus from the editorial pages, i.e., Reader's opinion section or Letters to the Editor Section containing 28K word forms has been manually annotated with sentence level subjectivity and discourse level theme words. Detailed reports about this news corpus development in Bengali can be found in (Das and Bandyopadhyay, 2009b).

2.1.2 Annotation

From the collected document set (Letters to the Editor Section), some documents have been chosen for the annotation task. Some statistics about the Bengali news corpus is represented in the Table 1. Documents that have appeared within an interval of four months are chosen on the hypothesis that these letters to the editors will be on related events. A simple annotation tool has been designed for annotating the sentences considered to be important for opinion summarization. Three annotators (Mr. X, Mr. Y and Mr. Z) participated in the present task.

```
<Story>
……..
……..
<SS><TW>Sargeant O'Leary</TW> said "the
<TW>incident</TW> took place at 2:00pm."</SS>
……..
</Story>
```

Figure 1: XML Annotation Format

Annotators were asked to annotate sentences for summary and to mark the theme words (topical expressions) in those sentences. The documents with such annotated sentences are saved in

XML format. Figure 1 shows the XML annotation format. "<SS>" marker denotes subjective sentences and "<TW>" denotes the theme words.

Bengali NEWS Corpus Statistics	
Total number of documents in the corpus	100
Total number of sentences in the corpus	2234
Average number of sentences in a document	22
Total number of wordforms in the corpus	28807
Average number of wordforms in a document	288
Total number of distinct wordforms in the corpus	17176

Table 1: Bengali News Corpus Statistics

The annotation tool highlights the sentiment words (Das and Bandyopadhyay, 2010a)[3] by four different colors within a document according to their POS categories (Noun, Adjective, Adverb and Verb). This technique helps to increase the speed of annotation process. Finally 100 annotated documents have been developed.

2.1.3 Inter-annotator Agreement

The agreement of annotations among three annotators has been evaluated. The agreements of tag values at theme words level and sentence levels are listed in Tables 2 and 3 respectively.

Annotators	X vs. Y	X Vs. Z	Y Vs. Z	Avg
Percentage	82.64%	71.78%	80.47%	78.30%
All Agree	69.06%			

Table 2: Agreement of annotators at theme words level

Annotators	X vs. Y	X Vs. Z	Y Vs. Z	Avg
Percentage	73.87%	69.06%	60.44%	67.8%
All Agree	58.66%			

Table 3: Agreement of annotators at sentence level

From the analysis of inter-annotator agreement, it is observed that the agreement drops fast as the number of annotator's increases. It is less possible to have consistent annotations when more annotators are involved. In the present task the inter-annotator agreement is better for theme words annotation rather than candidate sentence identification for summary though a small number of documents have been considered.

Further discussion with annotators reveals that the psychology of annotators is to grasp as many as possible theme words identification during annotation but the same groups of annotators are more cautious during sentence identification for summary as they are very conscious to find out the most concise set of sentences that best describe the opinionated snapshot of any document. The annotators were working independent of each other and they were not trained linguists.

2.2 Subjectivity Classifier

Work in opinion mining and classification often assumes the incoming documents to be opinionated. Opinion mining system makes false hits while attempting to summarize non-subjective or factual sentences or documents. It becomes imperative to decide whether a given document contains subjective information or not as well as to identify which portions of the document are subjective or factual. This task is termed as subjectivity detection in sentiment literature. The subjectivity classifier that uses SVM machine learning technique and described in (Das and Bandyopadhyay, 2009a) has been used here. The recall measure of the present classifier is greater than its precision value. The evaluation results of the classifier are 72.16% (Precision) and 76.00 (recall) on the News Corpus.

2.3 Feature Organization

The set of features used in the present task have been categorized as Lexico-Syntactic, Syntactic and Discourse level features. These are listed in the Table 4 below and have been described in the subsequent subsections.

Types	Features
Lexico-Syntactic	POS
	SentiWordNet
	Frequency
	Stemming
Syntactic	Chunk Label
	Dependency Parsing Depth
Discourse Level	Title of the Document
	First Paragraph
	Term Distribution
	Collocation

Table 4: Features

2.3.1 Lexico-Syntactic Features

2.3.1.1 Part of Speech (POS)

It has been shown in (Hatzivassiloglou et. al., 2000), (Chesley et. al., 2006) etc. that opinion bearing words in sentences are mainly adjective, adverb, noun and verbs. Many opinion mining tasks, like (Nasukawa et. al., 2003) are mostly based on adjective words. Details of the Bengali POS tagger used can be found in (Das and Bandyopadhyay 2009b).

[3] http://www.amitavadas.com/sentiwordnet.php

2.3.1.2 SentiWordNet (Bengali)

Words that are present in the SentiWordNet carry opinion information. The developed SentiWordNet (Bengali) (Das and Bandyopadhyay, 2010a) is used as an important feature during the learning process. These features are individual sentiment words or word n-grams (multiword entities) with strength measure as strong subjective or weak subjective. Strong and weak subjective measures are treated as a binary feature in the supervised classifier. Words which are collected directly from SentiWordNet (Bengali) are tagged with positivity or negativity score. The subjectivity score of these words are calculated as:

$$E_s = | S_p | + | S_n |$$

where E_s is the resultant subjective measure and S_p , S_n are the positivity and negativity scores respectively.

2.3.1.3 Frequency

Frequency always plays a crucial role in identifying the importance of a word in the document. The system generates four separate high frequent word lists for four POS categories: Adjective, Adverb, Verb and Noun after function words are removed. Word frequency values are then effectively used as a crucial feature in the Theme Detection technique.

2.3.1.4 Stemming

Several words in a sentence that carry opinion information may be present in inflected forms and stemming is necessary for them before they can be searched in appropriate lists. Due to non availability of good stemmers in Indian languages especially in Bengali, a stemmer (Das and Bandyopadhyay, 2010b) based on stemming cluster technique has been used. This stemmer analyzes prefixes and suffixes of all the word forms present in a particular document. Words that are identified to have the same root form are grouped in a finite number of clusters with the identified root word as cluster center.

2.3.2 Syntactic Features

2.3.2.1 Chunk Label

Chunk level information is effectively used as a feature in supervised classifier. Chunk labels are defined as B-X (Beginning), I-X (Intermediate) and E-X (End), where X is the chunk label. In the task of identification of Theme expressions, chunk label markers play a crucial role. Further details of development of chunking system could be found in (Das and Bandyopadhyay 2009b).

2.3.2.2 Dependency Parser

Dependency depth feature is very useful to identify Theme expressions. A particular Theme word generally occurs within a particular range of depths in a dependency tree. Theme expressions may be a Named Entity (NE: person, organization or location names), a common noun (Ex: accident, bomb blast, strike etc) or words of other POS categories. It has been observed that depending upon the nature of Theme expressions it can occur within a certain depth in the dependency tree for the sentence. A statistical dependency parser has been used for Bengali as described in (Ghosh et al., 2009).

2.3.3 Discourse Level Features

2.3.3.1 Positional Aspect

Depending upon the position of the thematic clue, every document is divided into a number of zones. The features considered for each document are Title words of the document, the first paragraph words and the words from the last two sentences. A detailed study was done on the Bengali news corpus to identify the roles of the positional aspect features of a document (first paragraph, last two sentences) in the detection of theme words and subjective sentences for generating the summary of the document. The importance of these positional features is shown in Tables 5 on the Bengali gold standard set.

2.3.3.2 Title Words

Title words of a document always carry some meaningful thematic information. The title word feature has been used as a binary feature during CRF based machine learning.

2.3.3.3 First Paragraph Words

People usually give a brief idea of their beliefs and speculations in the first paragraph of the document and subsequently elaborate or support them with relevant reasoning or factual information. Hence first paragraph words are informative in the detection of Thematic Expressions.

2.3.3.4 Words From Last Two Sentences

Generally every document concludes with a summary of the opinions expressed in the document.

Positional Factors	Bengali
First Paragraph	56.80%
Last Two Sentences	78.00%

Table 5: Statistics on Positional Aspect.

2.3.3.5 Term Distribution Model

An alternative to the classical TF-IDF weighting mechanism of standard IR has been proposed as a model for the distribution of a word. The model characterizes and captures the informativeness of a word by measuring how regularly the word is distributed in a document. As discussed in Section 1, Carenini et al. (2006) have introduced the opinion distribution function feature to capture the overall opinion distributed in the corpus. Thus the objective is to estimate $f_d(w_i)$ that measures the distribution pattern of the k occurrences of the word w_i in a document d. Zipf's law describes distribution patterns of words in an entire corpus. In contrast, term distribution models capture regularities of word occurrence in subunits of a corpus (e.g., documents, paragraphs or chapters of a book). A good understanding of the distribution patterns is useful to assess the likelihood of occurrences of a word in some specific positions (e.g., first paragraph or last two sentences) of a unit of text. Most term distribution models try to characterize the informativeness of a word identified by inverse document frequency (IDF). In the present work, the distribution pattern of a word within a document formalizes the notion of topic-sentiment informativeness. This is based on the Poisson distribution. Significant Theme words are identified using TF, Positional and Distribution factor. The distribution function for each theme word in a document is evaluated as follows:

$$f_d(w_i) = \sum_{i=1}^{n} (S_i - S_{i-1}) / n + \sum_{i=1}^{n} (TW_i - TW_{i-1}) / n$$

where n=number of sentences in a document with a particular theme word, S_i=sentence id of the current sentence containing the theme word and S_{i-1}=sentence id of the previous sentence containing the query term, TW_i is the positional id of current Theme word and TW_{i-1} is the positional id of the previous Theme word.

2.3.3.6 Collocation

Collocation with other thematic word/expression is undoubtedly an important clue for identification of theme sequence patterns in a document. A window size of 5 including the present word is considered during training to capture the collocation with other thematic words/expressions.

3 Theme Detection

Term Frequency (TF) plays a crucial role to identify document relevance in Topic-Based Information Retrieval. The motivation behind developing Theme detection technique is that in many documents relevant words may not occur frequently or irrelevant words may occur frequently. Moreover for sentiment analysis topic words should have sentiment conceptuality. The Theme detection technique has been proposed to resolve these issues to identify discourse level relevant topic-semantic nodes in terms of word or expressions using a standard machine learning technique. The machine learning technique used here is Conditional Random Field (CRF)[4]. The theme word detection is defined as a sequence labeling problem. Depending upon the series of input feature, each word is tagged as either Theme Word (TW) or Other (O).

4 Theme Clustering

Theme clustering algorithms partition a set of documents into finite number of topic based groups or clusters in terms of theme words/expressions. The task of document clustering is to create a reasonable set of clusters for a given set of documents. A reasonable cluster is defined as the one that maximizes the within-cluster document similarity and minimizes between-cluster similarities. There are two principal motivations for the use of this technique in theme clustering setting: efficiency, and the **cluster hypothesis**.

The **cluster hypothesis** (Jardine and van Rijsbergen, 1971) takes this argument a step further by asserting that retrieval from a clustered collection will not only be more efficient, but will in fact improve retrieval performance in terms of recall and precision. The basic notion behind this hypothesis is that by separating documents according to topic, relevant documents will be found together in the same cluster, and non-relevant documents will be avoided since they will reside in clusters that are not used for retrieval. Despite the plausibility of this hypothesis, there is only mixed experimental support for it. Results vary considerably based on the clus-

[4] http://crfpp.sourceforge.net

tering algorithm and document collection in use (Willett, 1988; Shaw et al., 1996).

Application of the clustering technique to the three sample documents results in the following theme-by-document matrix, A, where the rows represent Docl, Doc7 and Doc13 and the columns represent the themes politics, sport, and travel.

$$A = \begin{bmatrix} election & cricket & hotel \\ parliament & sachin & vacation \\ governor & soccer & tourist \end{bmatrix}$$

The similarity between vectors is calculated by assigning numerical weights to these words and then using the cosine similarity measure as specified in the following equation.

$$s\left(\vec{q}_k, \vec{d}_j\right) = \vec{q}_k . \vec{d}_j = \sum_{i=1}^{N} w_{i,k} \times w_{i,j} \text{ ---- (1)}$$

This equation specifies what is known as the dot product between vectors. Now, in general, the dot product between two vectors is not particularly useful as a similarity metric, since it is too sensitive to the absolute magnitudes of the various dimensions. However, the dot product between vectors that have been length normalized has a useful and intuitive interpretation: it computes the **cosine** of the angle between the two vectors. When two documents are identical they will receive a cosine of one; when they are orthogonal (share no common terms) they will receive a cosine of zero. Note that if for some reason the vectors are not stored in a normalized form, then the normalization can be incorporated directly into the similarity measure as follows.

$$s\left(\vec{q}_k, \vec{d}_j\right) = \frac{\sum_{i=1}^{N} w_{i,k} \times w_{i,j}}{\sqrt{\sum_{i=1}^{N} w_{i,k}^2} \times \sqrt{\sum_{i=1}^{N} w_{i,k}^2}} \text{ ----(2)}$$

Of course, in situations where the document collection is relatively static, it makes sense to normalize the document vectors once and store them, rather than include the normalization in the similarity metric.

Calculating the similarity measure and using a predefined threshold value, documents are classified using standard bottom-up soft clustering k-means technique. The predefined threshold value is experimentally set to 0.5 as shown in Table 6.

A set of initial cluster centers is necessary in the beginning. Each document is assigned to the cluster whose center is closest to the document. After all documents have been assigned, the center of each cluster is recomputed as the centroid or mean $\vec{\mu}$ (where $\vec{\mu}$ is the clustering coefficient) of its members, that is $\vec{\mu} = \left(1/\left|c_j\right|\right)\sum_{x \in c_j} \vec{x}$. The distance function is the **cosine vector** similarity function.

ID	Themes	1	2	3
1	**প্রশাসন** (administration)	**0.63**	0.12	0.04
1	সুশাসন (good-government)	**0.58**	0.11	0.06
1	সমাজ (Society)	**0.58**	0.12	0.03
1	আইন (Law)	**0.55**	0.14	0.08
2	গবেষণা (Research)	0.11	**0.59**	0.02
2	কলেজ (College)	0.15	**0.55**	0.01
2	**উচ্চশিক্ষা (Higher Study)**	0.12	**0.66**	0.01
3	জেহাদি (Jehadi)	0.13	0.05	**0.58**
3	মসজিদ (Mosque)	0.05	0.01	**0.86**
3	মুশারফ (Musharaf)	0.05	0.01	**0.86**
3	**কাশ্মীর (Kashmir)**	0.03	0.01	**0.93**
3	পাকিস্তান (Pakistan)	0.06	0.02	**0.82**
3	নয়াদিল্লী (New Delhi)	0.12	0.04	**0.65**
3	বর্ডার (Border)	0.08	0.03	**0.79**

Table 6: Five cluster centroids (mean $\vec{\mu}_j$)

Table 6 gives an example of theme centroids from the K-means clustering. Bold words in Theme column are cluster centers. Cluster centers are assigned by maximum clustering coefficient. For each theme word, the cluster from table 6 is still the dominating cluster. For example, "প্রশাসন" has a higher membership probability in cluster 1. But each theme word also has some non-zero membership in all other clusters. This is useful for assessing the strength of association between a theme word and a topic. Comparing two members of the cluster2, "কাশ্মীর" and "নয়াদিল্লী", it is seen that "নয়াদিল্লী" is strongly associated with cluster2 (p=0.65) but has some affinity with other clusters as well (e.g., p =0.12 with the cluster1). This is a good example of the utility of soft clustering. These non-zero values are still useful for calculating vertex weights during Theme Relational Graph generation.

5 Construction of Document Level Theme Relational Graph

Representation of input text document(s) in the form of graph is the key to our design principle. The idea is to build a document graph G=<V,E> from a given source document $d \in D$. First, the input document *d* is parsed and split into a number of text fragments (sentence) using sentence delimiters (Bengali sentence marker "।", "?" or "!"). At this preprocessing stage, text is tokenized, stop words are eliminated, and words are

stemmed (Das and Bandyopadhyay, 2010b). Thus, the text in each document is split into fragments and each fragment is represented with a vector of constituent theme words. These text fragments become the nodes V in the document graph.

The similarity between two nodes is expressed as the weight of each edge E of the document graph. A weighted edge is added to the document graph between two nodes if they either correspond to adjacent text fragments in the text or are semantically related by theme words. The weight of an edge denotes the degree of the relationship. The weighted edges not only denote document level similarity between nodes but also inter document level similarity between nodes. Thus to build a document graph G, only the edges with edge weight greater than some predefined threshold value are added to G, which basically constitute the edges E of the graph G.

The Cosine similarity measure has been used here. In cosine similarity, each document d is denoted by the vector $\vec{V}(d)$ derived from d, with each component in the vector for each Theme words. The cosine similarity between two documents (nodes) d1 and d2 is computed using their vector representations $\vec{V}(d1)$ and $\vec{V}(d2)$ as equation (1) and (2) (Described in Section 4). Only a slight change has been done i.e. the dot product of two vectors $\vec{V}(d1) \bullet \vec{V}(d2)$ is defined as $\sum_{i=1}^{M} V(d1)V(d2)$. The Euclidean length of d is defined to be $\sqrt{\sum_{i=1}^{M} \vec{V}_i^{\,2}(d)}$ where M is the total number of documents in the corpus. Theme nodes within a cluster are connected by vertex, weight is calculated by the clustering co-efficient of those theme nodes. No inter cluster vertex are there. Cluster centers are interconnected with weighted vertex. The weight is calculated by cluster distance as measured by cosine similarity measure as discussed earlier.

To better aid our understanding of the automatically determined category relationships we visualized this network using the Fruchterman-Reingold force directed graph layout algorithm (Fruchterman and Reingold, 1991) and the NodeXL network analysis tool (Smith et al., 2009)[5]. A theme relational model graph drawn by NodeXL is shown in Figure 2.

[5] Available from http://www.codeplex.com/NodeXL

6 Summarization System

Present system is an extractive opinion summarization system for Bengali. In the previous sections, we described how to identify theme clusters that relates to different shared topics and subtopics, from a given input document set. But identifying those clusters is not only a step toward generating document level opinionated news summary rather another major step is to extract thematic sentences from each theme cluster that reflects the contextual concise content of the current theme cluster. Extraction of sentences based on their importance in representing the shared subtopic (cluster) is an important issue and it regulates the quality of the output summary. We have used Information Retrieval (IR) based technique to identify the most "informed" sentences from any cluster and it can be termed as IR based cluster center for that particular cluster. With the adaptation of ideas from page rank algorithms (Page et al., 1998), it can be easily observed that a text fragment (sentence) in a document is relevant if it is highly related to many relevant text fragments of other documents in the same cluster. Since, in our document graph structure, the edge score reflects the correlation measure between two nodes, it can be used to identify the most salient/informed sentence from a sentence cluster. We computed the relevance of a node/sentence by summing up the edge scores of those edges connecting the node with other nodes in the same cluster. Then the nodes are given rank according to their calculated relevance scores and the top ranking sentences is selected as the candidate sentence representing the opinion summary. For example four such candidate sentences are shown in Table 7. The words in bold are the theme words based on those theme words the sentences are extracted.

Candidate Sentence	IR Score
মহম্মদ আমিনের মতো **পলিটব্যুরোর** 'নবীনতম' সদস্যকেও কিন্তু বয়সের দিক হইতে নবীন ভাবা কঠিন।	151
এবার চিন্তা আরওএকটু বেশি, কারণ এই **মূল্যবৃদ্ধির** পিছনে যেমন দেশের ভিতরে জিনিসপত্রের জোগান কমে যাওয়া আছে, তেমনই আছে **আন্তর্জাতিক বাজারে মূল্যবৃদ্ধির** প্রবণতা।	167
স্বাধীনতার পর ষাট বছর গত হইল, এখনও প্রায় সকল **সরকারি পরিকল্পনার** পিছনে এই একটিই ভাবাদর্শ কাজ করে: বিভিন্ন **ভোটব্যাঙ্ককে** তুষ্ট করিয়া যেন তেন প্রকারেণ নিজেদের দলীয় স্থিতি নিশ্চিত করা।	130

Table 7: Candidate sentences

Another issue that is very important in summarization is sentence ordering so that the Output summary looks coherent. Once all the relevant sentences are extracted across the input documents, the summarizer has to decide in which order to present them so that the whole text makes sense for the user. We prefer the original order of sentences as they occurred in original document.

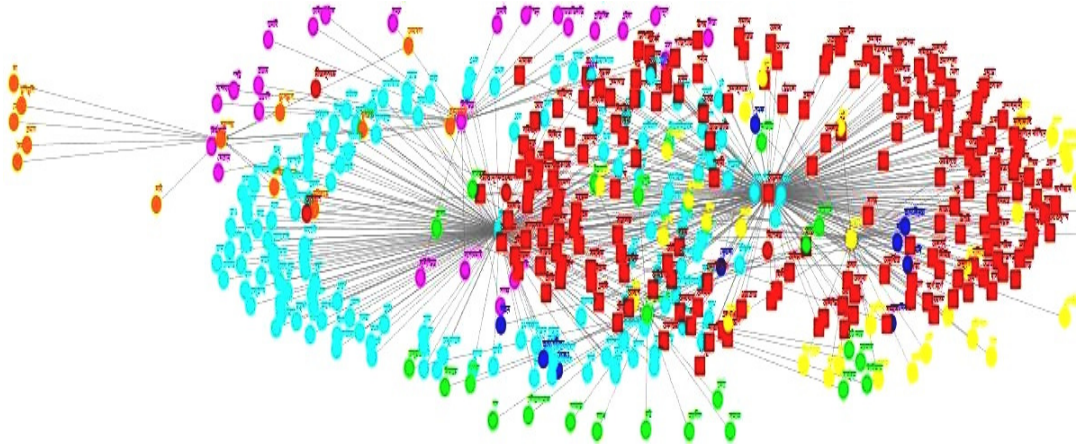


Figure 2: Document Level Theme Relational Graph by NodeXL

7 Experimental Result

The evaluation result of the CRF-based Theme Detection task for Bengali is presented in Table 8. The result is presented individually for every annotators and the overall result of the system.

	Metrics	X	Y	Z	Avg
Theme Detection	**Precision**	87.65%	85.06%	78.06%	83.60%
	Recall	80.78%	76.06%	72.46%	76.44%
	F-Score	84.07%	80.30%	75.16%	79.85%

Table 8: Results of CRF-based Theme Identifier

The evaluation result of subjective sentence identification of the system for opinion summary is in the Table 9.

	Metrics	X	Y	Z	Avg
Summarization	**Precision**	77.65%	67.22%	71.57%	72.15%
	Recall	68.76%	64.53%	68.68%	67.32%
	F-Score	72.94%	65.85%	70.10%	69.65%

Table 9: Final Results subjective sentence identification for summary

8 Error Analysis

The evaluation result of the present summarization system is reasonably good but still not outstanding. During the error analysis we found that the main false hits occurring for subjectivity identifier. It has been reported (Section 2.2) that the recall value of the classifier is higher than its precision. Hence some objective sentences are identified during subjectivity analysis. Some of the sentences get high score during Theme detection or Theme clustering and being included in final summary. Our observation is at least 2-3% sentences are included due to the wrong identification by Subjectivity identifier.

Another vital source of errors occurring in the accuracy level of linguistics resources and tools are the POS tagger, Chunker and Dependency Parser. These linguistics tools are not well performing hence the resultant Theme identification system is missing some of the important theme words. Successive Theme clustering, Document level weighted theme relational model fails to accumulate those important theme expressions. Our observation is at most 3-5% improvement could be possible on final system by granular improvement of every linguistic tool.

9 Conclusion

In this work we have reported our work on single-document opinion summarization for Bengali. The novelty of the proposed technique is the topic based document-level theme relational graphical representation. According to best of our knowledge this is the first attempt on opinion summarization for Bengali. The approach presented here is unique in every aspect as in literature and for a new language like Bengali.

Our next research target is to generate a hierarchical cluster of theme words with time-frame relations. Time-frame relations could be useful for time wise opinion tracking.

References

Carenini Giuseppe, Ng Raymond, and Pauls Adam. Multi-document summarization of evaluative text. In Proceedings of the European Chapter of the Association for Computational Linguistics (EACL), pages 305–312, 2006.

Chesley Paula, Vincent Bruce, Xu Li, and Srihari Rohini. Using verbs and adjectives to automatically classify blog sentiment. In AAAI Symposium on Computational Approaches to Analysing Weblogs (AAAI-CAAW), pages 27–29, 2006.

Das, A. and Bandyopadhyay S. (2009b) Theme Detection an Exploration of Opinion Subjectivity. In Proceeding of Affective Computing & Intelligent Interaction (ACII).

Das, A. and Bandyopadhyay, S. (2009a). Subjectivity Detection in English and Bengali: A CRF-based Approach., In Proceeding of ICON 2009, December 14th-17th, 2009, Hyderabad.

Das, A. and Bandyopadhyay, S. (2010a). SentiWordNet for Bangla. In Knowledge Sharing Event-4: Task 2: Building Electronic Dictionary , February 23th to 24th, 2010, Mysore.

Das, A. and Bandyopadhyay, S. (2010b). Morphologi-cal Stemming Cluster Identification for Bangla., In Knowledge Sharing Event-1: Task 3: Morphologi-cal Analyzers and Generators, January, 2010, Mysore.

Fruchterman Thomas M. J. and Edward M. Reingold. 1991. Graph drawing by force-directed placement. Software: Practice and Experience, 21(11):1129–1164.

Ghosh A., Das A., Bhaskar P., Bandyopadhyay S. (2009). Dependency Parser for Bengali : the JU System at ICON 2009., In NLP Tool Contest ICON 2009, December 14th-17th, 2009a, Hyderabad.

Hatzivassiloglou Vasileios and Wiebe Janyce. Effects of adjective orientation and gradability on sentence subjectivity. In Proceedings of the International Conference on Computational Linguistics (COLING), pages 299-305, 2000.

Hu M. and Liu B.. 2004a. Mining and summarizing-customer reviews. In Proc. of the 10th ACM-SIGKDD Conf., pages 168–177, New York, NY, USA. ACM Press.

Jardine, N. and van Rijsbergen, C. J. (1971). The use of hierarchic clustering in information retrieval. In-formation Storage and Retrieval, 7, 217-240.

Kawai Yukiko, Kumamoto Tadahiko, and Katsumi Tanaka. Fair News Reader: Recommending news articles with different sentiments based on user pre-ference. In Proceedings of Knowledge-Based Intel-ligent Information and Engineering Systems (KES), number 4692 in Lecture Notes in Computer Science, pages 612–622, 2007.

Ku Lun-Wei, Li Li-Ying, Wu Tung-Ho, and Chen Hsin-Hsi. Major topic detection and its application to opinion summarization. In Proceedings of the ACM Special Interest Group on Information Retrieval (SIGIR), pages 627–628, 2005. Poster paper.

Nasukawa Tetsuya and Yi Jeonghee. Sentiment analysis: Capturing favorability using natural language processing. In Proceedings of the Conference on Knowledge Capture (K-CAP), pages 70-77, 2003.

Page Lawrence, Brin Sergey, Rajeev Motwani, and Terry Winograd. 1998. The PageRank Citation Ranking: Bringing Order to the Web. Technical report, Stanford Digital Library Technologies Project.

Resnick Paul, Kuwabara Ko, Zeckhauser Richard, and Friedman Eric. Reputation systems. Communications of the Association for Computing Machinery (CACM), 43(12):45–48, 2000. ISSN 0001-0782.

Smith Marc, Shneiderman Ben, Natasa Milic-Frayling, Eduarda Mendes Rodrigues, Vladimir Barash, Cody Dunne, Tony Capone, Adam Perer, and Eric Gleave. 2009. Analyzing (social media) networks with NodeXL. In C&T '09: Proc. Fourth International Conference on Communities and Technologies, Lecture Notes in Computer Science. Springer.

Willerr, P. (1988). Recent trends in hierarchic document clustering: A critical review. Information Processing and Management, 24(5), 577-597.

Zhou Liang and Hovy Eduard. On the summarization of dynamically introduced information: Online dis-cussions and blogs. In AAAI Symposium on Com-putational Approaches to Analysing Weblogs (AAAI-CAAW), pages 237–242, 2006.

Enhanced Sentiment Learning Using Twitter Hashtags and Smileys

Dmitry Davidov[*1] **Oren Tsur**[*2] **Ari Rappoport**[2]

[1]ICNC / [2]Institute of Computer Science
The Hebrew University
{oren,arir}@cs.huji.ac.il

Abstract

Automated identification of diverse sentiment types can be beneficial for many NLP systems such as review summarization and public media analysis. In some of these systems there is an option of assigning a sentiment value to a single sentence or a very short text.

In this paper we propose a supervised sentiment classification framework which is based on data from Twitter, a popular microblogging service. By utilizing 50 Twitter tags and 15 smileys as sentiment labels, this framework avoids the need for labor intensive manual annotation, allowing identification and classification of diverse sentiment types of short texts. We evaluate the contribution of different feature types for sentiment classification and show that our framework successfully identifies sentiment types of untagged sentences. The quality of the sentiment identification was also confirmed by human judges. We also explore dependencies and overlap between different sentiment types represented by smileys and Twitter hashtags.

1 Introduction

A huge amount of social media including news, forums, product reviews and blogs contain numerous sentiment-based sentences. Sentiment is defined as "a personal belief or judgment that is not founded on proof or certainty"[1]. Sentiment expressions may describe the mood of the writer (happy/sad/bored/grateful/...) or the opinion of the writer towards some specific entity (X is great/I hate X, etc.).

Automated identification of diverse sentiment types can be beneficial for many NLP systems such as review summarization systems, dialogue systems and public media analysis systems. Sometimes it is directly requested by the user to obtain articles or sentences with a certain sentiment value (e.g Give me all positive reviews of product X/ Show me articles which explain why movie X is boring). In some other cases obtaining sentiment value can greatly enhance information extraction tasks like review summarization. While the majority of existing sentiment extraction systems focus on polarity identification (e.g., positive vs. negative reviews) or extraction of a handful of pre-specified mood labels, there are many useful and relatively unexplored sentiment types.

Sentiment extraction systems usually require an extensive set of manually supplied sentiment words or a handcrafted sentiment-specific dataset. With the recent popularity of article tagging, some social media types like blogs allow users to add sentiment tags to articles. This allows to use blogs as a large user-labeled dataset for sentiment learning and identification. However, the set of sentiment tags in most blog platforms is somewhat restricted. Moreover, the assigned tag applies to the whole blog post while a finer grained sentiment extraction is needed (McDonald et al., 2007).

With the recent popularity of the Twitter microblogging service, a huge amount of frequently

* Both authors equally contributed to this paper.

[1]WordNet 2.1 definitions.

Coling 2008: Poster Volume, pages 241–249,
Beijing, August 2010

self-standing short textual sentences (*tweets*) became openly available for the research community. Many of these tweets contain a wide variety of user-defined hashtags. Some of these tags are sentiment tags which assign one or more sentiment values to a tweet. In this paper we propose a way to utilize such tagged Twitter data for classification of a wide variety of sentiment types from text.

We utilize 50 Twitter tags and 15 smileys as sentiment labels which allow us to build a classifier for dozens of sentiment types for short textual sentences. In our study we use four different feature types (punctuation, words, n-grams and patterns) for sentiment classification and evaluate the contribution of each feature type for this task. We show that our framework successfully identifies sentiment types of the untagged tweets. We confirm the quality of our algorithm using human judges.

We also explore the dependencies and overlap between different sentiment types represented by smileys and Twitter tags.

Section 2 describes related work. Section 3 details classification features and the algorithm, while Section 4 describes the dataset and labels. Automated and manual evaluation protocols and results are presented in Section 5, followed by a short discussion.

2 Related work

Sentiment analysis tasks typically combine two different tasks: (1) Identifying sentiment expressions, and (2) determining the polarity (sometimes called *valence*) of the expressed sentiment. These tasks are closely related as the purpose of most works is to determine whether a sentence bears a positive or a negative (implicit or explicit) opinion about the target of the sentiment.

Several works (Wiebe, 2000; Turney, 2002; Riloff, 2003; Whitelaw et al., 2005) use lexical resources and decide whether a sentence expresses a sentiment by the presence of lexical items (sentiment words). Others combine additional feature types for this decision (Yu and Hatzivassiloglou, 2003; Kim and Hovy, 2004; Wilson et al., 2005; Bloom et al., 2007; McDonald et al., 2007; Titov and McDonald, 2008a; Melville et al., 2009).

It was suggested that sentiment words may have different senses (Esuli and Sebastiani, 2006; Andreevskaia and Bergler, 2006; Wiebe and Mihalcea, 2006), thus word sense disambiguation can improve sentiment analysis systems (Akkaya et al., 2009). All works mentioned above identify evaluative sentiment expressions and their polarity.

Another line of works aims at identifying a broader range of sentiment classes expressing various emotions such as happiness, sadness, boredom, fear, and gratitude, regardless (or in addition to) positive or negative evaluations. Mihalcea and Liu (2006) derive lists of words and phrases with happiness factor from a corpus of blog posts, where each post is annotated by the blogger with a mood label. Balog et al. (2006) use the mood annotation of blog posts coupled with news data in order to discover the events that drive the dominant moods expressed in blogs. Mishne (2005) used an ontology of over 100 moods assigned to blog posts to classify blog texts according to moods. While (Mishne, 2005) classifies a blog entry (post), (Mihalcea and Liu, 2006) assign a happiness factor to specific words and expressions. Mishne used a much broader range of moods. Strapparava and Mihalcea (2008) classify blog posts and news headlines to six sentiment categories.

While most of the works on sentiment analysis focus on full text, some works address sentiment analysis in the phrasal and sentence level, see (Yu and Hatzivassiloglou, 2003; Wilson et al., 2005; McDonald et al., 2007; Titov and McDonald, 2008a; Titov and McDonald, 2008b; Wilson et al., 2009; Tsur et al., 2010) among others.

Only a few studies analyze the sentiment and polarity of tweets targeted at major brands. Jansen et al. (2009) used a commercial sentiment analyzer as well as a manually labeled corpus. Davidov et al. (2010) analyze the use of the *#sarcasm* hashtag and its contribution to automatic recognition of sarcastic tweets. To the best of our knowledge, there are no works employing Twitter hashtags to learn a wide range of emotions and the relations between the different emotions.

3 Sentiment classification framework

Below we propose a set of classification features and present the algorithm for sentiment classification.

3.1 Classification features

We utilize four basic feature types for sentiment classification: single word features, n-gram features, pattern features and punctuation features. For the classification, all feature types are combined into a single feature vector.

3.1.1 Word-based and n-gram-based features

Each word appearing in a sentence serves as a binary feature with weight equal to the inverted count of this word in the Twitter corpus. We also took each consecutive word sequence containing 2–5 words as a binary n-gram feature using a similar weighting strategy. Thus n-gram features always have a higher weight than features of their component words, and rare words have a higher weight than common words. Words or n-grams appearing in less than 0.5% of the training set sentences do not constitute a feature. ASCII smileys and other punctuation sequences containing two or more consecutive punctuation symbols were used as single-word features. Word features also include the substituted meta-words for URLs, references and hashtags (see Subsection 4.1).

3.1.2 Pattern-based features

Our main feature type is based on surface patterns. For automated extraction of patterns, we followed the pattern definitions given in (Davidov and Rappoport, 2006). We classified words into high-frequency words (HFWs) and content words (CWs). A word whose corpus frequency is more (less) than F_H (F_C) is considered to be a HFW (CW). We estimate word frequency from the training set rather than from an external corpus. Unlike (Davidov and Rappoport, 2006), we consider all single punctuation characters or consecutive sequences of punctuation characters as HFWs. We also consider URL, REF, and HASHTAG tags as HFWs for pattern extraction. We define a pattern as an ordered sequence of high frequency words and slots for content words. Following (Davidov and Rappoport, 2008), the F_H and F_C thresholds were set to 1000 words per million (upper bound for F_C) and 100 words per million (lower bound for F_H)[2].

The patterns allow 2–6 HFWs and 1–5 slots for CWs. To avoid collection of patterns which capture only a part of a meaningful multiword expression, we require patterns to start and to end with a HFW. Thus a minimal pattern is of the form [HFW] [CW slot] [HFW]. For each sentence it is possible to generate dozens of different patterns that may overlap. As with words and n-gram features, we do not treat as features any patterns which appear in less than 0.5% of the training set sentences.

Since each feature vector is based on a single sentence (tweet), we would like to allow approximate pattern matching for enhancement of learning flexibility. The value of a pattern feature is estimated according the one of the following four scenarios[3]:

$$\begin{cases} \frac{1}{count(p)}: & \text{Exact match – all the pattern components appear in the sentence in correct order without any additional words.} \\ \frac{\alpha}{count(p)}: & \text{Sparse match – same as exact match but additional non-matching words can be inserted between pattern components.} \\ \frac{\gamma * n}{N * count(p)}: & \text{Incomplete match – only } n > 1 \text{ of } N \text{ pattern components appear in the sentence, while some non-matching words can be inserted in-between. At least one of the appearing components should be a HFW.} \\ 0: & \text{No match – nothing or only a single pattern component appears in the sentence.} \end{cases}$$

$0 \leq \alpha \leq 1$ and $0 \leq \gamma \leq 1$ are parameters we use to assign reduced scores for imperfect matches. Since the patterns we use are relatively long, exact matches are uncommon, and taking advantage of partial matches allows us to significantly reduce the sparsity of the feature vectors. We used $\alpha = \gamma = 0.1$ in all experiments.

This pattern based framework was proven efficient for sarcasm detection in (Tsur et al., 2010;

[2]Note that the F_H and F_C bounds allow overlap between some HFWs and CWs. See (Davidov and Rappoport, 2008) for a short discussion.

[3]As with word and n-gram features, the maximal feature weight of a pattern p is defined as the inverse count of a pattern in the complete Twitter corpus.

Davidov et al., 2010).

3.1.3 Efficiency of feature selection

Since we avoid selection of textual features which have a training set frequency below 0.5%, we perform feature selection incrementally, on each stage using the frequencies of the features obtained during the previous stages. Thus first we estimate the frequencies of single words in the training set, then we only consider creation of n-grams from single words with sufficient frequency, finally we only consider patterns composed from sufficiently frequent words and n-grams.

3.1.4 Punctuation-based features

In addition to pattern-based features we used the following generic features: (1) Sentence length in words, (2) Number of "!" characters in the sentence, (3) Number of "?" characters in the sentence, (4) Number of quotes in the sentence, and (5) Number of capitalized/all capitals words in the sentence. All these features were normalized by dividing them by the (maximal observed value *times* averaged maximal value of the other feature groups), thus the maximal weight of each of these features is equal to the averaged weight of a single pattern/word/n-gram feature.

3.2 Classification algorithm

In order to assign a sentiment label to new examples in the test set we use a k-nearest neighbors (kNN)-like strategy. We construct a feature vector for each example in the training and the test set. We would like to assign a sentiment class to each example in the test set. For each feature vector V in the test set, we compute the Euclidean distance to each of the matching vectors in the training set, where matching vectors are defined as ones which share at least one pattern/n-gram/word feature with v.

Let $t_i, i = 1 \ldots k$ be the k vectors with lowest Euclidean distance to v[4] with assigned labels $L_i, i = 1 \ldots k$. We calculate the mean distance $d(t_i, v)$ for this set of vectors and drop from the set up to five outliers for which the distance was more then twice the mean distance. The label assigned to v is the label of the majority of the remaining vectors.

If a similar number of remaining vectors have different labels, we assigned to the test vector the most frequent of these labels according to their frequency in the dataset. If there are no matching vectors found for v, we assigned the default "no sentiment" label since there is significantly more non-sentiment sentences than sentiment sentences in Twitter.

4 Twitter dataset and sentiment tags

In our experiments we used an extensive Twitter data collection as training and testing sets. In our training sets we utilize sentiment hashtags and smileys as classification labels. Below we describe this dataset in detail.

4.1 Twitter dataset

We have used a Twitter dataset generously provided to us by Brendan O'Connor. This dataset includes over 475 million tweets comprising roughly 15% of all public, non-"low quality" tweets created from May 2009 to Jan 2010. Tweets are short sentences limited to 140 UTF-8 characters. All non-English tweets and tweets which contain less than 5 proper English words[5] were removed from the dataset.

Apart of simple text, tweets may contain URL addresses, references to other Twitter users (appear as @<user>) or a content tags (also called *hashtags*) assigned by the tweeter (#<tag>) which we use as labels for our supervised classification framework.

Two examples of typical tweets are: *"#ipad #sucks and 6,510 people agree. See more on Ipad sucks page: http://j.mp/4OiYyg?"*, and *"Pay no mind to those who talk behind ur back, it simply means that u're 2 steps ahead. #ihatequotes"*. Note that in the first example the hashtagged words are a grammatical part of the sentence (it becomes meaningless without them) while *#ihatequotes* of the second example is a mere sentiment label and not part of the sentence. Also note that hashtags can be composed of multiple words (with no spaces).

[4] We used $k = 10$ for all experiments.

[5] Identification of proper English words was based on an available WN-based English dictionary

Category	# of tags	% agreement
Strong sentiment	52	87
Likely sentiment	70	66
Context-dependent	110	61
Focused	45	75
No sentiment	3564	99

Table 1: Annotation results (2 judges) for the 3852 most frequent tweeter tags. The second column displays the average number of tags, and the last column shows % of tags annotated similarly by two judges.

During preprocessing, we have replaced URL links, hashtags and references by URL/REF/TAG meta-words. This substitution obviously had some effect on the pattern recognition phase (see Section 3.1.2), however, our algorithm is robust enough to overcome this distortion.

4.2 Hashtag-based sentiment labels

The Twitter dataset contains above 2.5 million different user-defined hashtags. Many tweets include more than a single tag and 3852 "frequent" tags appear in more than 1000 different tweets. Two human judges manually annotated these frequent tags into five different categories: 1 – strong sentiment (e.g *#sucks* in the example above), 2 – most likely sentiment (e.g., *#notcute*), 3 – context-dependent sentiment (e.g., *#shoutsout*), 4 – focused sentiment (e.g., *#tmobilesucks* where the target of the sentiment is part of the hashtag), and 5 – no sentiment (e.g. *#obama*). Table 1 shows annotation results and the percentage of similarly assigned values for each category.

We selected 50 hashtags annotated "1" or "2" by both judges. For each of these tags we automatically sampled 1000 tweets resulting in 50000 labeled tweets. We avoided sampling tweets which include more than one of the sampled hashtags. As a no-sentiment dataset we randomly sampled 10000 tweets with no hashtags/smileys from the whole dataset assuming that such a random sample is unlikely to contain a significant amount of sentiment sentences.

4.3 Smiley-based sentiment labels

While there exist many "official" lists of possible ASCII smileys, most of these smileys are infrequent or not commonly accepted and used as sentiment indicators by online communities. We used the Amazon Mechanical Turk (AMT) service in order to obtain a list of the most commonly used and unambiguous ASCII smileys. We asked each of ten AMT human subjects to provide at least 6 commonly used ASCII mood-indicating smileys together with one or more single-word descriptions of the smiley-related mood state. From the obtained list of smileys we selected a subset of 15 smileys which were (1) provided by at least three human subjects, (2) described by at least two human subject using the same single-word description, and (3) appear at least 1000 times in our Twitter dataset. We then sampled 1000 tweets for each of these smileys, using these smileys as sentiment tags in the sentiment classification framework described in the previous section.

5 Evaluation and Results

The purpose of our evaluation was to learn how well our framework can identify and distinguish between sentiment types defined by tags or smileys and to test if our framework can be successfully used to identify sentiment types in new untagged sentences.

5.1 Evaluation using cross-validation

In the first experiment we evaluated the consistency and quality of sentiment classification using cross-validation over the training set. Fully automated evaluation allowed us to test the performance of our algorithm under several different feature settings: *Pn+W-M-Pt-*, *Pn+W+M-Pt-*, *Pn+W+M+Pt-*, *Pn-W-M-Pt+* and *FULL*, where $+/-$ stands for utilization/omission of the following feature types: *Pn*:punctuation, *W*:Word, *M*:n-grams (M stands for 'multi'), *Pt*:patterns. *FULL* stands for utilization of all feature types.

In this experimental setting, the training set was divided to 10 parts and a 10-fold cross validation test is executed. Each time, we use 9 parts as the labeled training data for feature selection and construction of labeled vectors and the remaining part is used as a test set. The process was repeated ten times. To avoid utilization of labels as strong features in the test set, we removed all instances of involved label hashtags/smileys from the tweets used as the test set.

Setup	Smileys	Hashtags
random	0.06	0.02
Pn+W-M-Pt-	0.16	0.06
Pn+W+M-Pt-	0.25	0.15
Pn+W+M+Pt-	0.29	0.18
Pn-W-M-Pt+	0.5	0.26
FULL	0.64	0.31

Table 2: Multi-class classification results for smileys and hashtags. The table shows averaged harmonic f-score for 10-fold cross validation. 51 (16) sentiment classes were used for hashtags (smileys).

Hashtags	Avg	#hate	#jealous	#cute	#outrageous
Pn+W-M-Pt-	0.57	0.6	0.55	0.63	0.53
Pn+W+M-Pt-	0.64	0.64	0.67	0.66	0.6
Pn+W+M+Pt-	0.69	0.66	0.67	0.69	0.64
Pn-W-M-Pt+	0.73	0.75	0.7	0.69	0.69
FULL	0.8	0.83	0.76	0.71	0.78
Smileys	Avg	:)	;)	$X($	$: d$
Pn+W-M-Pt-	0.64	0.66	0.67	0.56	0.65
Pn+W+M-Pt-	0.7	0.73	0.72	0.64	0.69
Pn+W+M+Pt-	0.7	0.74	0.75	0.66	0.69
Pn-W-M-Pt+	0.75	0.78	0.75	0.68	0.72
FULL	0.86	0.87	0.9	0.74	0.81

Table 3: Binary classification results for smileys and hashtags. *Avg* column shows averaged harmonic f-score for 10-fold cross validation over *all* 50(15) sentiment hashtags (smileys).

Multi-class classification. Under multi-class classification we attempt to assign a single label (51 labels in case of hashtags and 16 labels in case of smileys) to each of vectors in the test set. Note that the random baseline for this task is 0.02 (0.06) for hashtags (smileys). Table 2 shows the performance of our framework for these tasks.

Results are significantly above the random baseline and definitely nontrivial considering the equal class sizes in the test set. While still relatively low (0.31 for hashtags and 0.64 for smileys), we observe much better performance for smileys which is expected due to the lower number of sentiment types.

The relatively low performance of hashtags can be explained by ambiguity of the hashtags and some overlap of sentiments. Examination of classified sentences reveals that many of them can be reasonably assigned to more than one of the available hashtags or smileys. Thus a tweet *"I'm reading stuff that I DON'T understand again! hahaha...wth am I doing"* may reasonably match tags #sarcasm, #damn, #haha, #lol, #humor, #angry etc. Close examination of the incorrectly classified examples also reveals that substantial amount of tweets utilize hashtags to explicitly indicate the specific hashtagged sentiment, in these cases that no sentiment value could be perceived by readers unless indicated explicitly, e.g. *"De Blob game review posted on our blog. #fun"*. Obviously, our framework fails to process such cases and captures noise since no sentiment data is present in the processed text labeled with a specific sentiment label.

Binary classification. In the binary classification experiments, we classified a sentence as either appropriate for a particular tag or as not bearing any sentiment[6]. For each of the 50 (15) labels for hashtags (smileys) we have performed a binary classification when providing as training/test sets only positive examples of the specific sentiment label together with non-sentiment examples. Table 3 shows averaged results for this case and specific results for selected tags. We can see that our framework successfully identifies diverse sentiment types. Obviously the results are much better than those of multi-class classification, and the observed > 0.8 precision confirms the usefulness of the proposed framework for sentiment classification of a variety of different sentiment types.

We can see that even for binary classification settings, classification of smiley-labeled sentences is a substantially easier task compared to classification of hashtag-labeled tweets. Comparing the contributed performance of different feature types we can see that punctuation, word and pattern features, each provide a substantial boost for classification quality while we observe only a marginal boost when adding n-grams as classification features. We can also see that pattern features contribute the performance more than all other features together.

5.2 Evaluation with human judges

In the second set of experiments we evaluated our framework on a test set of unseen and untagged tweets (thus tweets that were not part of the train-

[6]Note that this is a useful application in itself, as a filter that extracts sentiment sentences from a corpus for further focused study/processing.

ing data), comparing its output to tags assigned by human judges. We applied our framework with its FULL setting, learning the sentiment tags from the training set for hashtags and smileys (separately) and executed the framework on the reduced Tweeter dataset (without untagged data) allowing it to identify at least five sentences for each sentiment class.

In order to make the evaluation harsher, we removed all tweets containing at least one of the relevant classification hashtags (or smileys). For each of the resulting 250 sentences for hashtags, and 75 sentences for smileys we generated an 'assignment task'. Each task presents a human judge with a sentence and a list of ten possible hashtags. One tag from this list was provided by our algorithm, 8 other tags were sampled from the remaining 49 (14) available sentiment tags, and the tenth tag is from the list of frequent non-sentiment tags (e.g. *travel* or *obama*). The human judge was requested to select the 0-2 most appropriate tags from the list. Allowing assignment of multiple tags conforms to the observation that even short sentences may express several different sentiment types and to the observation that some of the selected sentiment tags might express similar sentiment types.

We used the Amazon Mechanical Turk service to present the tasks to English-speaking subjects. Each subject was given 50 tasks for Twitter hashtags or 25 questions for smileys. To ensure the quality of assignments, we added to each test five manually selected, clearly sentiment bearing, assignment tasks from the tagged Twitter sentences used in the training set. Each set was presented to four subjects. If a human subject failed to provide the intended "correct" answer to at least two of the control set questions we reject him/her from the calculation. In our evaluation the algorithm is considered to be correct if one of the tags selected by a human judge was also selected by the algorithm. Table 4 shows results for human judgement classification. The agreement score for this task was $\kappa = 0.41$ (we consider agreement when at least one of two selected items are shared).

Table 4 shows that the majority of tags selected by humans matched those selected by the algorithm. Precision of smiley tags is substantially higher than of hashtag labels, due to the lesser number of possible smileys and the lesser ambiguity of smileys in comparison to hashtags.

Setup	% Correct	% No sentiment	Control
Smileys	84%	6%	92%
Hashtags	77%	10%	90%

Table 4: Results of human evaluation. The second column indicates percentage of sentences where judges find no appropriate tags from the list. The third column shows performance on the control set.

Hashtags	#happy	#sad	#crazy	# bored
#sad	0.67	-	-	-
#crazy	0.67	0.25	-	-
#bored	0.05	0.42	0.35	-
#fun	1.21	0.06	1.17	0.43
Smileys	**:)**	**;)**	**: (**	***X(***
;)	3.35	-	-	-
: (	3.12	0.53	-	-
X(	1.74	0.47	2.18	-
: *S*	1.74	0.42	1.4	0.15

Table 5: Percentage of co-appearance of tags in tweeter corpus.

5.3 Exploration of feature dependencies

Our algorithm assigns a single sentiment type for each tweet. However, as discussed above, some sentiment types overlap (e.g., *#awesome* and *#amazing*). Many sentences may express several types of sentiment (e.g., *#fun* and *#scary* in *"Oh My God http://goo.gl/fb/K2N5z #entertainment #fun #pictures #photography #scary #teaparty"*). We would like to estimate such inter-sentiment dependencies and overlap automatically from the labeled data. We use two different methods for overlap estimation: tag co-occurrence and feature overlap.

5.3.1 Tag co-occurrence

Many tweets contain more than a single hashtag or a single smiley type. As mentioned, we exclude such tweets from the training set to reduce ambiguity. However such tag co-appearances can be used for sentiment overlap estimation. We calculated the relative co-occurrence frequencies of some hashtags and smileys. Table 5 shows some of the observed co-appearance ratios. As expected some of the observed tags frequently co-appear with other similar tags.

Hashtags	#happy	#sad	#crazy	# bored
#sad	12.8	-	-	-
#crazy	14.2	3.5	-	-
#bored	2.4	11.1	2.1	-
#fun	19.6	2.1	15	4.4
Smileys	:)	;)	:(	$X($
;)	35.9	-	-	-
:(	31.9	10.5	-	-
$X($	8.1	10.2	36	-
$:S$	10.5	12.6	21.6	6.1

Table 6: Percentage of shared features in feature vectors for different tags.

Interestingly, it appears that a relatively high ratio of co-appearance of tags is with opposite meanings (e.g., *"#ilove eating but #ihate feeling fat lol"* or *"happy days of training going to end in a few days #sad #happy"*). This is possibly due to frequently expressed contrast sentiment types in the same sentence – a fascinating phenomena reflecting the great complexity of the human emotional state (and expression).

5.3.2 Feature overlap

In our framework we have created a set of feature vectors for each of the Twitter sentiment tags. Comparison of shared features in feature vector sets allows us to estimate dependencies between different sentiment types even when direct tag co-occurrence data is very sparse. A feature is considered to be shared between two different sentiment labels if for both sentiment labels there is at least a single example in the training set which has a positive value of this feature. In order to automatically analyze such dependencies we calculate the percentage of shared Word/n-gram/Pattern features between different sentiment labels. Table 6 shows the observed feature overlap values for selected sentiment tags.

We observe the trend of results obtained by comparison of shared feature vectors is similar to those obtained by means of label co-occurrence, although the numbers of the shared features are higher. These results, demonstrating the pattern-based similarity of conflicting, sometimes contradicting, emotions are interesting from a psychological and cognitive perspective.

6 Conclusion

We presented a framework which allows an automatic identification and classification of various sentiment types in short text fragments which is based on Twitter data. Our framework is a supervised classification one which utilizes Twitter hashtags and smileys as training labels. The substantial coverage and size of the processed Twitter data allowed us to identify dozens of sentiment types without any labor-intensive manually labeled training sets or pre-provided sentiment-specific features or sentiment words.

We evaluated diverse feature types for sentiment extraction including punctuation, patterns, words and n-grams, confirming that each feature type contributes to the sentiment classification framework. We also proposed two different methods which allow an automatic identification of sentiment type overlap and inter-dependencies.

In the future these methods can be used for automated clustering of sentiment types and sentiment dependency rules. While hashtag labels are specific to Twitter data, the obtained feature vectors are not heavily Twitter-specific and in the future we would like to explore the applicability of Twitter data for sentiment multi-class identification and classification in other domains.

References

Akkaya, Cem, Janyce Wiebe, and Rada Mihalcea. 2009. Subjectivity word sense disambiguation. In *EMNLP*.

Andreevskaia, A. and S. Bergler. 2006. Mining wordnet for fuzzy sentiment: Sentiment tag extraction from wordnet glosses. In *EACL*.

Balog, Krisztian, Gilad Mishne, and Maarten de Rijke. 2006. Why are they excited? identifying and explaining spikes in blog mood levels. In *EACL*.

Bloom, Kenneth, Navendu Garg, and Shlomo Argamon. 2007. Extracting appraisal expressions. In *HLT/NAACL*.

Davidov, D. and A. Rappoport. 2006. Efficient unsupervised discovery of word categories using symmetric patterns and high frequency words. In *COLING-ACL*.

Davidov, D. and A. Rappoport. 2008. Unsupervised discovery of generic relationships using pattern clusters and its evaluation by automatically generated sat analogy questions. In *ACL*.

Davidov, D., O. Tsur, and A. Rappoport. 2010. Semi-supervised recognition of sarcastic sentences in twitter and amazon. In *CoNLL*.

Esuli, Andrea and Fabrizio Sebastiani. 2006. Sentiwordnet: A publicly available lexical resource for opinion mining. In *LREC*.

Jansen, B.J., M. Zhang, K. Sobel, and A. Chowdury. 2009. Twitter power: Tweets as electronic word of mouth. *Journal of the American Society for Information Science and Technology*.

Kim, S.M. and E. Hovy. 2004. Determining the sentiment of opinions. In *COLING*.

McDonald, Ryan, Kerry Hannan, Tyler Neylon, Mike Wells, and Jeff Reynar. 2007. Structured models for fine-to-coarse sentiment analysis. In *ACL*.

Melville, Prem, Wojciech Gryc, and Richard D. Lawrence. 2009. Sentiment analysis of blogs by combining lexical knowledge with text classification. In *KDD*. ACM.

Mihalcea, Rada and Hugo Liu. 2006. A corpus-based approach to finding happiness. In *In AAAI 2006 Symposium on Computational Approaches to Analysing Weblogs*. AAAI Press.

Mishne, Gilad. 2005. Experiments with mood classification in blog posts. In *Proceedings of the 1st Workshop on Stylistic Analysis Of Text*.

Riloff, Ellen. 2003. Learning extraction patterns for subjective expressions. In *EMNLP*.

Strapparava, Carlo and Rada Mihalcea. 2008. Learning to identify emotions in text. In *SAC*.

Titov, Ivan and Ryan McDonald. 2008a. A joint model of text and aspect ratings for sentiment summarization. In *ACL/HLT*, June.

Titov, Ivan and Ryan McDonald. 2008b. Modeling online reviews with multi-grain topic models. In *WWW*, pages 111–120, New York, NY, USA. ACM.

Tsur, Oren, Dmitry Davidov, and Ari Rappoport. 2010. Icwsm – a great catchy name: Semi-supervised recognition of sarcastic sentences in product reviews. In *AAAI-ICWSM*.

Turney, Peter D. 2002. Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. In *ACL '02*, volume 40.

Whitelaw, Casey, Navendu Garg, and Shlomo Argamon. 2005. Using appraisal groups for sentiment analysis. In *CIKM*.

Wiebe, Janyce and Rada Mihalcea. 2006. Word sense and subjectivity. In *COLING/ACL*, Sydney, AUS.

Wiebe, Janyce M. 2000. Learning subjective adjectives from corpora. In *AAAI*.

Wilson, Theresa, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *HLT/EMNLP*.

Wilson, Theresa, Janyce Wiebe, and Paul Hoffmann. 2009. Recognizing contextual polarity: An exploration of features for phrase-level sentiment analysis. *Computational Linguistics*, 35(3):399–433.

Yu, Hong and Vasileios Hatzivassiloglou. 2003. Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. In *EMNLP*.

Topic models for meaning similarity in context

Georgiana Dinu
Dept. of Computational Linguistics
Saarland University
dinu@coli.uni-sb.de

Mirella Lapata
School of Informatics
University of Edinburgh
mlap@inf.ed.ac.uk

Abstract

Recent work on distributional methods for similarity focuses on using the context in which a target word occurs to derive context-sensitive similarity computations. In this paper we present a method for computing similarity which builds vector representations for words *in context* by modeling senses as latent variables in a large corpus. We apply this to the Lexical Substitution Task and we show that our model significantly outperforms typical distributional methods.

1 Introduction

Distributional methods for word similarity ((Landauer and Dumais, 1997), (Schuetze, 1998)) are based on co-occurrence statistics extracted from large amounts of text. Typically, each word is assigned a representation as a point in a high-dimensional space, where the dimensions represent contextual features such as co-occurring words. Following this, meaning relatedness scores are computed by using various similarity measures on the vector representations.

One of the major issues that all distributional methods have to face is sense ambiguity. Since vector representations reflect *mixtures of uses* additional methods have to be employed in order to capture specific meanings of a word in context. Consider the occurrence of verb *shed* in the following SemEval 2007 Lexical Substitution Task (McCarthy and Navigli, 2007) example:

Cats in the latent phase only have the virus internally , but feel normal and do not ***shed*** *the virus to other cats and the environment .*

Human participants in this task provided words such as *transmit* and *spread* as good substitutes for *shed* in this context, however a vector space representation of *shed* will not capture this infrequent sense.

For these reasons, recent work on distributional methods for similarity such as (Mitchell and Lapata, 2008) (Erk and Padó, 2008) (Thater et al., 2009) focuses on using the context in which a target word occurs to derive context-sensitive similarity computations.

In this paper we present a method for computing similarity which builds vector representations for words *in context*. Most distributional methods so far extract representations from large texts, and only as a follow-on step they either 1) alter these in order to reflect a *disambiguated* word (such as (Erk and Padó, 2008)) or 2) directly asses the appropriateness of a similarity judgment, given a specific context (such as (Pantel et al., 2007)). Our approach differs from this as we assume ambiguity of words at the, initial, acquisition step, by encoding senses of words as a hidden variable in the text we process.

In this paper we focus on a particular distributional representation inspired by (Lin and Pantel, 2001a) and induce context-sensitive similarity between phrases represented as paths in dependency graphs. It is inspired by recent work on topic models and it deals with sense-ambiguity in a natural manner by modeling senses as latent variables in a large corpus. We apply this to the Lexical Substitution Task and we show that our model outperforms the (Lin and Pantel, 2001a) method by inducing context-appropriate similarity judgments.

Coling 2008: Poster Volume, pages 250–258,
Beijing, August 2010

2 Related work

Discovery of Inference Rules from Text (DIRT) A popular distributional method for meaning relatedness is the DIRT algorithm for extracting inference rules (Lin and Pantel, 2001a). In this algorithm a $pattern$ is a noun-ending path in a dependency graph and the goal is to acquire pairs of patterns for which entailment holds (in at least one direction) such as *(X solve Y, X find solution to Y)*.

The method can be seen a particular instance of a vector space. Each pattern is represented by the sets of its left hand side (X) and right hand side (Y) noun fillers in a large corpus. Two patterns are compared in the X-filler space, and correspondingly in the Y-filler space by using the Lin similarity measure:

$$\mathrm{sim}_{\mathrm{Lin}}(v, w) = \frac{\sum_{i \in I(v) \cap I(w)} (v_i + w_i)}{\sum_{i \in I(v)} v_i + \sum_{l \in I(w)} w_i}$$

where values in v and w are point-wise mutual information, and $I(\cdot)$ gives the indices of positive values in a vector.

The final similarity score between two patterns is obtained by multiplying the X and Y similarity scores. Table 1 shows a fragment of a DIRT-like vector space.

	..	*case*	*problem*	..
(*X solve Y,* ***Y***)	..	6.1	4.4	..
(*X settle Y,* ***Y***)	..	5.2	5.9	..

Table 1: DIRT-like vector representation in the **Y**-filler space. The values represent mutual information.

Further on, this similarity method is used for the task of paraphrasing. A total set of patterns is extracted from a large corpus and each of them can be paraphrased by returning its most similar patterns, according to the similarity score. Although relatively accurate[1], it has been noted (Lin and Pantel, 2001b) that the paraphrases extracted this way reflect, as expected, various meanings, and that a context-sensitive representation would be appropriate.

[1]Precision is estimated to lie around 50% for the most confident paraphrases

Context-sensitive extensions of DIRT (Pantel et al., 2007) and (Basili et al., 2007) focus on making DIRT rules context-sensitive by attaching appropriate semantic classes to the X and Y slots of an inference rule. For this purpose, the initial step in their methods is to acquire an inference rule database, using the DIRT algorithm. Following this, given an inference rule, they identify semantic classes for the X and Y fillers which make the application of the rule appropriate. For this (Pantel et al., 2007) build a set of semantic classes using WordNet in one case and CBC clustering algorithm in the other; for each rule, they use the overlap of the fillers found in the input corpus as an indicator of the correct semantic classes. The same idea is used in (Basili et al., 2007) where, this time, the X and Y fillers are clustered for each rule individually; these nouns are clustered using an LSA-vector representation extracted from a large corpus.

(Connor and Roth, 2007) take a slightly different approach as they attempt to classify the context of a rule as appropriate or not, again using the overlap of fillers as an indicator. They all show improvement over DIRT by evaluating on occurrences of rules in context which are annotated as correct/incorrect by human participants. On a common data set (Pantel et al., 2007) and (Basili et al., 2007) achieve significant improvements over DIRT at 95% confidence level when employing the clustering methods. (Szpektor et al., 2008) propose a general framework for these methods and show that some of these settings obtain significant (level 0.01) improvements over the DIRT algorithm on data derived from the ACE 2005 event detection task.

Related work on topic models Topic models have been previously used for semantic tasks. Work such as (Cai et al., 2007) or (Boyd-Graber et al., 2007) use the document-level topics extracted with Latent Dirichlet Allocation (LDA) as indicators of meanings for word sense disambiguation. More related to our work are (Brody and Lapata, 2009) or (Toutanova and Johnson, 2008) who use LDA-based models which induce latent variables from task-specific data rather than from simple documents.

(Brody and Lapata, 2009) apply such a model for word sense induction on a set of 35 target nouns. They assume senses as latent variables and context features as observations; unlike our model they induce local senses specific to every target word by estimating separate models with the final goal of explicitly inducing word senses.

(Toutanova and Johnson, 2008) use an LDA-based model for semi-supervised part-of-speech tagging. They build a word context model in which each token involves: generating a distribution over tags, sampling a tag, and finally generating context words according to a tag-specific word distribution (context words are observations). Their model achieves highest performance when combined with a ambiguity class component which uses a dictionary for possible tags of target words.

Both these papers show improvements over state-of-the-art systems for their tasks.

3 Generative model for similarity in context

We develop a method for computing similarity of patterns in context, i.e. patterns with instantiated X and Y values. We do not enhance the representation of an inference rule with sense (context-appropriateness) information but rather focus on the task of assigning similarity scores to such pairs of *instantiated* patterns. Unlike previous work, we do not employ any other additional resources, investigating this way whether structurally richer information can be learned from the same input co-occurrence matrix as the original DIRT method.

Our model, as well as the DIRT algorithm, uses *context* information extracted from large corpora to learn similarities between *patterns*; however ideally we would like to learn contextual preferences (or, in general, some form of sense-disambiguation) for these patterns. This is achieved in our model by assuming an intermediate layer consisting of *meanings* (senses): the *context* surrounding a pattern is indicative of *meanings*, and preference for some *meanings* gives the characterization of a *pattern*.

For this we use a generative model inspired by Latent Dirichlet Allocation (Blei et al., 2003) (Griffiths and Steyvers, 2004) which is successfully employed for modeling collections of documents and the underlying topics which occur in them. The statistical model is characterized by the following distributions:

$w_i \vert z_i, \phi^{z_i}$	$Discrete(\phi^{z_i})$
ϕ^z	$Dirichlet(\beta)$
$z_i \vert \theta^p$	$Discrete(\theta^p)$
θ^p	$Dirichlet(\alpha)$

θ^p is the distribution over meanings associated to a pattern p and ϕ^z is the distribution over words associated to a meaning z. The occurrence of each filler word w_i with a pattern p, is then generated by sampling 1) a meaning conditioned on the meaning distribution associated to p: $z_i|\theta^p$ and 2) a word conditioned on the word distribution associated to the meaning z_i: $w_i|z_i, \phi^{z_i}$. θ^p and ϕ^z are assumed to be Dirichlet distributions with parameters α and β.

The set of context words (X and Y fillers) occurring with a pattern p form the *document* (in LDA terms) associated to a pattern p. Table 2 lists a fragment of the document associated to pattern *X solve Y*. These are built simply by listing for each pattern, occurrence counts with specific filler words. Since we want our model to differentiate between X and Y fillers, words occurring as fillers are made disjoint by adding a corresponding suffix.

X solve Y
we-X:122, country-X:89, government-X:82, it-X:69,..., problem-Y:1088, issue-Y:134, crisis-Y:99, dispute-Y:78,...

Table 2: Fragments of the *document* associated to *X solve Y*. *we-X: 122* indicates that *X solve Y* occurs 122 times with we as an X filler.

The total set of such *documents* extracted from a large corpus is then used for estimating the model. We use Gibbs sampling[2] and the result is a set of samples from $P(z|w)$ (i.e. meaning assignments for each occurring filler word) from which θ^p (pattern-meaning distributions) and ϕ^z(meaning-word distributions) can be estimated.

Our model has the advantage that, once these

[2] *http://gibbslda.sourceforge.net/*

distributions are estimated, given a pattern p and a context w_n, *in-context* vector representations can be built in a straightforward manner.

Meaning representation in-context Let K be the assumed number of meanings, $(z_1, ..., z_K)$. We associate to a pattern in context (p,w_n), the ***K***-dimensional vector containing for each meaning z_i ($i : 1..K$), the probability of z_i, conditioned on pattern p and context word w_n:

$$vec(p, w_n) = (P(z_1|w_n, p), ..., P(z_K|w_n, p)) \quad (1)$$

where,

$$P(z_i|w_n, p) = \frac{P(z_i, p)P(w_n|z_i)}{\Sigma_{i=1}^{K} P(z_i, p)P(w_n|z_i)} \quad (2)$$

This is the probability that w_n is generated by meaning z_i conditioned on p, therefore, the probability that pattern p has meaning z_i in context w_n, exactly the concept we want to model.

Meaning representation out-of-context We can also associate to pattern p an *out-of-context* vector representation: the ***K***-dimensional vector representing its distribution over meanings:

$$vec(p) = (P(z_1|p), ..., P(z_K|p)) \quad (3)$$

This can be seen as a dimensionality reduction method, since we bring vector representations to a lower dimensional space over (ideally) meaningful concepts.

From the generative model we obtain the desired distributions $P(z_i|p) = \theta_i^p$ and $P(w_n|z_i) = \phi_n^{z_i}$.[3]

Computing similarity between patterns The similarity between patterns occurring with X and Y filler-words is computed following (Lin and Pantel, 2001a) by multiplying the similarities obtained separately in the X and Y spaces.:

$$\begin{aligned} sim((w_{X1}, p_1, w_{Y1})(w_{X2}, p_2, w_{Y2})) = \\ sim(vec(p_1, w_{X1}), vec(p_2, w_{X2}))* \\ sim(vec(p_1, w_{Y1}), vec(p_2, w_{Y2})) \end{aligned} \quad (4)$$

[3]For similarity in context, we use the conditional $P(z_i|p)$ instead of the joint $P(z_i, p)$ which is computationally equivalent for the paraphrasing setting.

$we \xleftarrow{subj} make \xrightarrow{dobj} statement$	
$we \xleftarrow{subj} give \xrightarrow{dobj} statement$	*good*
$we \xleftarrow{subj} prepare \xrightarrow{dobj} statement$	*bad*

Table 3: Development set: good/bad substitutes for $we \xleftarrow{subj} make \xrightarrow{dobj} statement$

Out-of-context similarity is defined in a straightforward manner:

$$sim(p_1, p_2) = sim(vec(p_1,), vec(p_2)) \quad (5)$$

4 Evaluation setup

In this paper we evaluate our model on computing similarities between pairs of the type $(X, pattern, Y), (X, pattern', Y)$ where two different patterns are compared in identical contexts. For this we use the Semeval Lexical Substitution dataset, which requires human participants to provide substitutes for a set of target words occurring in different contexts. This section describes the evaluation methodology for this data as well as the automatically generated data set we use for development.

Development set For finding good model parameters, we use the SemCor corpus providing text in which all content words are tagged with WordNet 1.6 senses. We used this data in the following manner: We parse the text using Stanford parser and extract occurrences of triples *(X, pattern, Y)*. Given these triples we generate *good* and *bad* substitutes for them: the *good* substitutes are generated by replacing the words occurring in the patterns with sense-appropriate synonyms, while *bad* ones are obtained by substitution with synonyms corresponding to the rest of the senses (the wrong senses). The synonyms are extracted from WordNet 1.6 synsets using the sense annotation present in the text.

For evaluation we feed the models pairs of instantiated patterns. One of them is the original phrase encountered in the data, and the other one is a *good/bad* substitute for it. Table 3 shows an example of the data.

We evaluate the output of a system by requiring that, for each instance, every good substitute is scored more similar to the original phrase than

every bad substitute. This leads to an accuracy score which can be compared against a random baseline of 50%.

The data set obtained is far from being a very reliable resource for the task of lexical substitution, however this method of generating data has the advantage of producing a large number of instances which can be easily acquired from any sense-annotated data set. In our experiments we use the Brown2 fragment from which we extract over 3000 instances of patterns in context.

Lexical substitution task The Lexical Substitution Task (McCarthy and Navigli, 2007) presents 5 annotators with a set of target words, each in different context sentences. The task requires the participants to provide appropriate substitute words for each occurrence of the target words.

We use this data similarly to (Erk and Padó, 2008) and (Thater et al., 2009) and for each target word, we pool together all the substitutes given for *all* context sentences. Similarly to the SemCor data, we do not use the entire sentence as a context as we extract only *patterns* containing target words together with their X and Y fillers. The models assign similarity scores to each candidate by comparing them to the pattern occurring in the original sentence. A ranked list of candidates is obtained which in turn is compared with the substitutes provided by the participants. Table 4 gives an example of this data set (for each substitute we list the number of participants providing it).

To evaluate the performance of a model we employ two similarity measures, which capture different aspects of the task. Kendall τ rank coefficient measures the correlation between two ranks; since the gold ranking is usually only a partial order, we use τ_b which makes adjustments for ties. We employ a second evaluation measure: Generalized Average Precision (Kishida, 2005). This is a measure inspired from information retrieval and has been previously used for evaluating this task (Thater et al., 2009). It evaluates a system on its ability to retrieve correct substitutes using the gold ranking together with the associated confidence scores. The confidence scores are in turn determined by the number of people providing each substitute.

pattern	*human substitutes*
$study \xleftarrow{subj} shed \xrightarrow{dobj} light$	throw 3, reveal 2, shine 1
$cat \xleftarrow{subj} shed \xrightarrow{dobj} virus$	spread 2, pass 2, transmit 2, emit 1

Table 4: Lexical substitution data set: target verb *shed*

5 Experiments

5.1 Model selection

The data we use to estimate our models is extracted from a GigaWord fragment containing approximately 100 million tokens. We parse the text with Stanford dependency parser to obtain dependency graphs from which we extract paths together with counts of their left and right fillers. We extract paths containing at most four words, including the two noun anchors. Furthermore we impose a frequency threshold on patterns and words, leading us to a collection of ≈80 000 paths, with filler nouns over a vocabulary of ≈40 000 words.

We estimate a total number of 20 models. We set $\beta = 0.01$ as previous work (Wang et al., 2009) reports good results with this value. For parameter α we test 4 settings: $\alpha_1 = \frac{2}{K}$ and $\alpha_4 = \frac{50}{K}$ which are reported in the literature as good ((Porteous et al., 2008) and (Griffiths and Steyvers, 2004)), as well as 2 intermediate values: $\alpha_2 = \frac{5}{K}$ and $\alpha_3 = \frac{10}{K}$. We test a set of 5 K values: $\{800, 1000, 1200, 1400, 1600\}$. These are chosen to be large since they represent the global set of meanings shared by all the patterns in the collection.

As vector similarity measure we test scalar product (*sp*), which in our model is interpreted as the probability that two patterns share a common meaning. Additionally we test cosine (cos) similarity and inverse Jensen-Shannon (JS) divergence, which is a popular measure for comparing probability distributions:

$$JSD(v, w) = \frac{1}{2}KLD(v|m) + \frac{1}{2}KLD(w|m)$$

with $m = \frac{1}{2}(v + w)$ and KLD the standard Kullback-Leibler divergence: $KLD(v|w) = \Sigma_i v_i log(\frac{v_i}{w_i})$.

We perform both in-context (using eq. (4)) as well as out-of-context computations (eq. (5)). Similarly to previous work (Erk and Padó, 2008), we observe that comparing a contextualized representation against a non-contextualized one brings significant improvements over comparing two representations in context. We assume this is specific to the type of data we work with, in which two patterns are compared in an identical context, rather than across different contexts; we therefore compute context-sensitive similarities by contextualizing just the target word.

Number of topics Although the parameters cover relatively large ranges the models perform surprisingly similar across different α and K values, as well as across all three similarity measures. For sp similarity, the accuracy scores we obtain are in the range [56.5-59.5] with a average deviation from the mean of just 0.8%; similar figures are obtained using the other similarity measures. Figure 1 plots the average of the accuracy scores using sp as similarity measure, across different number of topics. A small preference for higher K values is observed, all models performing consistently good at 1200, 1400 and 1600 topics.

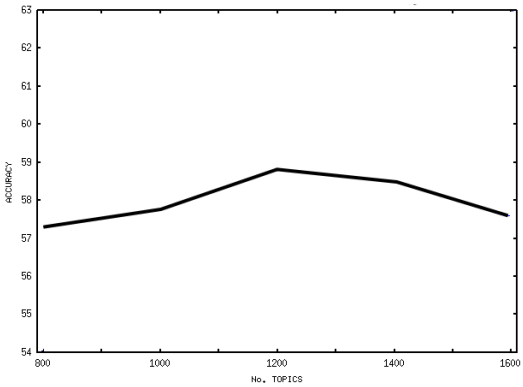


Figure 1: Average accuracy across the 5 K values.

Mixture models This leads us to attempting a very simple mixture model, which computes the similarity score between two patterns as the average similarity obtained across a number of models. For each α setting, we mix models across the three best topic numbers: $\{1200, 1400, 1600\}$. In Figure 2 we plot this mixture model together with the three single ones, at each α value. It can be

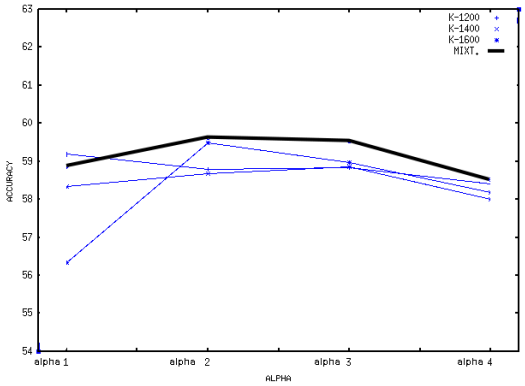


Figure 2: Mixture model $\{1200, 1400, 1600\}$ (bold) vs. the three individual models, across the 4 α values.

noticed that the mixture model improves over all three single models for three out of the four α values.

In-context vs. out-of-context computations Further on we compare *in-context* versus *out-of-context* computations. The similarity measures exhibit significant differences in regard to this aspect. In Figure 3 we plot *in-context* vs. *out-of-context* computations using scalar product (left) and JS (right) with the mixture model previously defined, plotted at different α values. For sp *in-context* computations significantly outperform *out-of-context* ones and the two intermediate alpha values seem to be the best. However for JS similarity the *out-of-context* computations are significantly better and a clear preference for smaller α values can be observed.

Finally, on the test data, we use the following models (where $GM_{mixt/sing,sim}$ stands for a *mixture* or *single* model with similarity measure *sim*):

- $GM_{mixt,sp/cos}$
  mixt($\{1200, 1400, 1600\}$x$\{\alpha_2, \alpha_3\}$)
- $GM_{mixt,js}$
  mixt($\{1200, 1400, 1600\}$x$\{\alpha_1, \alpha_2\}$)
- $GM_{sing,sp}$: (1600, $\alpha 2$)
- $GM_{sing,cos/js}$: (1200, $\alpha 1$)

The mixture models are build based on the observations previously made while the single mod-

Model	*In-context*	*Out-of-context*
$GM_{mixt,sp}$	59.89	58.68
$GM_{mixt,cos}$	59.50	58.67
$GM_{mixt,js}$	59.73	60.68
$GM_{sing,sp}$	59.48	58.86
$GM_{sing,cos}$	59.43	57.87
$GM_{sing,js}$	58.65	59.36

Table 5: Accuracy results on development set

els are the best performing ones, for each similarity measure. The accuracy scores obtained with these models are given in Table 5. Mixture models generally outperform single ones and *in-context* computations outperform *out-of-context* ones for sp and cos. The best results on the development set are however achieved by *out-of-context* models using JS as similarity measure.

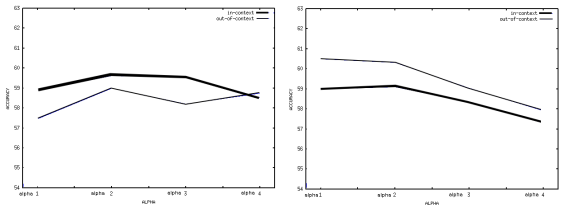


Figure 3: *In-context* (bold) vs. *out-of-context* computations across the 4 α values using scalar product (left) and JS (right)

5.2 Results

Table 6 shows the results for the Lexical Substitution data set. We use the subset of the data containing sentences in which the target word is part of a syntactic path which is present in the total collection of patterns. This leads to a set containing 165 instances of patterns in context, most of these containing target verbs.

Since sp and cos measures perform very similarly we only list results with cosine similarity measure. In addition to the models with settings determined on the development set, we also test a very simple mixture model: $GM_{mixt-all,sim}$. This simply averages over *all* 20 configurations and its purpose is to investigate the necessity of a carefully selected mixture model.

It can be noticed that all GM mixture models outperform DIRT, which is reflected in both

Model	τ_b	GAP
Random	0.0	34.91
DIRT	14.53	48.06
$\mathbf{GM_{mixt,cos}}$	**22.35**	**52.04**
$GM_{mixt,js}$	18.17	50.80
$\mathbf{GM_{mixt-all,cos}}$	**20.42**	**51.13**
$GM_{mixt-all,js}$	19.03	51.15
$GM_{sing,cos}$	15.10	48.20
$GM_{sing,js}$	14.17	47.97

Table 6: Results on Lexical Substitution data

similarity measures. Notably the very simple model which averages all the configurations implemented is surprisingly performant. Using randomized significance testing we obtained that $GM_{mixt,cos}$ is significantly better than DIRT at p level 1e-03 on both GAP and τ_b. $GM_{mixt-all,cos}$ outperforms DIRT at level 0.05.

In terms of similarity measures, the observations made on the development set hold, as for the *in-context* computations cos and sp outperform JS. However, unlike on the development data, the single models perform much worse than the mixture ones which can indicate that the development set is not perfectly suited for choosing model parameters.

Out-of-context computations for all models and all similarity measures are significantly outperformed, leading to scores in ranges [11-14] τ_b and [45-48] GAP.

In Table 7 we list the rankings produced by three models for the target word $shed$ in context $virus \xleftarrow{obj} shed \xrightarrow{prep} to \xrightarrow{pobj} cat$. As it can be observed, the model performing context-sensitive computations $GM_{mixt,cos}$-*in-context* returns a better ranking in comparison to the $DIRT$ and $GM_{mixt,cos}$-*out-of-context* models.

6 Conclusion

We have addressed the task of computing meaning similarity in context using distributional methods. The specific representation we use follows (Lin and Pantel, 2001a): we extract *patterns* (paths in dependency trees which connect two nouns) and we use the co-occurrence with these nouns to build high-dimensional vectors. Using this data

$virus \xleftarrow{obj} shed \xrightarrow{prep} to \xrightarrow{pobj} cat$

$GM_{mixt,cos}$ *in-context*	$GM_{mixt,cos}$ *out-of-context*	$DIRT$	GOLD	
lose	*lose*	*drop*	pass	2
drop	*drop*	*lose*	spread	2
transmit	*relinquish*	*give*	transmit	2
spread	*reveal*	**transmit**		
pass	**pass**	**spread**		
relinquish	*throw*	*reveal*		
reveal	**spread**	*relinquish*		
throw	**transmit**	*throw*		
give	*give*	**pass**		

Table 7: Ranks returned for $virus \xleftarrow{obj} shed \xrightarrow{prep} to \xrightarrow{pobj} cat$

we develop a principled method to induce context-sensitive representations by modeling the *meaning* of a pattern as a latent variable in the input corpus. We apply this model to the task of Lexical Substitution and we show it allows the computation of context-sensitive similarities; it significantly outperforms the original method, while using the exact same input data.

In future work, we plan to use our model for generating paraphrases for patterns occurring in context, a scenario closer to real applications than *out-of-context* paraphrasing.

Finally, a formulation of our model in a typical bag-of-words semantic space for word similarity can be employed in a wider range of applications and will allow comparison with other methods for building context-sensitive vector representations.

7 Acknowledgments

This work was partially supported by DFG (IRTG 715).

References

Basili, Roberto, Diego De Cao, Paolo Marocco, and Marco Pennacchiotti. 2007. Learning selectional preferences for entailment or paraphrasing rules. In *In Proceedings of RANLP 2007*, Borovets, Bulgaria.

Blei, David M., Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022.

Boyd-Graber, Jordan, David M. Blei, and Xiaojin Zhu. 2007. A topic model for word sense disambiguation. In *Empirical Methods in Natural Language Processing*.

Brody, Samuel and Mirella Lapata. 2009. Bayesian word sense induction. In *EACL '09: Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 103–111, Morristown, NJ, USA. Association for Computational Linguistics.

Cai, Jun Fu, Wee Sun Lee, and Yee Whye Teh. 2007. Nus-ml:improving word sense disambiguation using topic features. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 249–252, Prague, Czech Republic, June. Association for Computational Linguistics.

Connor, Michael and Dan Roth. 2007. Context sensitive paraphrasing with a global unsupervised classifier. In *ECML '07: Proceedings of the 18th European conference on Machine Learning*, pages 104–115, Berlin, Heidelberg. Springer-Verlag.

Erk, Katrin and Sabastian Padó. 2008. A structured vector space model for word meaning in context. In *Proceedings of EMNLP 2008*, Waikiki, Honolulu, Hawaii.

Griffiths, T. L. and M. Steyvers. 2004. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101(Suppl. 1):5228–5235, April.

Kishida, Kazuaki. 2005. Property of average precision and its generalization: An examination of evaluation indicator for information retrieval experiments. *NII Technical Report*.

Landauer, Thomas K. and Susan T. Dumais. 1997. A solution to plato's problem: The latent semantic analysis theory of acquisition, induction and representation of knowledge. *Psychological Review*, 104(2):211–240.

Lin, Dekang and Patrick Pantel. 2001a. DIRT – Discovery of Inference Rules from Text. In *Proceedings of the ACM Conference on Knowledge Discovery and Data Mining (KDD-01)*, San Francisco, CA.

Lin, Dekang and Patrick Pantel. 2001b. Discovery of inference rules for question-answering. *Nat. Lang. Eng.*, 7(4):343–360.

McCarthy, D. and R. Navigli. 2007. SemEval-2007 Task 10: English Lexical Substitution Task. In *Proceedings of SemEval*, pages 48–53, Prague.

Mitchell, Jeff and Mirella Lapata. 2008. Vector-based models of semantic composition. In *Proceedings of ACL-08: HLT*, pages 236–244, Columbus, Ohio.

Pantel, Patrick, Rahul Bhagat, Bonaventura Coppola, Timothy Chklovski, and Eduard Hovy. 2007. ISP: Learning inferential selectional preferences. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics*, Rochester, New York.

Porteous, Ian, David Newman, Alexander Ihler, Arthur Asuncion, Padhraic Smyth, and Max Welling. 2008. Fast collapsed gibbs sampling for latent dirichlet allocation. In *KDD '08: Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 569–577, New York, NY, USA. ACM.

Schuetze, Hinrich. 1998. Automatic word sense discrimination. *Journal of Computational Linguistics*, 24:97–123.

Szpektor, Idan, Ido Dagan, Roy Bar-Haim, and Jacob Goldberger. 2008. Contextual preferences. In *Proceedings of ACL-08: HLT*, pages 683–691, Columbus, Ohio, June. Association for Computational Linguistics.

Thater, Stefan, Georgiana Dinu, and Manfred Pinkal. 2009. Ranking paraphrases in context. In *Proceedings of TextInfer ACL 2009*.

Toutanova, Kristina and Mark Johnson. 2008. A bayesian lda-based model for semi-supervised part-of-speech tagging. In Platt, J.C., D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 1521–1528. MIT Press, Cambridge, MA.

Wang, Yi, Hongjie Bai, Matt Stanton, Wen-Yen Chen, and Edward Y. Chang. 2009. Plda: Parallel latent dirichlet allocation for large-scale applications. In *Proc. of 5th International Conference on Algorithmic Aspects in Information and Management*.

Recognizing Medication related Entities in Hospital Discharge Summaries using Support Vector Machine

Son Doan and **Hua Xu**
Department of Biomedical Informatics
School of Medicine, Vanderbilt University
Son.Doan@Vanderbilt.edu, Hua.Xu@Vanderbilt.edu

Abstract

Due to the lack of annotated data sets, there are few studies on machine learning based approaches to extract named entities (NEs) in clinical text. The 2009 i2b2 NLP challenge is a task to extract six types of medication related NEs, including medication names, dosage, mode, frequency, duration, and reason from hospital discharge summaries. Several machine learning based systems have been developed and showed good performance in the challenge. Those systems often involve two steps: 1) recognition of medication related entities; and 2) determination of the relation between a medication name and its modifiers (e.g., dosage). A few machine learning algorithms including Conditional Random Field (CRF) and Maximum Entropy have been applied to the Named Entity Recognition (NER) task at the first step. In this study, we developed a Support Vector Machine (SVM) based method to recognize medication related entities. In addition, we systematically investigated various types of features for NER in clinical text. Evaluation on 268 manually annotated discharge summaries from i2b2 challenge showed that the SVM-based NER system achieved the best F-score of 90.05% (93.20% Precision, 87.12% Recall), when semantic features generated from a rule-based system were included.

1 Introduction

Named Entity Recognition (NER) is an important step in natural language processing (NLP). It has many applications in general language domain such as identifying person names, locations, and organizations. NER is crucial for biomedical literature mining as well (Hirschman, Morgan, & Yeh, 2002; Krauthammer & Nenadic, 2004) and many studies have focused on biomedical entities, such as gene/protein names. There are mainly two types of approaches to identify biomedical entities: rule-based and machine learning based approaches. While rule-based approaches use existing biomedical knowledge/resources, machine learning (ML) based approaches rely much on annotated training data. The advantage of rule-based approaches is that they usually can achieve stable performance across different data sets due to the verified resources, while machine learning approaches often report better results when the training data are good enough. In order to harness the advantages of both approaches, the combination of them, called the hybrid approach, has often been used as well. CRF and SVM are two common machine learning algorithms that have been widely used in biomedical NER (Takeuchi & Collier, 2003; Kazama, Makino, Ohta, & Tsujii, 2002; Yamamoto, Kudo, Konagaya, & Matsumoto, 2003; Torii, Hu, Wu, & Liu, 2009; Li, Savova, & Kipper-Schuler, 2008). Some studies reported better results using CRF (Li, Savova, & Kipper-Schuler, 2008), while others showed that the SVM was better (Tsochantaridis, Joachims, & Hofmann, 2005) in NER. Keerthi & Sundararajan (Keerthi & Sundararajan, 2007) conducted some experiments and demonstrated that CRF and SVM were quite close in performance, when identical feature functions were used.

2 Background

There has been large ongoing effort on processing clinical text in Electronic Medical Records (EMRs). Many clinical NLP systems

Coling 2008: Poster Volume, pages 259–266,
Beijing, August 2010

have been developed, including MedLEE (Friedman, Alderson, Austin, Cimino, & Johnson, 1994), SymTex (Haug et al., 1997), MetaMap (Aronson, 2001). Most of those systems recognize clinical named entities such as diseases, medications, and labs, using rule-based methods such as lexicon lookup, mainly because of two reasons: 1) there are very rich knowledge bases and vocabularies of clinical entities, such as the Unified Medical Language System (UMLS) (Lindberg, Humphreys, & McCray, 1993), which includes over 100 controlled biomedical vocabularies, such as RxNorm, SNOMED, and ICD-9-CM; 2) very few annotated data sets of clinical text are available for machine learning based approaches.

Medication is one of the most important types of information in clinical text. Several studies have worked on extracting drug names from clinical notes. Evans et al. (Evans, Brownlow, Hersh, & Campbell, 1996) showed that drug and dosage phrases in discharge summaries could be identified by the CLARIT system with an accuracy of 80%. Chhieng et al. (Chhieng, Day, Gordon, & Hicks, 2007) reported a precision of 83% when using a string matching method to identify drug names in clinical records. Levin et al. (Levin, Krol, Doshi, & Reich, 2007) developed an effective rule-based system to extract drug names from anesthesia records and map to RxNorm concepts with 92.2% sensitivity and 95.7% specificity. Sirohi and Peissig (Sirohi & Peissig, 2005) studied the effect of lexicon sources on drug extraction. Recently, Xu et al. (Xu et al., 2010) developed a rule-based system for medication information extraction, called MedEx, and reported F-scores over 90% on extracting drug names, dose, route, and frequency from discharge summaries.

Starting 2007, Informatics for Integrating Biology and the Bedside (i2b2), an NIH-funded National Center for Biomedical Computing (NCBC) based at Partners Healthcare System in Boston, organized a series of shared tasks of NLP in clinical text. The 2009 i2b2 NLP challenge was to extract medication names, as well as their corresponding signature information including dosage, mode, frequency, duration, and reason from de-identified hospital discharge summaries (Uzüner, Solti, & Cadag, 2009). At the beginning of the challenge, a training set of 696 notes were provided by the organizers. Among them, 17 notes were annotated by the i2b2 organizers, based on an annotation guideline (see Table 1 for examples of medication information in the guideline), and the rest were un-annotated notes. Participating teams would develop their systems based on the training set, and they were

Class	#	Example	Description
Medication	12773	“Lasix”, “Caltrate plus D”, “fluocinonide 0.5% cream”, “TYLENOL (ACETAMINOPHEN)”	Prescription substances, biological substances, over-the-counter drugs, excluding diet, allergy, lab/test, alcohol.
Dosage	4791	“1 TAB”, “One tablet”, “0.4 mg” “0.5 m.g.”, “100 MG”, “100 mg x 2 tablets”	The amount of a single medication used in each administration.
Mode	3552	“Orally”, “Intravenous”, “Topical”, “Sublingual”	Describes the method for administering the medication.
Frequency	4342	“Prn”, “As needed”, “Three times a day as needed”, “As needed three times a day”, “x3 before meal”, “x3 a day after meal as needed”	Terms, phrases, or abbreviations that describe how often each dose of the medication should be taken.
Duration	597	“x10 days”, “10-day course”, “For ten days”, “For a month”, “During spring break”, “Until the symptom disappears”, “As long as needed”	Expressions that indicate for how long the medication is to be administered.
Reason	1534	“Dizziness”, “Dizzy”, “Fever”, “Diabetes”, “frequent PVCs”, “rare angina”	The medical reason for which the medication is stated to be given.

Table 1.Number of classes and descriptions with examples in i2b2 2009 dataset.

allowed to annotate additional notes in the training set. The test data set included 547clinical notes, from which 251 notes were randomly picked by the organizers. Those 251 notes were then annotated by participating teams, as well as the organizers, and they served as the gold standard for evaluating the performance of systems submitted by participating teams. An example of original text and annotated text were shown in Figure 1.

The results of systems submitted by the participating teams were presented at the i2b2 workshop and short papers describing each system were available at i2b2 web site with protected passwords. Among top 10 systems which achieved the best performance, there were 6 rule-based, 2 machine learning based, and 2 hybrid systems. The best system, which used a machine learning based approach, reached the highest F-score of 85.7% (Patrick & Li, 2009). The second best system, which was a rule-based system using the existing MedEx tool, reported an F-score of 82.1% (Doan, Bastarache L., Klimkowski S., Denny J.C., & Xu, 2009). The difference between those two systems was statistically significant. However, this finding was not very surprising, as the machine learning based system utilized additional 147 annotated notes by the participating team, while the rule-based system mainly used 17 annotated training data to customize the system.

Interestingly, two machine learning systems in the top ten systems achieved very different performance, one (Patrick et al., 2009) achieved an F-score of 85.7%, ranked the first; while another (Li et al., 2009) achieved an F-score of 76.4%, ranked the 10th on the final evaluation. Both systems used CRF for NER, on the equivalent number of training data (145 and 147 notes respectively). The large difference in F-score of those two systems could be due to: the quality of training set, and feature sets using for classification. More recently, i2b2 organizers also reported a Maximum Entropy (ME) based approach for the 2009 challenge (Halgrim, Xia, Solti, Cadag, & Uzuner, 2010). Using the same annotated data set as in (Patrick et al., 2009), they reported an F-score of 84.1%, when combined features such as unigram, word bigrams/trigrams, and label of previous words were used. These results indicated the importance of feature sets used in machine learning algorithms in this task.

For supervised machine learning based systems in the i2b2 challenge, the task was usually divided into two steps: 1) NER of six medication related findings; and 2) determination of the relation between detected medication names and other entities. It is obvious that NER is the first crucial step and it affects the performance of the whole system. However, short papers presented at the i2b2 workshop did not show much detailed evaluation on NER components in machine learning based systems. The variation in performance of different machine learning based systems also motivated us to further investigate the effect of different types of features on recogniz-

# Line	Original text
70	DISCHARGE MEDICATION:
..	…
74	Additionally, Percocet 1-2 tablets p.o. q 4 prn, Colace 100 mg p.o.
75	b.i.d. , insulin NPH 10 units subcu b.i.d. , sliding scale insulin...

Annotated text:
m="colace" 74:10 74:10||do="100 mg" 74:11 74:12||mo="p.o." 74:13 74:13||f="b.i.d." 75:0 75:0||du="nm" ||r="nm"||ln="list"
m="percocet" 74:2 74:2||do="1-2 tablets" 74:3 74:4||mo="p.o." 74:5 74:5||f="q 4 prn" 74:6 74:8||du="nm"||r="nm"||ln="list"

Figure. 1. An example of the i2b2 data, 'm' is for MED NAME, 'do' is for DOSE, 'mo' is for MODE, 'f' is for FREQ, 'du' is for DURATION, 'r' is for REASON, 'ln' is for "list/narrative."

ing medication related entities.

In this study, we developed an SVM-based NER system for recognizing medication related entities, which is a sub-task of the i2b2 challenge. We systematically investigated the effects of typical local contextual features that have been reported in many biomedical NER studies. Our studies provided some valuable insights to NER tasks of medical entities in clinical text.

3 Methods

A total of 268 annotated discharge summaries (17 from training set and 251 from test set) from i2b2 challenge were used in this study. This annotated corpus contains 9,689 sentences, 326,474 words, and 27,589 entities. Annotated notes were converted into a BIO format and different types of feature sets were used in an SVM classifier for NER. Performance of the NER system was evaluated using precision, recall, and F-score, based on 10-fold cross validation.

3.1 Preprocessing

The annotated corpus was converted into a BIO format (see an example in Figure 2). Specifically, it assigned each word into a class as follows: **B** means beginning of an entity, **I** means inside an entity, and **O** means outside of an entity. As we have six types of entities, we have six different B classes and six different I classes. For example, for medication names, we define the B class as "B-m", and the I class as "I-m". Therefore, we had total 13 possible classes to each word (including O class).

DISCHARGE			MEDICATION:	
O			**O**	
Additionally,		Percocet	1-2	Tablets
O		**B-m**	**B-do**	**I-do**
p.o.	Q	4	prn,	
B-mo	**B-f**	**I-f**	**I-f**	

Figure 2. An example of the BIO representation of annotated clinical text (Where m as medication, do as dose, mo as mode, and f as frequency).

After preprocessing, the NER problem now can be considered as a classification problem, which is to assigns one of the 13 class labels to each word.

3.2 SVM

Support Vector Machine (SVM) is a machine learning method that is widely used in many NLP tasks such as chunking, POS, and NER. Essentially, it constructs a binary classifier using labeled training samples. Given a set of training samples, the SVM training phrase tries to find the optimal hyperplane, which maximizes the distance of training sample nearest to it (called support vectors). SVM takes an input as a vector and maps it into a feature space using a kernel function.

In this paper we used TinySVM[1] along with Yamcha[2] developed at NAIST (Kudo & Matsumoto, 2000; Kudo & Matsumoto, 2001). We used a polynomial kernel function with the degree of kernel as 2, context window as +/-2, and the strategy for multiple classification as pairwise (one-against-one). Pairwise strategy means it will build K(K-1)/2 binary classifiers in which K is the number of classes (in this case K=13). Each binary classifier will determine whether the sample should be classified as one of the two classes. Each binary classifier has one vote and the final output is the class with the maximum votes. These parameters were used in many biomedical NER tasks such as (Takeuchi & Collier, 2003; Kazama et al., 2002; Yamamoto et al., 2003).

3.3 Features sets

In this study, we investigated different types of features for the SVM-based NER system for medication related entities, including 1) words; 2) Part-of-Speech (POS) tags; 3) morphological clues; 4) orthographies of words; 5) previous history features; 6) semantic tags determined by MedEx, a rule based medication extraction system. Details of those features are described below:

- Words features: Words only. We referred it as a baseline method in this study.
- POS features: Part-of-Speech tags of words. To obtain POS information, we used a POS tagger in the NLTK package[3].

[1] Available at http://chasen.org/~taku/software/TinySVM/

[2] Available at http://chasen.org/~taku/software/YamCha/

[3] www.nltk.org

- Morphologic features: suffix/prefix of up to 3 characters within a word.
- Orthographic features: information about if a word contains capital letters, digits, special characters etc. We used orthographic features described in (Collier, Nobata, & Tsujii, 2000) and modified some as for medication information such as "digit and percent". We had totally 21 labels for orthographic features.
- Previous history features: Class assignments of preceding words, by the NER system itself.
- Semantic tag features: semantic categories of words. Typical NER systems use dictionary lookup methods to determine semantic categories of a word (e.g., gene names in a dictionary). In this study, we used MedEx, the best rule-based medication extraction system in the i2b2 challenge, to assign medication specific categories into words.

MedEx was originally developed at Vanderbilt University, for extracting medication information from clinical text (Xu et al., 2010). MedEx labels medication related entities with a pre-defined semantic categories, which has overlap with the six entities defined in the i2b2 challenge, but not exactly same. For example, MedEx breaks the phrase *"fluocinonide 0.5% cream"* into *drug nam*e: *"fluocinonide", strength*: *"0.5%",* and *form*: *"cream";* while i2b2 labels the whole phrase as a medication name. There are a total of 11 pre-defined semantic categories which are listed in (Xu et al., 2010c). When the Vanderbilt team applied MedEx to the i2b2 challenge, they customized and extended MedEx to label medication related entities as required by i2b2. Those customizations included:

- Customized Rules to combine entities recognized by MedEx into i2b2 entities, such as combine drug name: *"fluocinonide",* strength: *"0.5%", and* form: *"cream"* into one medication nam*e "fluocinonide 0.5% cream".*
- A new Section Tagger to filter some drug names in sections such as "allergy" and "labs".
- A new Spell Checker to check whether a word can be a misspelled drug names.

In a summary, the MedEx system will produce two sets of semantic tags: 1) initial tags that are identified by the original MedEx system; 2) final tags that are identified by the customized MedEx system for the i2b2 challenge. The initial tagger will be equivalent to some simple dictionary look up methods used in many NER systems. The final tagger is a more advanced method that integrates other level of information such as sections and spellings. The outputs of initial tag include 11 pre-defined semantic tags in MedEx, and outputs of final tags consist of 6 types of NEs as in the i2b2 requirements. Therefore, it is interesting to us to study effects of both types of tags from MedEx in this study. These semantic tags were also converted into the BIO format when they were used as features.

4 Results and Discussions

In this study, we measured Precision, Recall, and

Features	**Pre**	**Rec**	**F-score**
Words (Baseline)	87.09	77.05	81.76
Words + History	90.34	78.17	83.81
Words + History + Morphology	91.72	81.08	86.06
Words + History + Morphology + POS	91.81	81.06	86.10
Words + History + Morphology + POS + Orthographies	91.78	81.29	86.22
Words + Semantic Tags (Original MedEx)	90.15	83.17	86.51
Words + Semantic Tags (Customized MedEx)	92.38	86.73	89.47
Words + History + Morphology + POS + Orthographies + Semantic Tags (Original MedEx)	91.43	84.2	87.66
Words + History + Morphology + POS + Orthographies + Semantic Tags (Customized MedEx)	**93.2**	**87.12**	**90.05**

Table 2. Performance of the SVM-based NER system for different feature combinations.

F-score using the CoNLL evaluation script[4]. Precision is the ratio between the number of correctly identified NE chunks by the system and the total number of NE chunks found by the system; Recall is the ratio between the number of correctly identified NE chunks by the system and the total number of NE chunks in the gold standard. Experiments were run in a Linux machine with 16GB RAM and 8 cores of Intel Xeon 2.0GHz processor. The performance of different types of feature sets was evaluated using 10-fold cross-validation.

Table 2 shows the precision, recall, and F-score of the SVM-based NER system for all six types of entities, when different combinations of feature sets were used. Among them, the best F-score of 90.05% was achieved, when all feature sets were used. A number of interesting findings can be concluded from those results. First, the contribution of different types of features to the system's performance varies. For example, the "previous history feature" and the "morphology feature" improved the performance substantially (F-score from 81.76% to 83.83%, and from 83.81% to 86.06% respectively). These findings were consistent with previous reported results on protein/gene NER (Kazama et al., 2002; Takeuchi and Collier, 2003; Yamamoto et al., 2003). However, "POS" and "orthographic" features contributed very little, not as much as in protein/gene names recognition tasks. This could be related to the differences between gene/protein phrases and medication phrases – more orthographic clues are observed in gene/protein names. Second, the "semantic tags" features alone, even just using the original tagger in MedEx, improved the performance dramatically (from 81.76% to 86.51% or 89.47%). This indicates that the knowledge bases in the biomedical domain are crucial to biomedical NER. Third, the customized final semantic tagger in MedEx had much better performance than the original tagger, which indicated that advanced semantic tagging methods that integrate other levels of linguistic information (e.g., sections) were more useful than simple dictionary lookup methods.

Table 3 shows the precision, recall, and F-score for each type of entity, from the MedEx alone, and the baseline and the best runs of the SVM-based NER system. As we can see, the best SVM-based NER system that combines all types of features (including inputs from MedEx) was much better than the MedEx system alone (90.05% vs. 85.86%). This suggested that the combination of rule-based systems with machine learning approaches could yield the most optimized performance in biomedical NER tasks.

Among six types of medication entities, we noticed that four types of entities (medication names, dosage, mode, and frequency) got very high F-scores (over 92%); while two others (duration and reason) had low F-scores (up to 50%). This finding was consistent with results from i2b2 challenge. Duration and reason are more difficult to identify because they do not have well-formed patterns and few knowledge bases exist for duration and reasons.

This study only focused on the first step of the i2b2 medication extraction challenge – NER. Our next plan is to work on the second step of determining relations between medication names and other entities, thus allowing us to compare our results with those reported in the i2b2 challenge. In addition, we will also evaluate and compare the performance of other ML algorithms such as CRF and ME on the same NER task.

Entity	**MedEx only**			**SVM (Baseline)**			**SVM (Best)**		
	Pre	Rec	F-score	Pre	Rec	F-score	Pre	Rec	F-score
ALL	**87.85**	**83.97**	**85.86**	87.09	77.05	81.76	**93.2**	**87.12**	**90.05**
Medication	87.25	90.21	88.71	88.38	75.03	81.16	93.3	91.35	92.31
Dosage	92.79	83.94	88.14	89.43	83.65	86.41	94.38	90.99	92.65
Mode	95.86	90.06	92.87	96.18	93.30	94.70	97.12	93.8	95.41
Frequency	92.67	89.00	90.80	90.33	87.60	88.94	95.88	93.04	94.43
Duration	42.65	40.15	41.36	24.16	19.62	21.45	65.18	40.16	49.57
Reason	54.23	36.72	43.79	48.40	25.51	33.30	69.21	37.39	48.4

Table 3. Comparison between a rule based system and the SVM based system.

[4] Available at http://www.cnts.ua.ac.be/conll2002/ner/bin/conlleval.txt

5 Conclusions

In this study, we developed an SVM-based NER system for medication related entities. We systematically investigated different types of features and our results showed that by combining semantic features from a rule-based system, the ML-based NER system could achieve the best F-score of 90.05% in recognizing medication related entities, using the i2b2 annotated data set. The experiments also showed that optimized usage of external knowledge bases were crucial to high performance ML based NER systems for medical entities such as drug names.

Acknowledgements

Authors would like to thank i2b2 organizers for organizing the 2009 i2b2 challenge and providing dataset for research studies. This study was in part supported by NCI grant R01CA141307-01.

References:

Aronson, A. R. (2001). Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap program. *Proc AMIA Symp.,* 17-21.

Chhieng, D., Day, T., Gordon, G., & Hicks, J. (2007). Use of natural language programming to extract medication from unstructured electronic medical records. *AMIA.Annu.Symp.Proc.,* 908.

Collier, N., Nobata, C., & Tsujii, J. (2000). Extracting the names of genes and gene products with a hidden Markov model. *Proc.of the 18th Conf.on Computational linguistics., 1,* 201-207.

Doan, S., Bastarache L., Klimkowski S., Denny J.C., & Xu, H. (2009). Vanderbilt's System for Medication Extraction. *Proc of 2009 i2b2 workshop.*.

Evans, D. A., Brownlow, N. D., Hersh, W. R., & Campbell, E. M. (1996). Automating concept identification in the electronic medical record: an experiment in extracting dosage information. *Proc.AMIA.Annu.Fall.Symp.,* 388-392.

Friedman, C., Alderson, P. O., Austin, J. H., Cimino, J. J., & Johnson, S. B. (1994). A general natural-language text processor for clinical radiology. *J.Am.Med.Inform.Assoc., 1,* 161-174.

Halgrim, S., Xia, F., Solti, I., Cadag, E., & Uzuner, O. (2010). Statistical Extraction of Medication Information from Clinical Records. *AMIA Summit on Translational Bioinformatics,* 10-12.

Haug, P. J., Christensen, L., Gundersen, M., Clemons, B., Koehler, S., & Bauer, K. (1997). A natural language parsing system for encoding admitting diagnoses. *Proc AMIA Annu.Fall.Symp.,* 814-818.

Hirschman, L., Morgan, A. A., & Yeh, A. S. (2002). Rutabaga by any other name: extracting biological names. *J.Biomed.Inform., 35,* 247-259.

Kazama, J., Makino, T., Ohta, Y., & Tsujii, T. (2002). Tuning Support Vector Machines for Biomedical Named Entity Recognition. *Proceedings of the ACL-02 Workshop on Natural Language Processing in the Biomedical Domain,* 1-8.

Keerthi, S. & Sundararajan, S. (2007). CRF versus SVM-struct for sequence labeling. *Yahoo Research Technical Report*.

Krauthammer, M. & Nenadic, G. (2004). Term identification in the biomedical literature. *J.Biomed.Inform., 37,* 512-526.

Kudo, T. & Matsumoto, Y. (2000). Use of Support Vector Learning for Chunk Identification. *Proc.of CoNLL-2000*.

Kudo, T. & Matsumoto, Y. (2001). Chunking with Support Vector Machines. *Proc.of NAACL 2001*.

Levin, M. A., Krol, M., Doshi, A. M., & Reich, D. L. (2007). Extraction and mapping of drug names from free text to a standardized nomenclature. *AMIA.Annu.Symp.Proc.,* 438-442.

Li, D., Savova, G., & Kipper-Schuler, K. (2008). Conditional random fields and support vector machines for disorder named entity recognition in clinical texts. *Proceedings of the workshop on current trends in biomedical natural language processing (BioNLP'08),* 94-95.

Li, Z., Cao, Y., Antieau, L., Agarwal, S., Zhang, Z., & Yu, H. (2009). A Hybrid Approach to Extracting Medication Information from Medical Discharge Summaries. *Proc of 2009 i2b2 workshop.*.

Lindberg, D. A., Humphreys, B. L., & McCray, A. T. (1993). The Unified Medical Language System. *Methods Inf.Med., 32,* 281-291.

Patrick, J. & Li, M. (2009). A Cascade Approach to Extract Medication Event (i2b2 challenge 2009). *Proc of 2009 i2b2 workshop.*.

Sirohi, E. & Peissig, P. (2005). Study of effect of drug lexicons on medication extraction from electronic medical records. *Pac.Symp.Biocomput.,* 308-318.

Takeuchi, K. & Collier, N. (2003). Bio-medical entity extraction using Support Vector Machines. *Proceedings of the ACL 2003 workshop on Natural language processing in biomedicine,* 57-64.

Torii, M., Hu, Z., Wu, C. H., & Liu, H. (2009). BioTagger-GM: a gene/protein name recognition system. *J.Am.Med.Inform.Assoc., 16,* 247-255.

Tsochantaridis, I., Joachims, T., & Hofmann, T. (2005). Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research, 6,* 1453-1484.

Uzüner, O., Solti, I., & Cadag, E. (2009). The third 2009 i2b2 challenge.
In https://www.i2b2.org/NLP/Medication/.

Xu, H., Stenner, S. P., Doan, S., Johnson, K. B., Waitman, L. R., & Denny, J. C. (2010). MedEx: a medication information extraction system for clinical narratives. *J.Am.Med.Inform.Assoc., 17,* 19-24.

Yamamoto, K., Kudo, T., Konagaya, A., & Matsumoto, Y. (2003). Protein name tagging for biomedical annotation in text. *Proceedings of ACL 2003 Workshop on Natural Language Processing inBiomedicine,2003, 13,* 65-72.

Exploring the Data-Driven Prediction of Prepositions in English

Anas Elghafari **Detmar Meurers** **Holger Wunsch**
Seminar für Sprachwissenschaft
Universität Tübingen
{aelgafar,dm,wunsch}@sfs.uni-tuebingen.de

Abstract

Prepositions in English are a well-known challenge for language learners, and the computational analysis of preposition usage has attracted significant attention. Such research generally starts out by developing models of preposition usage for native English based on a range of features, from shallow surface evidence to deep linguistically-informed properties.

While we agree that ultimately a combination of shallow and deep features is needed to balance the preciseness of exemplars with the usefulness of generalizations to avoid data sparsity, in this paper we explore the limits of a purely surface-based prediction of prepositions.

Using a web-as-corpus approach, we investigate the classification based solely on the relative number of occurrences for target n-grams varying in preposition usage. We show that such a surface-based approach is competitive with the published state-of-the-art results relying on complex feature sets.

Where enough data is available, in a surprising number of cases it thus is possible to obtain sufficient information from the relatively narrow window of context provided by n-grams which are small enough to frequently occur but large enough to contain enough predictive information about preposition usage.

1 Introduction

The correct use of prepositions is a well-known difficulty for learners of English, and correspondingly the computational analysis of preposition usage has attracted significant attention in recent years (De Felice and Pulman, 2007; De Felice, 2008; Lee and Knutsson, 2008; Gamon et al., 2008; Chodorow et al., 2007; Tetreault and Chodorow, 2008a, 2008b).

As a point of reference for the detection of preposition errors in learner language, most of the research starts out by developing a model of preposition usage for native English. For this purpose, virtually all previous approaches employ a machine learning setup combining a range of features, from surface-based evidence to deep linguistically-informed properties. The overall task is approached as a classification problem where the classes are the prepositions and the instances to be classified are the contexts, i.e., the sentences with the prepositions omitted.

A focus of the previous literature is on the question which linguistic and lexical features are the best predictors for preposition usage. Linguistic features used include the POS tags of the surrounding words, PP attachment sites, WordNet classes of PP object and modified item. Lexical features used include the object of the PP and the lexical item modified by the PP. Those syntactic, semantic and lexical features are then extracted from the training instances and used by the machine learning tool to predict the missing preposition in a test instance.

While we agree that ultimately a combination of shallow and linguistically informed features is needed to balance the preciseness of exemplars

Coling 2008: Poster Volume, pages 267–275,
Beijing, August 2010

with the usefulness of generalizations to avoid data sparsity problems, in this paper we want to explore the limits of a purely surface-based prediction of prepositions. Essentially, our question is how much predictive information can be found in the immediate distributional context of the preposition. Is it possible to obtain n-gram contexts for prepositions which are small enough to occur frequently enough in the available training data but large enough to contain enough predictive information about preposition usage?

This perspective is related to that underlying the variation-n-gram approach for detecting errors in the linguistic annotation of corpora (Dickinson and Meurers, 2003; Dickinson and Meurers, 2005; Boyd et al., 2008). Under that approach, errors in the annotation of linguistic properties (lexical, constituency, or dependency information) are detected by identifying units which recur in the corpus with sufficient identical context so as to make variation in their annotation unlikely to be correct. In a sense, the recurring n-gram contexts are used as exemplar references for the local domains in which the complex linguistic properties are established. The question now is to what extent basic[1] n-gram contexts can also be successfully used to capture the linguistic properties and relations determining preposition usage, exploring the trade-off expressed in the question ending the previous paragraph.

To address this question, in this paper we make use of a web-as-corpus approach in the spirit of Lapata and Keller (2005). We employ the Yahoo search engine to investigate a preposition classification setup based on the relative number of web counts obtained for target n-grams varying in the preposition used. We start the discussion with a brief review of key previous approaches and the results they obtain for the preposition classification task in native English text. In section 2, we then describe the experimental setup we used for our exploration and discuss our results in section 3.

1.1 Previous work and results

The previous work on the preposition prediction task varied in i) the features selected, ii) the number of prepositions tackled, and iii) the training and testing corpora used.

De Felice (2008) presents a system that (among other things) is used to predict the correct preposition for a given context. The system tackles the nine most frequent prepositions in English: *of, to, in, for, on, with, at, by, from*. The approach uses a wide variety of syntactic and semantic features: the lexical item modified by the PP, the lexical item that occurs as the object of the preposition, the POS tags of three words to the left and three words to the right of the preposition, the grammatical relation that the preposition is in with its object, the grammatical relation the preposition is in with the word modified by the PP, and the WordNet classes of the preposition's object and the lexical item modified by the PP. De Felice (2008) also used a named entity recognizer to extract generalizations about which classes of named entities can occur with which prepositions. Further, the verbs' subcategorization frames were taken as features. For features that used lexical sources (WordNet classes, verbs subcategorization frames), only partial coverage of the training and testing instances is available.

The overall accuracy reported by De Felice (2008) for this approach is 70.06%, testing on section J of the *British National Corpus (BNC)* after training on the other sections. As the most extensive discussion of the issue, using an explicit set of prepositions and a precisely specified and publicly accessible test corpus, De Felice (2008) is well-suited as a reference approach. Correspondingly, our study in this paper is based on the same set of prepositions and the same test corpus.

Gamon et al. (2008) introduce a system for the detection of a variety of learner errors in non-native English text, including preposition errors. For the preposition task, the authors combine the outputs of a classifier and a language model. The language model is a 5-gram model trained on the English Gigaword corpus. The classifier is trained

[1] While Dickinson and Meurers (2005) also employ discontinuous n-grams, we here focus only on contiguous n-gram contexts. Using discontinuous n-gram contexts for preposition prediction could be interesting to explore in the future, once, as a prerequisite for the effective generation of discontinuous n-grams, heuristics have been identified for when which kind of discontinuities should be allowed to arise for preposition classification contexts.

on Encarta encyclopedia and Reuters news text. It operates in two stages: The presence/absence classifier predicts first whether a preposition needs to be inserted at a given location. Then, the choice classifier determines which preposition is to be inserted. The features that are extracted for each possible insertion site come from a six-token window around the possible insertion site. Those features are the relative positions, POS tags, and surface forms of the tokens in that window. The choice classifier predicts one of 13 prepositions: *in, for, of, on, to, with, at, by, as, from, since, about, than,* and *other*. The accuracy of the choice classifier, the part of the system to which the work at hand is most similar, is 62.32% when tested on text from Encarta and Reuters news.

Tetreault and Chodorow (2008a) present a system for detecting preposition errors in learner text. Their approach extracts a total of 25 features from the local contexts: the adjacent words, the heads of the nearby phrases, and the POS tags of all those. They combine word-based features with POS tag features to better handle cases where a word from the test instance has not been seen in training. For each test instance, the system predicts one of 34 prepositions. In training and testing performed on the Encarta encyclopedia, Reuters news text and additional training material an accuracy figure of 79% is achieved.

Bergsma et al. (2009) extract contextual features from the Google 5-gram corpus to train an SVM-based classifier for predicting prepositions. They evaluate on 10 000 sentences taken from the New York Times section of the Gigaword corpus, and achieve an accuracy of 75.4%.

Following De Felice (2008, p. 66), we summarize the main results of the mentioned approaches to preposition prediction for native text in Figure 1.[2] Since the test sets and the prepositions targeted differ between the approaches, such a comparison must be interpreted with caution. In terms of the big picture, it is useful to situate the results with respect to the majority baseline reported by De Felice (2008). It is obtained by always choosing *of* as the most common preposition in section J of the BNC. De Felice also reports another interesting figure included in Figure 1, namely the accuracy of the human agreement with the original text, averaged over two English native-speakers.

Approach	Accuracy
Gamon et al. (2008)	62.32%
Tetreault and Chodorow (2008a)	79.00%
Bergsma et al. (2009)	75.50%
De Felice (2008) system	70.06%
Majority baseline *(of)*	26.94%
Human agreement	88.60%

Figure 1: Preposition prediction results

2 Experiments

2.1 Data

As our test corpus, we use section J of the BNC, the same corpus used by De Felice (2008). Based on the tokenization as given in the corpus, we join the tokens with a single space, which also means that punctuation characters end up as separate, white-space separated tokens. We select all sentences that contain one or more prepositions, using the POS annotation in the corpus to identify the prepositions. The BNC is POS-annotated with the CLAWS-5 tagset, which distinguishes the two tags `PRF` for *of* and `PRP` for all other prepositions.[3] We mark every occurrence of these preposition tags in the corpus, yielding one prediction task for each marked preposition. For example, the sentence (1) yields four prediction tasks, one for each of the prepositions *for, of, from,* and *in* in the sentence.

(1) But **for** the young, it is rather a question **of** the scales falling **from** their eyes, and having nothing to believe **in** any more.

In each task, one preposition is masked using the special marker `-*-MASKED-*-`. Figure 2 shows the four marked-up prediction tasks resulting for example (1).

Following De Felice (2008), we focus our experiments on the top nine prepositions in the BNC: *of, to, in, for, on, with, at, by, from*. For

[2] The Gamon et al. (2008) result differs from the one reported in De Felice (2008); we rely on the original paper.

[3] `http://www.natcorp.ox.ac.uk/docs/URG/posguide.html#guidelines`

```
But -*-MASKED-*-for the young , it is
rather a question of the scales falling
from their eyes , and having nothing to
believe in any more .

But for the young , it is rather a
question -*-MASKED-*-of the scales
falling from their eyes , and having
nothing to believe in any more .

But for the young , it is rather
a question of the scales falling
-*-MASKED-*-from their eyes , and having
nothing to believe in any more .

But for the young , it is rather a
question of the scales falling from
their eyes , and having nothing to
believe -*-MASKED-*-in any more .
```

Figure 2: Four prediction tasks for example (1)

each occurrence of these nine prepositions in section J of the BNC, we extract one prediction task, yielding a test set of 522 313 instances.

Evaluating on this full test set would involve a prohibitively large number of queries to the Yahoo search engine. We therefore extract a randomly drawn subset of 10 000 prediction tasks. From this subset, we remove all prediction tasks which are longer than 4000 characters in length, as Yahoo only supports queries up to that length. Finally, in a web-as-corpus setup, the indexing of the web pages performed by the search engine essentially corresponds to the training step in a typical machine learning setup. In order to avoid testing on the training data, we thus need to ensure that the test cases are based on text not indexed by the search engine. To exclude any such cases, we query the search engine with each complete sentence that a prediction task is based on and remove any prediction task for which the search engine returns hits for the complete sentence. The final test set consists of 8060 prediction tasks.[4]

2.2 Experimental Setup

Recall that the general issue we are interested in is whether one can obtain sufficient information from the relatively narrow distributional window of context provided by n-grams which are small enough to occur frequently enough in the training data but large enough to contain enough predictive information about preposition usage for the instances to be classified. By using a web-as-corpus approach we essentially try to maximize the training data size. For the n-gram size, we explore the use of a maximum order of 7, containing the preposition in the middle and three words of context on either side.

For each prediction task, we successively insert one of the nine most frequent prepositions into the marked preposition slot of the 8060 n-grams obtained from the test set. Thus, for each prediction task, we get a *cohort* consisting of nine different individual queries, one query for each potential preposition. For example, the second prediction task of Figure 2 yields the cohort of nine queries in Figure 3 below, where the candidate prepositions replace the location marked by `-*-MASKED-*-of`. The correct preposition *of* is stripped off and kept for later use in the evaluation step.

1. `rather a question` **`of`** `the scales falling`
2. `rather a question` **`to`** `the scales falling`
3. `rather a question` **`in`** `the scales falling`

   ⋮

9. `rather a question` **`from`** `the scales falling`

Figure 3: Cohort of nine queries resulting for the second prediction task of Figure 2

In cases where a preposition is closer than four words to the beginning or the end of the corresponding sentence, a lower-order n-gram results. For example, in the first prediction task in Figure 2, the preposition occurs already as the second word in the sentence, thus not leaving enough context to the left of the preposition for a symmetric 7-gram. Here, the truncated asymmetric 5-gram "`But` **`<prep>`** `the young ,`" including only one word of context on the left would get used.

We issue each query in a cohort to the Yahoo search engine, and determine the number of hits returned for that query. To that end, we use Yahoo's BOSS service, which offers a

[4] For a copy of the test set, just send us an email.

JSON interface supporting straightforward automated queries. As part of its response to a query, the BOSS service includes the `deephits` field, which gives an "approximate count that reflects duplicate documents and all documents from a host".[5] In other words, this number is an approximate measure of how many web pages there are that contain the search pattern.

With the counts for all nine queries in a cohort retrieved from Yahoo, we select the preposition of the query with the highest count. For the cases in which none of the counts in a 7-gram cohort is greater than zero, we use one of two strategies:

In the **baseline** condition, for all n-gram cohorts with zero counts (5160 out of the 8060 cases) we predict the most frequent preposition *of*, i.e., the majority baseline. This results in an overall accuracy of 50%.

In the **full back-off** condition, we explore the trade-off between the predictive power of the n-gram as context and the likelihood of having seen this n-gram in the training material, i.e., finding it on the web. In this paper we never abstract or generalize away from the surface string (e.g., by mapping all proper names to an abstract name tag; but see the outlook discussion at the end of the paper), so the only option for increasing the number of occurrences of an n-gram is to approximate it with multiple shorter n-grams.

Concretely, if no hits could be found for any of the queries in a cohort, we back off to the sum of the hits for the two overlapping 6-grams constructed in the way illustrated in Figure 4.

```
[rather a question of the scales falling]
                    ⇓
[rather a question of the scales]
       [a question of the scales falling]
```

Figure 4: Two overlapping 6-grams approximate a 7-gram for back-off.

If still no hits can be obtained after backing off to 6-grams for any of the queries in a cohort, the system backs off further to overlapping 5-grams, and so on, down to trigrams.[6]

[5]Cited from `http://developer.yahoo.com/search/boss/boss_guide/ch02s02.html`

[6]When backing off, the left-most and the right-most trigram do not include the target preposition of the original 7-gram. However, this only affects 13 cases, cf. Figure 6.

3 Results

Figure 5 shows the results of the **full back-off** approach. Compared to the baseline condition, accuracy goes up significantly to 76.5%. Thus, the back-off strategy is effective in increasing the amount of available data using lower-order n-grams. This increase of data is also reflected in the number of cases with zero counts for a cohort, which goes down to none.

	Full back-off
Correct	6166
Incorrect	1894
Total	8060
Accuracy	76.5%

Figure 5: Overall results of our experiments.

Figure 6 provides a detailed analysis of the back-off experiment. It lists back-off sequences separately for each maximum n-gram order. The prediction tasks for which a full 7-gram can be extracted are displayed in the third column, with back-off orders of 6 down to 3. Prediction tasks for which only asymmetric 6-grams can be extracted follow in column 4, and so on until 4-grams. There are no predictions tasks that are shorter than four words. Therefore, n-grams with a length of less than 4 do not occur.

The "sum" column shows the combined results of the full 7-gram prediction tasks and the prediction tasks involving truncated, asymmetric n-grams of lower orders.

There are 6999 prediction tasks for which full 7-grams can be extracted. The remaining 1061 of the 8060 prediction tasks are the cases where the system extracts only asymmetric lower-order n-grams, for the reasons explained in section 2.2.

For 2195 of the 6999 7-gram prediction tasks, we find full 7-gram contexts on the web, of which 1931 lead to a correct prediction, and 264 to an incorrect one, leaving 4804 prediction tasks still to be solved through the back-off approach. Thus, full 7-gram contexts lead to high-quality predictions at 88% precision, but they are rare and with a recall of 28,7% cover only a fraction of all cases.

	sum	**7-grams** (3 + prep + 3)	**6-grams** (truncated 7-gram)	**5-grams** (truncated 7-gram)	**4-grams** (truncated 7-gram)
Total	8060	6999	656	182	223
Predictions	2900	2195	379	119	207
correct	*2495*	*1931*	*326*	*91*	*147*
incorrect	*405*	*264*	*53*	*28*	*60*
Requiring back-off	5160	4804	277	63	16
Precision	86%	88%	86%	76.5%	71%
Recall	32.6%	28.7%	79.6%	59.1%	90.2%
		Back-off order 6			
Predictions	2028	2028			
correct	*1620*	*1620*			
incorrect	*408*	*408*			
Still requiring back-off	2776	2776			
Predict. orders 7+6	4223	4223			
correct	*3551*	*3551*			
incorrect	*672*	*672*			
Precision	84.1%	84.1%			
Recall	56.1%	56.1%			
		Back-off order 5			
Predictions	2180	2020	160		
correct	*1542*	*1411*	*131*		
incorrect	*638*	*609*	*29*		
Still requiring back-off	873	756	117		
Predict. orders 7 – 5	6782	6243	539		
correct	*5419*	*4962*	*457*		
incorrect	*1363*	*1281*	*82*		
Precision	79.9%	79.5%	84.8%		
Recall	86.1%	86.8%	79.6%		
		Back-off order 4			
Predictions	905	743	106	56	
correct	*488*	*382*	*68*	*38*	
incorrect	*417*	*361*	*38*	*18*	
Still requiring back-off	31	13	11	7	
Predict. orders 7 – 4	7806	6986	645	175	
correct	*5998*	*5344*	*525*	*129*	
incorrect	*1808*	*1642*	*120*	*46*	
Precision	76.8%	76.5%	81.4%	73.7%	
Recall	99.5%	99.8%	97.9%	94.9%	
		Back-off order 3			
Predictions	47	13	11	7	16
correct	*21*	*5*	*7*	*3*	*6*
incorrect	*26*	*8*	*4*	*4*	*10*
Still requiring back-off	0	0	0	0	0
Predict. orders 7 – 3	8060	6999	656	182	223
correct	*6166*	*5349*	*532*	*132*	*153*
incorrect	*1894*	*1650*	*124*	*50*	*70*
Precision	76.5%	76.4%	81.1%	72.5%	68.6%
Recall	100%	100%	100%	100%	100%

Figure 6: The results of our experiments

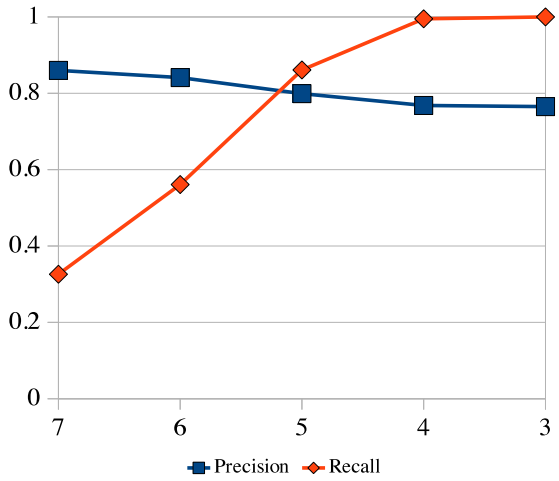


Figure 7: Development of precision and recall in relation to back-off order

Approximating 7-grams with two overlapping 6-grams as the first back-off step provides the evidence needed to correctly predict 1620 additional prepositions, with 408 additional false predictions. The number of correctly solved prediction tasks thus rises to 3551, and the number of incorrect predictions rises to 672. This back-off step almost doubles recall (56.1%). At the same time, precision drops to 84.1%. For 2776 prediction tasks, a further back-off step is necessary since still no evidence can be found for them. This pattern repeats with the back-off steps that follow. To summarize, by adding more data using less restricted contexts, more prediction tasks can be solved. The better coverage however comes at the price of reduced precision: Less specific contexts are worse predictors of the correct preposition than more specific contexts.

Figure 7 visualizes the development of precision and recall with full and truncated 7-grams counted together as in the "sum" column in Figure 6. With each back-off step, more prediction tasks can be solved (as shown by the rising recall curve). At the same time, the overall quality of the predictions drops due to the less specific contexts (as shown by the slightly dropping precision curve). While the curve for recall rises steeply, the curve for precision remains relatively flat. The back-off approach thus succeeds in adding data while preserving prediction quality.

As discussed above, we use the same set of prepositions and test corpus as De Felice (2008), but only make use of 8060 test cases. Figure 8 shows that the accuracy stabilizes quickly after about 1000 predictions, so that the difference in the size of the test set should have no impact on the reported results.

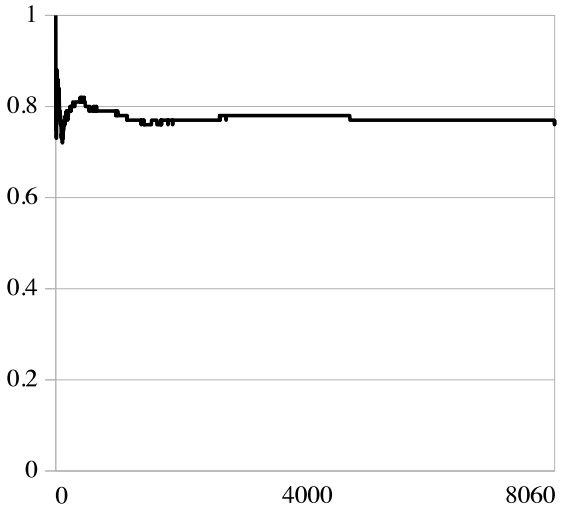


Figure 8: The accuracy of the n-gram prediction stabilizes quickly.

4 Conclusions and Outlook

In this paper, we explored the potential and the limits of a purely surface-based strategy of predicting prepositions in English. The use of surface-based n-grams ensures that fully specific exemplars of a particular size are stored in training, but avoiding abstractions in this way leads to the well-known data sparsity issues. We showed that using a web-as-corpus approach maximizing the size of the "training data", one can work with n-grams which are large enough to predict the occurrence of prepositions with significant precision while at the same time ensuring that these specific n-grams have actually been encountered during "training", i.e., evidence for them can be found on the web.

For the random sample of the BNC section J we tested on, the surface-based approach results in an accuracy of 77% for the 7-gram model with back-off to overlapping shorter n-grams. It thus outperforms De Felice's (2008) machine learning

approach which uses the same set of prepositions and the full BNC section J as test set. In broader terms, the result of our surface-based approach is competitive with the state-of-the art results for preposition prediction in English using machine learning to combine sophisticated sets of lexical and linguistically motivated features.

In this paper, we focused exclusively on the impact of n-gram size on preposition prediction. Limiting ourselves to pure surface-based information made it possible to maximize the "training data" by using a web-as-corpus approach. Returning from this very specific experiment to the general issue, there are two well-known approaches to remedy the data sparseness problem arising from storing large, specific surface forms in training. On the one hand, one can use smaller exemplars, which is the method we used as backoff in our experiments in this paper. This only works if the exemplars contain enough context for the linguistic property or relation that we need to capture the predictive power. On the other hand, one can abstract parts of the surface-based training instances to more general classes. The crucial question this raises is which generalizations preserve the predictive power of the exemplars and can reliably be identified. The linguistically-informed features used in the previous approaches in the literature naturally provide interesting instances of answers to this question. In the future, we intend to compare the results we obtained using the web-as-corpus approach with one based on the Google-5-gram corpus to study using controlled, incremental shallow-to-deep feature development which abstractions or linguistic generalizations best preserve the predictive context while lowering the demands on the size of the training data.

Turning to a linguistic issue, it could be useful to distinguish between lexical and functional prepositions when reporting test results. This is an important distinction because the information needed to predict functional prepositions typically is in the local context, whereas the information needed to predict lexical prepositions is not necessarily present locally. To illustrate, a competent human speaker presented with the sentence *John is dependent ___ his brother* and asked to fill in the missing preposition, would correctly pick *on*. This is a case of a functional preposition where the relevant information is locally present: the adjective *dependent* selects *on*. On the other hand, the sentence *John put his bag ___ the table* is more problematic, even for a human, since both *on* and *under* are reasonable choices; the information needed to predict the omitted preposition in this case is not locally present. In line with the previous research, in the work in this paper we made predictions for all prepositions alike. In the future, it could be useful to annotate the test set so that one can distinguish functional and lexical uses and report separate figures for these two classes in order to empirically confirm their differences with respect to locality.

References

Bergsma, Shane, Dekang Lin, and Randy Goebel. 2009. Web-scale n-gram models for lexical disambiguation. In *IJCAI'09: Proceedings of the 21st international jont conference on Artifical intelligence*, pages 1507–1512, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Boyd, Adriane, Markus Dickinson, and Detmar Meurers. 2008. On detecting errors in dependency treebanks. *Research on Language and Computation*, 6(2):113–137.

Chodorow, Martin, Joel Tetreault, and Na-Rae Han. 2007. Detection of grammatical errors involving prepositions. In *Proceedings of the 4th ACL-SIGSEM Workshop on Prepositions*, pages 25–30, Prague, Czech Republic, June.

De Felice, Rachele and Stephen Pulman. 2007. Automatically acquiring models of preposition use. In *Proceedings of the 4th ACL-SIGSEM Workshop on Prepositions*, pages 45–50, Prague, Czech Republic, June. Association for Computational Linguistics.

De Felice, Rachele. 2008. *Automatic Error Detection in Non-native English*. Ph.D. thesis, St Catherine's College, University of Oxford.

Dickinson, Markus and W. Detmar Meurers. 2003. Detecting errors in part-of-speech annotation. In *Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics (EACL-03)*, pages 107–114, Budapest, Hungary.

Dickinson, Markus and W. Detmar Meurers. 2005. Detecting errors in discontinuous structural anno-

tation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 322–329.

Gamon, Michael, Jianfeng Gao, Chris Brockett, Alexander Klementiev, William Dolan, Dmitriy Belenko, and Lucy Vanderwende. 2008. Using contextual speller techniques and language modeling for esl error correction. In *Proceedings of IJCNLP*, Hyderabad, India.

Lapata, Mirella and Frank Keller. 2005. Web-based models for natural language processing. *ACM Transactions on Speech and Language Processing*, 2(1):1–30, February.

Lee, John and Ola Knutsson. 2008. The role of pp attachment in preposition generation. In Gelbukh, A., editor, *Proceedings of CICLing 2008*.

Tetreault, Joel and Martin Chodorow. 2008a. Native judgments of non-native usage: Experiments in preposition error detection. In *Proceedings of COLING-08*, Manchester.

Tetreault, Joel and Martin Chodorow. 2008b. The ups and downs of preposition error detection in esl writing. In *Proceedings of COLING-08*, Manchester.

A Comparison of Features for Automatic Readability Assessment

Lijun Feng
City University of New York
lijun7.feng@gmail.com

Martin Jansche
Google, Inc.
jansche@acm.org

Matt Huenerfauth
City University of New York
matt@cs.qc.cuny.edu

Noémie Elhadad
Columbia University
noemie@dbmi.columbia.edu

Abstract

Several sets of explanatory variables – including shallow, language modeling, POS, syntactic, and discourse features – are compared and evaluated in terms of their impact on predicting the grade level of reading material for primary school students. We find that features based on in-domain language models have the highest predictive power. Entity-density (a discourse feature) and POS-features, in particular nouns, are individually very useful but highly correlated. Average sentence length (a shallow feature) is more useful – and less expensive to compute – than individual syntactic features. A judicious combination of features examined here results in a significant improvement over the state of the art.

1 Introduction

1.1 Motivation and Method

Readability Assessment quantifies the difficulty with which a reader understands a text. Automatic readability assessment enables the selection of appropriate reading material for readers of varying proficiency. Besides modeling and understanding the linguistic components involved in readability, a readability-prediction algorithm can be leveraged for the task of automatic text simplification: as simplification operators are applied to a text, the readability is assessed to determine whether more simplification is needed or a particular reading level was reached.

Identifying text properties that are strongly correlated with text complexity is itself complex. In this paper, we explore a broad range of text properties at various linguistic levels, ranging from discourse features to language modeling features, part-of-speech-based grammatical features, parsed syntactic features and well studied shallow features, many of which are inspired by previous work.

We use grade levels, which indicate the number of years of education required to completely understand a text, as a proxy for reading difficulty. The corpus in our study consists of texts labeled with grade levels ranging from grade 2 to 5. We treat readability assessment as a classification task and evaluate trained classifiers in terms of their prediction accuracy. To investigate the contributions of various sets of features, we build prediction models and examine how the choice of features influences the model performance.

1.2 Related Work

Many traditional readability metrics are linear models with a few (often two or three) predictor variables based on superficial properties of words, sentences, and documents. These shallow features include the average number of syllables per word, the number of words per sentence, or binned word frequency. For example, the Flesch-Kincaid Grade Level formula uses the average number of words per sentence and the average number of syllables per word to predict the grade level (Flesch, 1979). The Gunning FOG index (Gunning, 1952) uses average sentence length and the percentage of words with at least three syllables. These traditional metrics are easy to compute and use, but they are not reliable, as demonstrated by several recent studies in the field (Si and Callan, 2001; Petersen and Ostendorf, 2006; Feng et al., 2009).

Coling 2008: Poster Volume, pages 276–284,
Beijing, August 2010

With the advancement of natural language processing tools, a wide range of more complex text properties have been explored at various linguistic levels. Si and Callan (2001) used unigram language models to capture content information from scientific web pages. Collins-Thompson and Callan (2004) adopted a similar approach and used a smoothed unigram model to predict the grade levels of short passages and web documents. Heilman et al. (2007) continued using language modeling to predict readability for first and second language texts. Furthermore, they experimented with various statistical models to test their effectiveness at predicting reading difficulty (Heilman et al., 2008). Schwarm/Petersen and Ostendorf (Schwarm and Ostendorf, 2005; Petersen and Ostendorf, 2006) used support vector machines to combine features from traditional reading level measures, statistical language models and automatic parsers to assess reading levels. In addition to lexical and syntactic features, several researchers started to explore discourse level features and examine their usefulness in predicting text readability. Pitler and Nenkova (2008) used the Penn Discourse Treebank (Prasad et al., 2008) to examine discourse relations. We previously used a lexical-chaining tool to extract entities that are connected by certain semantic relations (Feng et al., 2009).

In this study, we systematically evaluate all above-mentioned types of features, as well as a few extensions and variations. A detailed description of the features appears in Section 3. Section 4 discusses results of experiments with classifiers trained on these features. We begin with a description of our data in the following section.

2 Corpus

We contacted the Weekly Reader[1] corporation, an on-line publisher producing magazines for elementary and high school students, and were granted access in October 2008 to an archive of their articles. Among the articles retrieved, only those for elementary school students are labeled with grade levels, which range from 2 to 5. We selected only this portion of articles (1629 in total) for the study.[2] These articles are intended to build children's general knowledge and help them practice reading skills. While pre-processing the texts, we found that many articles, especially those for lower grade levels, consist of only puzzles and quizzes, often in the form of simple multiple-choice questions. We discarded such texts and kept only 1433 full articles. Some distributional statistics of the final corpus are listed in Table 1.

Table 1: Statistics for the Weekly Reader Corpus

Grade	docs.	words/document		words/sentence	
		mean	std. dev.	mean	std. dev.
2	174	128.27	106.03	9.54	2.32
3	289	171.96	106.05	11.39	2.42
4	428	278.03	187.58	13.67	2.65
5	542	335.56	230.25	15.28	3.21

3 Features

3.1 Discourse Features

We implement four subsets of discourse features: entity-density features, lexical-chain features, coreference inference features and entity grid features. The coreference inference features are novel and have not been studied before. We previously studied entity-density features and lexical-chain features for readers with intellectual disabilities (Feng et al., 2009). Entity-grid features have been studied by Barzilay and Lapata (2008) in a stylistic classification task. Pitler and Nenkova (2008) used the same features to evaluate how well a text is written. We replicate this set of features for grade level prediction task.

3.1.1 Entity-Density Features

Conceptual information is often introduced in a text by entities, which consist of general nouns and named entities, e.g. people's names, locations, organizations, etc. These are important in text comprehension, because established entities form basic components of concepts and propositions, on which higher level discourse processing is based. Our prior work illustrated the importance of entities in text comprehension (Feng et al., 2009).

[1] http://www.weeklyreader.com

[2] A corpus of Weekly Reader articles was previously used in work by Schwarm and Ostendorf (2005). However, the two corpora are not identical in size nor content.

Table 2: New Entity-Density Features

1	percentage of named entities per document
2	percentage of named entities per sentences
3	percentage of overlapping nouns removed
4	average number of remaining nouns per sentence
5	percentage of named entities in total entities
6	percentage of remaining nouns in total entities

We hypothesized that the number of entities introduced in a text relates to the working memory burden on their targeted readers – individuals with intellectual disabilities. We defined entities as a union of named entities and general nouns (nouns and proper nouns) contained in a text, with overlapping general nouns removed. Based on this, we implemented four kinds of entity-density features: total number of entity mentions per document, total number of unique entity mentions per document, average number of entity mentions per sentence, and average number of unique entity mentions per sentence.

We believe entity-density features may also relate to the readability of a text for a general audience. In this paper, we conduct a more refined analysis of general nouns and named entities. To collect entities for each document, we used OpenNLP's[3] name-finding tool to extract named entities; general nouns are extracted from the output of Charniak's Parser (see Section 3.3). Based on the set of entities collected for each document, we implement 12 new features. We list several of these features in in Table 2.

3.1.2 Lexical Chain Features

During reading, a more challenging task with entities is not just to keep track of them, but to resolve the semantic relations among them, so that information can be processed, organized and stored in a structured way for comprehension and later retrieval. In earlier work (Feng et al., 2009), we used a lexical-chaining tool developed by Galley and McKeown (2003) to annotate six semantic relations among entities, e.g. synonym, hypernym, hyponym, etc. Entities that are connected by these semantic relations were linked through the text to form lexical chains. Based on these chains, we implemented six features, listed in Table 3, which we use in our current study. The length of a chain is the number of entities contained in the chain, the span of chain is the distance between the index of the first and last entity in a chain. A chain is defined to be active for a word or an entity if this chain passes through its current location.

[3] http://opennlp.sourceforge.net/

Table 3: Lexical Chain Features

1	total number of lexical chains per document
2	avg. lexical chain length
3	avg. lexical chain span
4	num. of lex. chains with span $\geq$ half doc. length
5	num. of active chains per word
6	num. of active chains per entity

Table 4: Coreference Chain Features

1	total number of coreference chains per document
2	avg. num. of coreferences per chain
3	avg. chain span
4	num. of coref. chains with span $\geq$ half doc. length
5	avg. inference distance per chain
6	num. of active coreference chains per word
7	num. of active coreference chains per entity

3.1.3 Coreference Inference Features

Relations among concepts and propositions are often not stated explicitly in a text. Automatically resolving implicit discourse relations is a hard problem. Therefore, we focus on one particular type, referential relations, which are often established through anaphoric devices, e.g. pronominal references. The ability to resolve referential relations is important for text comprehension.

We use OpenNLP to resolve coreferences. Entities and pronominal references that occur across the text and refer to the same person or object are extracted and formed into a coreference chain. Based on the chains extracted, we implement seven features as listed in Table 4. The chain length, chain span and active chains are defined in a similar way to the lexical chain features. Inference distance is the difference between the index of the referent and that of its pronominal reference. If the same referent occurs more than once in a chain, the index of the closest occurrence is used when computing the inference distance.

3.1.4 Entity Grid Features

Coherent texts are easier to read. Several computational models have been developed to represent and

measure discourse coherence (Lapata and Barzilay, 2005; Soricut and Marcu, 2006; Elsner et al., 2007; Barzilay and Lapata, 2008) for NLP tasks such as text ordering and text generation. Although these models are not intended directly for readability research, Barzilay and Lapata (2008) have reported that distributional properties of local entities generated by their grid models are useful in detecting original texts from their simplified versions when combined with well studied lexical and syntactic features. This approach was subsequently pursued by Pitler and Nenkova (2008) in their readability study. Barzilay and Lapata's entity grid model is based on the assumption that the distribution of entities in locally coherent texts exhibits certain regularities. Each text is abstracted into a grid that captures the distribution of entity patterns at the level of sentence-to-sentence transitions. The entity grid is a two-dimensional array, with one dimension corresponding to the salient entities in the text, and the other corresponding to each sentence of the text. Each grid cell contains the grammatical role of the specified entity in the specified sentence: whether it is a subject (S), object (O), neither of the two (X), or absent from the sentence (-).

We use the Brown Coherence Toolkit (v0.2) (Elsner et al., 2007), based on (Lapata and Barzilay, 2005), to generate an entity grid for each text in our corpus. The distribution patterns of entities are traced between each pair of adjacent sentences, resulting in 16 entity transition patterns[4]. We then compute the distribution probability of each entity transition pattern within a text to form 16 entity-grid-based features.

3.2 Language Modeling Features

Our language-modeling-based features are inspired by Schwarm and Ostendorf's (2005) work, a study that is closely related to ours. They used data from the same data – the Weekly Reader – for their study. They trained three language models (unigram, bigram and trigram) on two paired complex/simplified corpora (Britannica and LiteracyNet) using an approach in which words with high information gain are kept and the remaining words are replaced with their parts of speech. These language models were then used to score each text in the Weekly Reader corpus by perplexity. They reported that this approach was more successful than training LMs on text sequences of word labels alone, though without providing supporting statistics.

It's worth pointing out that their LMs were not trained on the Weekly Reader data, but rather on two unrelated paired corpora (Britannica and LiteracyNet). This seems counter-intuitive, because training LMs directly on the Weekly Reader data would provide more class-specific information for the classifiers. They justified this choice by stating that splitting limited Weekly Reader data for training and testing purposes resulted in unsuccessful performance.

We overcome this problem by using a hold-one-out approach to train LMs directly on our Weekly Reader corpus, which contains texts ranging from Grade 2 to 5. We use grade levels to divide the whole corpus into four smaller subsets. In addition to implementing Schwarm and Ostendorf's information-gain approach, we also built LMs based on three other types of text sequences for comparison purposes. These included: word-token-only sequence (i.e., the original text), POS-only sequence, and paired word-POS sequence. For each grade level, we use the SRI Language Modeling Toolkit[5] (with Good-Turing discounting and Katz backoff for smoothing) to train 5 language models (1- to 5-gram) using each of the four text sequences, resulting in $4 \times 5 \times 4 = 80$ perplexity features for each text tested.

3.3 Parsed Syntactic Features

Schwarm and Ostendorf (2005) studied four parse tree features (average parse tree height, average number of SBARs, noun phrases, and verb phrases per sentences). We implemented these and additional features, using the Charniak parser (Charniak, 2000). Our parsed syntactic features focus on clauses (SBAR), noun phrases (NP), verb phrases (VP) and prepositional phrases (PP). For each phrase, we implement four features: total number of the phrases per document, average number of phrases per sentence, and average phrase length

[4]These 16 transition patterns are: "SS", "SO", "SX", "S-", "OS", "OO", "OX", "O-", "XS", "XO", "XX", "X-", "-S", "-O", "-X", "--".

[5]http://www.speech.sri.com/projects/srilm/

measured by number of words and characters respectively. In addition to average tree height, we implement two non-terminal-node-based features: average number of non-terminal nodes per parse tree, and average number of non-terminal nodes per word (terminal node).

3.4 POS-based Features

Part-of-speech-based grammatical features were shown to be useful in readability prediction (Heilman et al., 2007; Leroy et al., 2008). To extend prior work, we systematically studied a number of common categories of words and investigated to what extent they are related to a text's complexity. We focus primarily on five classes of words (nouns, verbs, adjectives, adverbs, and prepositions) and two broad categories (content words, function words). Content words include nouns, verbs, numerals, adjectives, and adverbs; the remaining types are function words. The part of speech of each word is obtained from examining the leaf node based on the output of Charniak's parser, where each leaf node consists of a word and its part of speech. We group words based on their POS labels. For each class of words, we implement five features. For example, for the adjective class, we implemented the following five features: percent of adjectives (tokens) per document, percent of unique adjectives (types) per document, ratio of unique adjectives per total unique words in a document, average number of adjectives per sentence and average number of unique adjectives per sentence.

3.5 Shallow Features

Shallow features refer to those used by traditional readability metrics, such as Flesch-Kincaid Grade Level (Flesch, 1979), SMOG (McLaughlin, 1969), Gunning FOG (Gunning, 1952), etc. Although recent readability studies have strived to take advantage of NLP techniques, little has been revealed about the predictive power of shallow features. Shallow features, which are limited to superficial text properties, are computationally much less expensive than syntactic or discourse features. To enable a comparison against more advanced features, we implement 8 frequently used shallow features as listed in Table 5.

Table 5: Shallow Features

1	average number of syllables per word
2	percentage of poly-syll. words per doc.
3	average number of poly-syll. words per sent.
4	average number of characters per word
5	Chall-Dale difficult words rate per doc.
6	average number of words per sentence
7	Flesch-Kincaid score
8	total number of words per document

3.6 Other Features

For comparison, we replicated 6 out-of-vocabulary features described in Schwarm and Ostendorf (2005). For each text in the Weekly Reader corpus, these 6 features are computed using the most common 100, 200 and 500 word tokens and types based on texts from Grade 2. We also replicated the 12 perplexity features implemented by Schwarm and Ostendorf (2005) (see Section 3.2).

4 Experiments and Discussion

Previous studies on reading difficulty explored various statistical models, e.g. regression vs. classification, with varying assumptions about the measurement of reading difficulty, e.g. whether labels are ordered or unrelated, to test the predictive power of models (Heilman et al., 2008; Petersen and Ostendorf, 2009; Aluisio et al., 2010). In our research, we have used various models, including linear regression; standard classification (Logistic Regression and SVM), which assumes no relation between grade levels; and ordinal regression/ classification (provided by Weka, with Logistic Regression and SMO as base function), which assumes that the grade levels are ordered. Our experiments show that, measured by mean squared error and classification accuracy, linear regression models perform considerably poorer than classification models. Measured by accuracy and F-measure, ordinal classifiers perform comparable or worse than standard classifiers. In this paper, we present the best results, which are obtained by standard classifiers. We use two machine learning packages known for efficient high-quality multi-class classification: LIBSVM (Chang and Lin, 2001) and the Weka machine learning toolkit (Hall et al., 2009), from which we choose Logistic Regression as classifiers. We train and evaluate various prediction

Table 6: Comparison of discourse features

Feature Set	LIBSVM	Logistic Regress.
Entity-Density	59.63 ± 0.632	57.59 ± 0.375
Lexical Chain	45.86 ± 0.815	42.58 ± 0.241
Coref. Infer.	40.93 ± 0.839	42.19 ± 0.238
Entity Grid	45.92 ± 1.155	42.14 ± 0.457
all combined	60.50 ± 0.990	58.79 ± 0.703

Table 7: Comparison of lang. modeling features

Feature Set	LIBSVM	Logistic Regress.
IG	62.52 ± 1.202	62.14 ± 0.510
Text-only	60.17 ± 1.206	60.31 ± 0.559
POS-only	56.21 ± 2.354	57.64 ± 0.391
Word/POS pair	60.38 ± 0.820	59.00 ± 0.367
all combined	68.38 ± 0.929	66.82 ± 0.448
IG by Schwarm	52.21 ± 0.832	51.89 ± 0.405

Table 8: Comparison of parsed syntactic features

Feature Set	# Feat.	LIBSVM
Original features	4	50.68 ± 0.812
Expanded features	21	57.79 ± 1.023

models using the features described in Section 3. We evaluate classification accuracy using repeated 10-fold cross-validation on the Weekly Reader corpus. Classification accuracy is defined as the percentage of texts predicted with correct grade levels. We repeat each experiment 10 times and report the mean accuracy and its standard deviation.

4.1 Discourse Features

We first discuss the improvement made by extending our earlier entity-density features (Feng et al., 2009). We used LIBSVM to train and test models on the Weekly Reader corpus with our earlier features and our new features respectively. With earlier features only, the model achieves 53.66% accuracy. With our new features added, the model performance is 59.63%.

Table 6 presents the classification accuracy of models trained with discourse features. We see that, among four subsets of discourse features, entity-density features perform significantly better than the other three feature sets and generate the highest classification accuracy (LIBSVM: 59.63%, Logistic Regression: 57.59%). While Logistic Regression results show that there is not much performance difference among lexical chain, coreference inference, and entity grid features, classification accuracy of LIBSVM models indicates that lexical chain features and entity grid features are better in predicting text readability than coreference inference features. Combining all discourse features together does not significantly improve accuracy compared with models trained only with entity-density features.

4.2 Language Modeling Features

Table 7 compares the performance of models generated using our approach and our replication of Schwarm and Ostendorf's (2005) approach. In our approach, features were obtained from language models trained on the Weekly Reader corpus. Not surprisingly, these are more effective than LMs trained on the Britannica and LiteracyNet corpora, in Schwarm and Ostendorf's approach. Our results support their claim that LMs trained with information gain outperform LMs trained with POS labels. However, we also notice that training LMs on word labels alone or paired word/POS sequences achieved similar classification accuracy to the IG approach, while avoiding the complicated feature selection of the IG approach.

4.3 Parsed Syntactic Features

Table 8 compares a classifier trained on the four parse features of Schwarm and Ostendorf (2005) to a classifier trained on our expanded set of parse features. The LIBSVM classifier with the expanded feature set scored 7 points higher than the one trained on only the original four features, improving from 50.68% to 57.79%. Table 9 shows a detailed comparison of particular parsed syntactic features. The two non-terminal-node-based features (average number of non-terminal nodes per tree and average number of non-terminal nodes per word) have higher discriminative power than average tree height. Among SBARs, NPs, VPs and PPs, our experiments show that VPs and NPs are the best predictors.

4.4 POS-based Features

The classification accuracy generated by models trained with various POS features is presented in Table 10. We find that, among the five word classes investigated, noun-based features gener-

Table 9: Detailed comp. of syntactic features

Feature Set	LIBSVM	Logistic Regress.
Non-term.-node ratios	53.02 ± 0.571	51.80 ± 0.171
Average tree height	44.26 ± 0.914	43.45 ± 0.269
SBARs	44.42 ± 1.074	43.50 ± 0.386
NPs	51.56 ± 1.054	48.14 ± 0.408
VPs	53.07 ± 0.597	48.67 ± 0.484
PPs	49.36 ± 1.277	46.47 ± 0.374
all combined	57.79 ± 1.023	54.11 ± 0.473

Table 10: Comparison of POS features

Feature Set	LIBSVM	Logistic Regress.
Nouns	58.15 ± 0.862	57.01 ± 0.256
Verbs	54.40 ± 1.029	55.10 ± 0.291
Adjectives	53.87 ± 1.128	52.75 ± 0.427
Adverbs	52.66 ± 0.970	50.54 ± 0.327
Prepositions	56.77 ± 1.278	54.13 ± 0.312
Content words	56.84 ± 1.072	56.18 ± 0.213
Function words	52.19 ± 1.494	50.95 ± 0.298
all combined	59.82 ± 1.235	57.86 ± 0.547

ate the highest classification accuracy, which is consistent with what we have observed earlier about entity-density features. Another notable observation is that prepositions demonstrate higher discriminative power than adjectives and adverbs. Models trained with preposition-based features perform close to those trained with noun-based features. Among the two broader categories, content words (which include nouns) demonstrate higher predictive power than function words (which include prepositions).

4.5 Shallow Features

We present some notable findings on shallow features in Table 11. Experimental results generated by models trained with Logistic Regression show that average sentence length has dominating predictive power over all other shallow features. Features based on syllable counting perform much worse. The Flesch-Kincaid Grade Level score uses a fixed linear combination of average words per sentence and average syllables per word. Combining those two features (without fixed coefficients) results in the best overall accuracy, while using the Flesch-Kincaid score as a single feature is significantly worse.

Table 11: Comparison of shallow features

Feature Set	Logistic Regress.
Avg. words per sent.	52.17 ± 0.193
Avg. syll. per word	42.51 ± 0.264
above two combined	53.04 ± 0.514
Flesch-Kincaid score	50.83 ± 0.144
Avg. poly-syll. words per sent.	45.70 ± 0.306
all 8 features combined	52.34 ± 0.242

4.6 Comparison with Previous Studies

A trivial baseline of predicting the most frequent grade level (grade 5) predicts 542 out of 1433 texts (or 37.8%) correctly. With this in mind, we first compare our study with the widely-used Flesch-Kincaid Grade Level formula, which is a linear function of average words per sentence and average syllables per word that aims to predict the grade level of a text directly. Since this is a fixed formula with known coefficients, we evaluated it directly on our entire Weekly Reader corpus without cross-validation. We obtain the predicted grade level of a text by rounding the Flesch-Kincaid score to the nearest integer. For only 20 out of 1433 texts the predicted and labeled grade levels agree, resulting in a poor accuracy of 1.4%. By contrast, using the Flesch-Kincaid score as a feature of a simple logistic regression model achieves above 50% accuracy, as discussed in Section 4.5.

The most closely related previous study is the work of Schwarm and Ostendorf (2005). However, because their experiment design (85/15 training/test data split) and machine learning tool (SVM^{light}) differ from ours, their results are not directly comparable to ours. To make a comparison, we replicated all the features used in their study and then use LIBSVM and Weka's Logistic Regression to train two models with the replicated features and evaluate them on our Weekly Reader corpus using 10-fold cross-validation.

Using the same experiment design, we train classifiers with three combinations of our features as listed in Table 12. "All features" refers to a naive combination of all features. "AddOneBest" refers to a subset of features selected by a group-wise add-one-best greedy feature selection. "WekaFS" refers to a subset of features chosen by Weka's feature selection filter.

"WekaFS" consists of 28 features selected au-

Table 12: Comparison with previous work

baseline accuracy (majority class)		37.8	
Flesch-Kincaid Grade Level		1.4	
Feature Set	# Feat.	LIBSVM	Logistic Reg.
Schwarm	25	63.18 ± 1.664	60.50 ± 0.477
All features	273	72.21 ± 0.821	63.71 ± 0.576
AddOneBest	122	74.01 ± 0.847	69.22 ± 0.411
WekaFS	28	70.06 ± 0.777	65.46 ± 0.336

tomatically by Weka's feature selection filter using a best-first search method. The 28 features include language modeling features, syntactic features, POS features, shallow features and out-of-vocabulary features. Aside from 4 shallow features and 5 out-of-vocabulary features, the other 19 features are novel features we have implemented for this paper.

As Table 12 shows, a naive combination of all features results in classification accuracy of 72%, which is much higher than the current state of the art (63%). This is not very surprising, since we are considering a greater variety of features than any previous individual study. Our WekaFS classifier uses roughly the same number of features as the best published result, yet it has a higher accuracy (70.06%). Our best results were obtained by group-wise add-one-best feature selection, resulting in 74% classification accuracy, a big improvement over the state of the art.

5 Conclusions

We examined the usefulness of features at various linguistic levels for predicting text readability in terms of assigning texts to elementary school grade levels. We implemented a set of discourse features, enriched previous work by creating several new features, and systematically tested and analyzed the impact of these features.

We observed that POS features, in particular nouns, have significant predictive power. The high discriminative power of nouns in turn explains the good performance of entity-density features, based primarily on nouns. In general, our selected POS features appear to be more correlated to text complexity than syntactic features, shallow features and most discourse features.

For parsed syntactic features, we found that verb phrases appear to be more closely correlated with text complexity than other types of phrases. While SBARs are commonly perceived as good predictors for syntactic complexity, they did not prove very useful for predicting grade levels of texts in this study. In future work, we plan to examine this result in more detail.

Among the 8 shallow features, which are used in various traditional readability formulas, we identified that average sentence length has dominating predictive power over all other lexical or syllable-based features.

Not surprisingly, among language modeling features, combined features obtained from LMs trained directly on the Weekly Reader corpus show high discriminative power, compared with features from LMs trained on unrelated corpora.

Discourse features do not seem to be very useful in building an accurate readability metric. The reason could lie in the fact that the texts in the corpus we studied exhibit relatively low complexity, since they are aimed at primary-school students. In future work, we plan to investigate whether these discourse features exhibit different discriminative power for texts at higher grade levels.

A judicious combination of features examined here results in a significant improvement over the state of the art.

References

Sandra Aluisio, Lucia Specia, Caroline Gasperin, and Carolina Scarton. 2010. Readability assessment for text simplification. In *NAACL-HLT 2010: The 5th Workshop on Innovative Use of NLP for Building Educational Applications*.

Regina Barzilay and Mirella Lapata. 2008. Modeling local coherence: An entity-based approach. *Computational Linguistics*, 34(1):1–34.

Chih-Chung Chang and Chih-Jen Lin. 2001. *LIBSVM: A Library for Support Vector Machines*. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

Eugene Charniak. 2000. A maximum-entropy-inspired parser. In *Proceedings of the 1st Conference of the North American Chapter of the ACL*, pages 132–139.

Kevyn Collins-Thompson and Jamie Callan. 2004. A language modeling approach to predicting reading difficulty. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL 2004)*.

Micha Elsner, Joseph Austerweil, and Eugene Charniak. 2007. A unified local and global model for discourse coherence. In *Proceedings of the Conference on Human Language Technology and North American chapter of the Association for Computational Linguistics (HLT-NAACL 2007)*.

Lijun Feng, Noémie Elhadad, and Matt Huenerfauth. 2009. Cognitively motivated features for readability assessment. In *The 12th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2009)*.

Rudolf Flesch. 1979. *How to write plain English*. Harper and Brothers, New York.

Michel Galley and Kathleen McKeown. 2003. Improving word sense disambiguation in lexical chaining. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*.

Robert Gunning. 1952. *The Technique of Clear Writing*. McGraw-Hill.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. 2009. The WEKA data mining software: An update. *SIGKDD Explorations*, 11(1):10–18.

Michael J. Heilman, Kevyn Collins-Thompson, Jamie Callan, and Maxine Eskenazi. 2007. Combining lexical and grammatical features to improve readability measures for first and second language texts. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics*.

Michael J. Heilman, Kevyn Collins-Thompson, and Maxine Eskenazi. 2008. An analysis of statistical models and features for reading difficulty prediction. In *ACL 2008: The 3rd Workshop on Innovative Use of NLP for Building Educational Applications*.

Mirella Lapata and Regina Barzilay. 2005. Automatic evaluation of text coherence: Models and representations. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI'05)*, pages 1085–1090.

Gondy Leroy, Stephen Helmreich, James R. Cowie, Trudi Miller, and Wei Zheng. 2008. Evaluating online health information: Beyond readability formulas. In *AMIA 2008 Symposium Proceedings*.

G. Harry McLaughlin. 1969. Smog grading a new readability formula. *Journal of Reading*, 12(8):639–646.

Sarah E. Petersen and Mari Ostendorf. 2006. A machine learning approach to reading level assessment. Technical report, University of Washington CSE Technical Report.

Sarah E. Petersen and Mari Ostendorf. 2009. A machine learning approach to reading level assessment. *Computer Speech and Language*, 23:89–106.

Emily Pitler and Ani Nenkova. 2008. Revisiting readability: A unified framework for predicting text quality. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*.

R. Prasad, N. Dinesh, A. Lee, E. Miltsakaki, L. Robaldo, A. Joshi, and B. Webber. 2008. The Penn discourse treebank. In *The Sixth International Conference on Language Resources and Evaluation (LREC'08)*.

Sarah E. Schwarm and Mari Ostendorf. 2005. Reading level assessment using support vector machines and statistical language models. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*.

Luo Si and Jamie Callan. 2001. A statistical model for scientific readability. In *Proceedings of the Tenth International Conference on Information and Knowledge Management*.

Radu Soricut and Daniel Marcu. 2006. Discourse generation using utility-trained coherence models. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*.

An Efficient Shift-Reduce Decoding Algorithm for Phrased-Based Machine Translation

Yang Feng, Haitao Mi, Yang Liu and Qun Liu
Key Laboratory of Intelligent Information Processing
Institute of Computing Technology
Chinese Academy of Sciences
{fengyang,htmi,yliu,liuqun}@ict.ac.cn

Abstract

In statistical machine translation, decoding without any reordering constraint is an NP-hard problem. Inversion Transduction Grammars (ITGs) exploit linguistic structure and can well balance the needed flexibility against complexity constraints. Currently, translation models with ITG constraints usually employs the cube-time CYK algorithm. In this paper, we present a shift-reduce decoding algorithm that can generate ITG-legal translation from left to right in linear time. This algorithm runs in a *reduce-eager* style and is suited to phrase-based models. Using the state-of-the-art decoder Moses as the baseline, experiment results show that the shift-reduce algorithm can significantly improve both the accuracy and the speed on different test sets.

1 Introduction

In statistical machine translation, for the diversity of natural languages, the word order of source and target language may differ and searching through all possible translations is NP-hard (Knight, 1999). So some measures have to be taken to reduce search space: either using a search algorithm with pruning technique or restricting possible reorderings.

Currently, beam search is widely used (Tillmann and Ney, 2003; Koehn, 2004) to reduce search space. However, the pruning technique adopted by this algorithm is not risk-free. As a result, the best partial translation may be ruled out during pruning. The more aggressive the pruning is, the more likely the best translation escapes. There should be a tradeoff between the speed and the accuracy. If some heuristic knowledge is employed to guide the search, the search algorithm can discard some implausible hypotheses in advance and focus on more possible ones.

Inversion Transduction Grammars (ITGs) permit a minimal extra degree of ordering flexibility and are particularly well suited to modeling ordering shifts between languages (Wu, 1996; Wu, 1997). They can well balance the needed flexibility against complexity constraints. Recently, ITG has been successfully applied to statistical machine translation (Zens and Ney, 2003; Zens et al., 2004; Xiong et al., 2006). However, ITG generally employs the expensive CYK parsing algorithm which runs in cube time. In addition, the CYK algorithm can not calculate language model exactly in the process of decoding, as it can not catch the full history context of the left words in a hypothesis.

In this paper, we introduce a shift-reduce decoding algorithm with ITG constraints which runs in a left-to-right manner. This algorithm parses source words in the order of their corresponding translations on the target side. In the meantime, it gives all candidate ITG-legal reorderings. The shift-reduce algorithm is different from the CYK algorithm, in particular:

- It produces translation in a left-to-right manner. As a result, language model probability can be calculated more precisely in the light of full history context.

- It decodes much faster. Applied with distor-

Coling 2008: Poster Volume, pages 285–293,
Beijing, August 2010

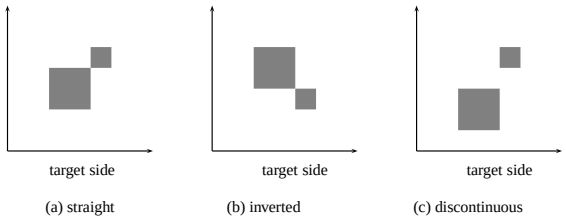


Figure 1: Orientation of two blocks.

tion limit, shift-reduce decoding algorithm can run in linear time, while the CYK runs in cube time.

- It holds ITG structures generated during decoding. That is to say, it can directly give ITG-legal spans, which leads to faster decoding. Furthermore, it can be extended to syntax-based models.

We evaluated the performance of the shift-reduce decoding algorithm by adding ITG constraints to the state-of-the-art decoder Moses. We did experiments on three data sets: NIST MT08 data set, NIST MT05 data set and China Workshop on Machine Translation 2007 data set. Compared to Moses, the improvements of the accuracy are 1.59, 0.62, 0.8 BLEU score, respectively, and the speed improvements are 15%, 24%, 30%, respectively.

2 Decoding with ITG constraints

In this paper, we employ the shift-reduce algorithm to add ITG constraints to phrase-based machine translation model. It is different from the traditional shift-reduce algorithm used in natural language parsing. On one hand, as natural language parsing has to cope with a high degree of ambiguity, it need take ambiguity into consideration. As a result, the traditional one often suffers *shift-reduce* divergence. Nonetheless, the shift-reduce algorithm in this paper does not pay attention to ambiguity and acts in a *reduce-eager* manner. On the other hand, the traditional algorithm can not ensure that all reorderings observe ITG constraints, so we have to modify the traditional algorithm to import ITG constraints.

We will introduce the shift-reduce decoding algorithm in the following two steps: First, we will deduce how to integrate the shift-reduce algorithm and ITG constraints and show its correctness (Section 2.1). Second, we will describe the shift-reduce decoding algorithm in details (Section 2.2).

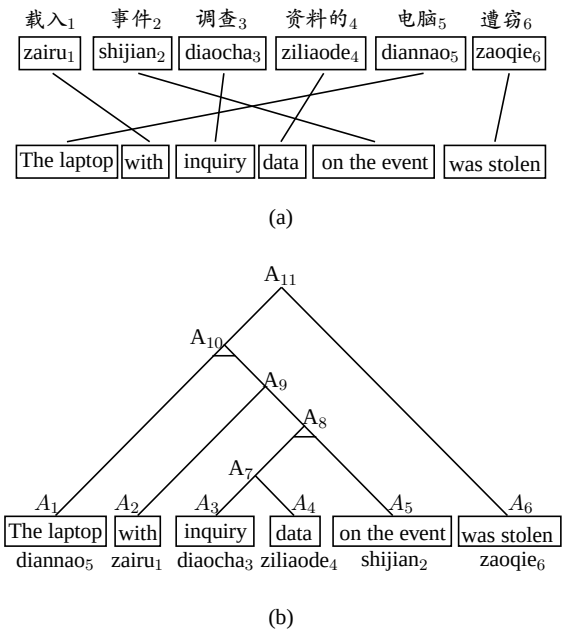


Figure 2: A Chinese-to-English sentence pair and its corresponding ITG tree.

2.1 Adding ITG constraints

In the process of decoding, a source phrase is regarded as a block and a source sentence is seen as a sequence of blocks. The orientation of two blocks whose translations are adjacent on the target side can be straight, inverted or discontinuous, as shown in Figure 1. According to ITG, two blocks which are straight or inverted can be merged into a single block. For parsing, different mergence order of a sequence of continuous blocks may yield different derivations. In contrast, the phrase-based machine translation does not compute reordering probabilities hierarchically, so the mergence order will not impact the computation of reordering probabilities. As a result, the shift-reduce decoding algorithm need not take into consideration the shift-reduce divergence. It merges two continuous blocks as soon as possible, acting in a reduce-eager style.

Every ITG-legal sentence pair has a corre-

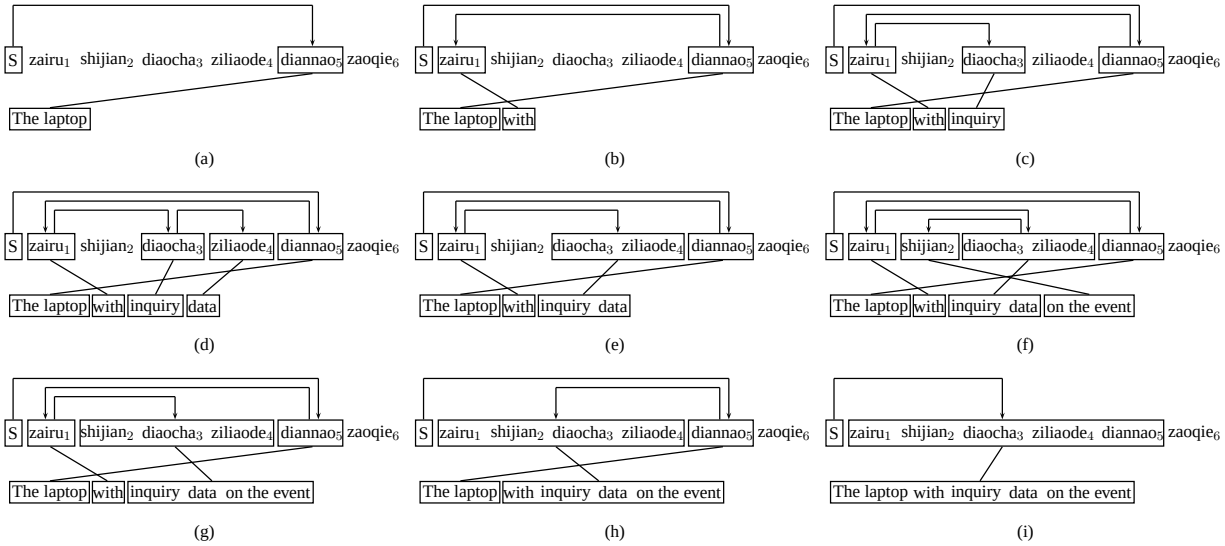


Figure 3: The partial translation procedure of the sentence in Figure 2.

sponding ITG tree, and source words covered by every node (eg. $A1, ..., A_{11}$ in Figure 2(b)) in the ITG tree can be seen as a block. By watching the tree in Figure 2, we can find that a block must be adjacent to the block either on its left or on its right, then they can be merged into a larger block. For example, A_2 matches the block $[zairu_1]$ and A_8 matches the block $[shijian_2\ diaocha_3\ ziliaode_4]$.[1] The two blocks are adjacent and they are merged into a larger block $[zairu_1\ shijian_2\ diaocha_3\ ziliaode_4]$, covered by A_9. The procedure of translating $zairu_1\ shijian_2\ diaocha_3\ ziliaode_4\ diannao_5$ is illustrated in Figure 3.

For a hypothesis during decoding, we assign it three factors: the current block, the left neighboring uncovered span and the right neighboring uncovered span. For example, in Figure 3(c), the current block is $[diaocha_3]$ and the left neighboring uncovered span is $[shijian_2]$ and the right neighboring uncovered span is $[ziliaode_4]$. $[zaoqie_6]$ is not thought of as the right neighboring block, for it is not adjacent to $[diaocha_3]$. The next covered block is $[ziliaode_4]$ (as shown in Figure 3(d)). For $[diaocha_3]$ and $[ziliaode_4]$ are adjacent, they are merged. In Figure 3(e), the current block is $[diaocha_3\ ziliaode_4]$.

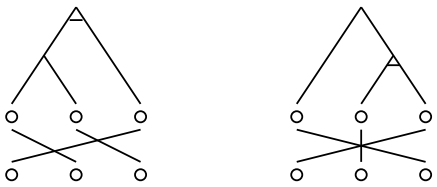
A sentence is translated with ITG constraints iff its source side can be covered by an ITG tree. That is to say, for every hypothesis during decoding, the next block to cover must be selected from the left or right neighboring uncovered span.

First, we show that if the next block to cover is selected in this way, the translation must observe ITG constraints. For every hypothesis during decoding, the immediate left and right words of the current block face the following three conditions:

(1) The immediately left word is not covered and the immediately right word is covered, then the next block to cover must be selected from the left neighboring uncovered span, eg. for the current block $[diaocha_3\ ziliaode_4]$ in Figure 3(e). In this condition, the ITG tree can be constructed in the following two ways: either all words in the left neighboring uncovered span are translated first, then this span is merged with the current span (taking three nodes as an example, this case is shown in Figure 4(a)), or the right part of the left neighboring uncovered span is merged with the current block first, then the new block is merged with the rest part of the left neighboring uncovered span (shown in Figure 4(b)). In a word, only after all words in the left neighboring uncovered span are covered, other words can be covered.

(2) The immediately right word is not covered and the immediately left word is covered. Similarly, only after all words in the right neighboring uncovered span are covered, other words can be

[1]The words within a block are sorted by their order in the source sentence.



(a) (b)

Figure 4: The two ways that the current block is merged with its left neighboring uncovered span. The third node in the first row denotes the current block, the first and second nodes in the first row denote left and right parts of the left neighboring uncovered span, respectively.

covered.

(3) The immediately left and right words are neither covered. The next block can be selected from either the left or the right neighboring uncovered span until the immediate left or right word is covered.

The above operations can be performed recursively until the whole source sentence is merged into a single block, so the reordering observes ITG constraints.

Now, we show that translation which is not generated in the above way must violate ITG constraints.

If the next block is selected out of the neighboring uncovered spans, the current block can be neither adjacent to the last covered block nor adjacent to the selected next block, so the current block can not be merged with any block and the whole sentence can not be covered by an ITG tree. As in Figure 3(b), if the next block to cover is $[zaoqie_6]$, then $[zairu_1]$ is neither adjacent to $[diannao_5]$ nor adjacent to $[zaoqie_6]$.

We can conclude that if we select the next block from the left or right neighboring uncovered span of the current block, then the translation must observe ITG constraints.

2.2 Shift-Reduce Decoding Algorithm

In order to generate the translation with ITG constraints, the shift-reduce algorithm have to keep trace of covered blocks, left and right neighboring uncovered spans. Formally, the shift-reduce decoding algorithm uses the following three stacks:

- S_t: the stack for covered blocks. The blocks are pushed in the order that they are covered, not the order that they are in the source sentence.

- S_l : the stack for the left uncovered spans of the current block. When a block is pushed into S_t, its corresponding left neighboring uncovered span is pushed into S_l.

- S_r :the stack for the right uncovered spans of the current block. When a block is pushed into S_t, its corresponding right neighboring uncovered span is pushed into S_r.

A translation configuration is a triple $c = \langle S_t,\ S_l,\ S_r \rangle$. Given a source sentence $f = f_1, f_2, ..., f_m$, we import a virtual start word and the whole translation procedure can be seen as a sequence of transitions from c_s to c_t, where $c_s = \langle [0],\ \emptyset,\ [1, m] \rangle$ is the initial configuration, $c_t = \langle [0, m],\ \emptyset,\ \emptyset \rangle$ is the terminal configuration. The configuration for Figure 3 (e) is $\langle [0][5][1][3, 4],\ [2],\ [6] \rangle$.

We define three types of transitions from a configuration to another . Assume the current configuration $c = \langle\ [f_{t_1 1}, f_{t_1 2}]...[f_{t_k 1}, f_{t_k 2}],\ [f_{l_1 1}, f_{l_1 2}]...[f_{l_u 1}, f_{l_u 2}],\ [f_{r_v 1}, f_{r_v 2}]...[f_{r 1 1}, f_{r 1 2}]\ \rangle$, then :

- Transitions *LShift* pop the top element $[f_{l_u 1}, f_{l_u 2}]$ from S_l and select a block $[i, j]$ from $[f_{l_u 1}, f_{l_u 2}]$ to translate. In addition, they push $[i, j]$ into S_t, and if $i \neq f_{l_u 1}$, they push $[f_{l_u 1}, i - 1]$ into S_l, and if $j \neq f_{l_u 2}$, they push $[j + 1, f_{l_u 2}]$ into S_r. The precondition to operate the transition is that S_l is not null and the top span of S_l is adjacent to the top block of S_t. Formally, the precondition is $f_{l_u 2} + 1 = f_{t_k 1}$.

- Transitions *RShift* pop the top element $[f_{r_v 1}, f_{r_v 2}]$ of S_r and select a block $[i, j]$ from $[f_{r_v 1}, f_{r_v 2}]$ to translate. In addition, they push $[i, j]$ into S_t, and if $i \neq f_{r_v 1}$, they push $[f_{r_v 1}, i-1]$ into S_l, and if $j \neq f_{r_v 2}$, they push $[j + 1, f_{r_v 2}]$ into S_r. The precondition is that S_r is not null and the top span of S_r is

adjacent to the top block of S_t. Formally, the precondition is $f_{t_k 2} + 1 = f_{r_v 1}$.

- Transitions *Reduce* pop the top two blocks $[f_{t_{k-1}1}, f_{t_{k-1}2}]$, $[f_{t_k 1}, f_{t_k 2}]$ from S_t and push the merged span $[f_{t_{k-1}1}, f_{t_k 2}]$ into S_t. The precondition is that the top two blocks are adjacent. Formally, the precondition is $f_{t_{k-1}2} + 1 = f_{t_k 1}$

The transition sequence of the example in Figure 2 is listed in Figure 5. For the purpose of efficiency, transitions *Reduce* are integrated with transitions *LShift* and *RShift* in practical implementation. Before transitions *LShift* and *RShift* push $[i, j]$ into S_t, they check whether $[i, j]$ is adjacent to the top block of S_t. If so, they change the top block into the merged block directly.

In practical implementation, in order to further restrict search space, distortion limit is applied besides ITG constraints: a source phrase can be covered next only when it is ITG-legal and its distortion does not exceed distortion limit. The distortion d is calculated by $d = |start_i - end_{i-1} - 1|$, where $start_i$ is the start position of the current phrase and end_{i-1} is the last position of the last translated phrase.

Transition	S_t	S_l	S_r
	$[0]$	$\emptyset$	$[1,6]$
RShift	$[0][5]$	$[1,4]$	$[6]$
LShift	$[0][5][1]$	$\emptyset$	$[2,4][6]$
RShift	$[0][5][1][3]$	$[2]$	$[4][6]$
RShift	$[0][5][1][3][4]$	$[2]$	$[6]$
Reduce	$[0][5][1][3,4]$	$[2]$	$[6]$
LShift	$[0][5][1][3,4][2]$	$\emptyset$	$[6]$
Reduce	$[0][5][1][2,4]$	$\emptyset$	$[6]$
Reduce	$[0][5][1,4]$	$\emptyset$	$[6]$
Reduce	$[0][1,5]$	$\emptyset$	$[6]$
Reduce	$[0,5]$	$\emptyset$	$[6]$
RShift	$[0,5][6]$	$\emptyset$	$\emptyset$
Reduce	$[0,6]$	$\emptyset$	$\emptyset$

Figure 5: Transition sequence for the example in Figure 2. The top nine transitions correspond to Figure 3 (a), ... , Figure 3 (i), respectively.

3 Related Work

Galley and Manning (2008) present a hierarchical phrase reordering model aimed at improving non-local reorderings. Via the hierarchical mergence of two blocks, the orientation of long distance words can be computed. Their shift-reduce algorithm does not import ITG constraints and admits the translation violating ITG constraints.

Zens et al. (2004) introduce a left-to-right decoding algorithm with ITG constraints on the alignment template system (Och et al., 1999). Their algorithm processes candidate source phrases one by one through the whole search space and checks if the candidate phrase complies with ITG constraints. Besides, their algorithm checks validity via cover vector and does not formalize ITG structure. The shift-reduce decoding algorithm holds ITG structure via three stacks. As a result, it can offer ITG-legal spans directly and decode faster. Furthermore, with the help of ITG structure, it can be extended to syntax-based models easily.

Xiong et al. (2006) propose a BTG-based model, which uses the context to determine the orientation of two adjacent spans. It employs the cube-time CYK algorithm.

4 Experiments

We compare the shift-reduce decoder with the state-of-the-art decoder Moses (Koehn et al., 2007). The shift-reduce decoder was implemented by modifying the normal search algorithm of Moses to our shift-reduce algorithm, without cube pruning (Huang and Chiang, 2005). We retained the features of Moses: four translation features, three lexical reordering features (straight, inverted and discontinuous), linear distortion, phrase penalty, word penalty and language model, without importing any new feature. The decoding configurations used by all the decoders, including beam size, phrase table limit and so on, were the same, so the performance was compared **fairly**.

First, we will show the performance of shift-reduce algorithm on three data sets with large training data sets (Section 4.1). Then, we will analyze the performance elaborately in terms of accuracy, speed and search ability with a smaller

training data set (Section 4.2). All experiments were done on Chinese-to-English translation tasks and all results are reported with case insensitive BLEU score. Statistical significance were computed using the sign-test described in Collins et al. (Collins et al., 2005).

4.1 Performance Evaluation

We did three experiments to compare the performance of the shift-reduce decoder, Moses and the decoder with ITG constraints using cover vector (denoted as CV). [2] The shift-reduce decoder decoded with two sets of parameters: one was tuned by itself (denoted as SR) and the other was tuned by Moses (denoted as SR-same), using MERT (Och, 2003). Two searching algorithms of Moses are considered: one is the normal search algorithm without cubing pruning (denoted as Moses), the other is the search algorithm with cube pruning (denoted as Moses-cb). For all the decoders, the distortion limit was set to 6, the nbest size was set to 100 and the phrase table limit was 50.

In the first experiment, the development set is part of NIST MT06 data set including 862 sentences, the test set is NIST MT08 data set and the training data set contains 5 million sentence pairs. We used a 5-gram language model which were trained on the Xinhua and AFP portion of the Gigaword corpus. The results are shown in Table 1(a).

In the second experiment, the development data set is NIST MT02 data set and the test set is NIST MT05 data set. Language model and the training data set are the same to that of the first experiment. The result is shown in Table 1(b).

In the third experiment, the development set is China Workshop on Machine Translation 2008 data set (denoted as CWMT08) and the test set is China Workshop on Machine Translation 2007 data set (denoted as CWMT07). The training set contains 2 Million sentence pairs and the language model are a 6-gram language model trained on the Reuter corpus and English corpus. Table 1(c) gives the results.

In the above three experiments, SR decoder improves the accuracy by 1.59, 0.62, 0.8 BLEU score ($p < .01$), respectively, and improves the speed by 15%, 24%, 30%, respectively. we can see that SR can improve both the accuracy and the speed while SR-same can increase the speed significantly with a slight improvement on the accuracy. As both SR and CV decode with ITG constraints, they match each other on the accu-

[2]The decoder CV is implemented by adding the ITG constraints to Moses using the algorithm described in (Zens et al., 2004).

	NIST06	NIST08	speed
Moses	30.24	25.08	4.827
Moses-cb	30.27	23.80	1.501
CV	30.35	26.23**	4.335
SR-same	——	25.09	3.856
SR	30.47	26.67**	4.126

(a)

	NIST02	NIST05	speed
Moses	35.68	35.80	7.142
Moses-cb	35.42	35.03	1.811
CV	35.45	36.56**	6.276
SR-same	——	35.84	5.008
SR	35.99*	36.42**	5.432

(b)

	CWMT08	CWMT07	speed
Moses	27.75	25.91	3.061
Moses-cb	27.82	25.16	0.548
CV	27.71	26.58**	2.331
SR-same	——	25.97	1.988
SR	28.14*	26.71**	2.106

(c)

Table 1: Performance comparison. Moses: Moses without cube pruning, Moses-cb: Moses with cube pruning, CV: the decoder using cover vector, SR-same: the shift-reduce decoder decoding with parameters tunes by Moses, SR: the shift-reduce decoder with parameters tuned by itself. The second column stands for develop set, the third column stands for test set and speed column shows the average time (seconds) of translating one sentence in the test set. **: significance at the .01 level.

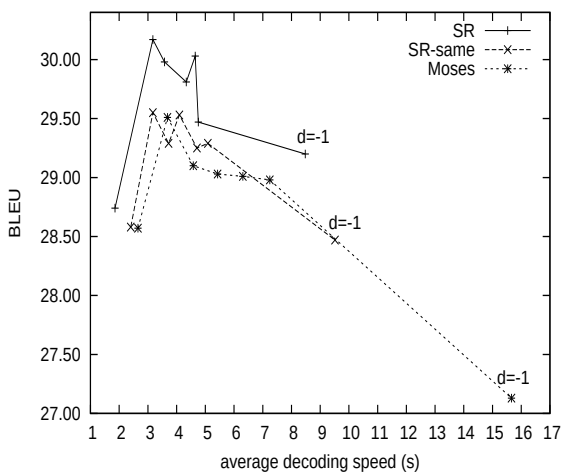


Figure 6: Performance comparison on NIST05. For a curve, the dots correspond to distortion limit $4, 6, 8, 10, 14$ and no distortion from left to right. $d = -1$ stands for no distortion limit.

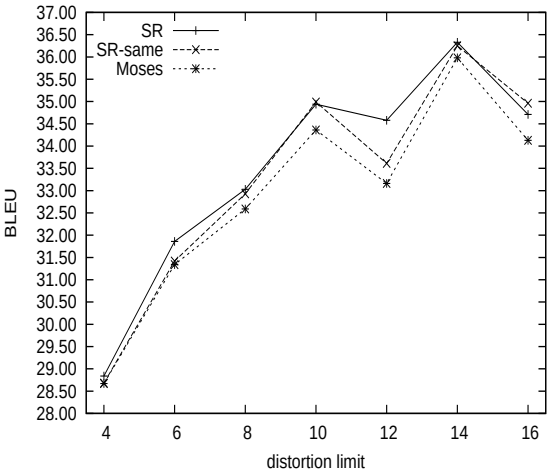
racy. However, the speed of SR is faster than CV. Cube pruning can improve decoding speed dramatically, but it is not risk-free pruning technology, so the BLEU score declines obviously.

4.2 Performance Analysis

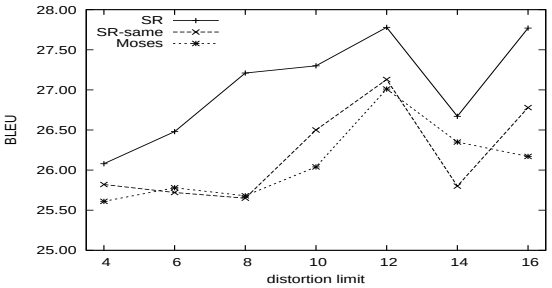
We make performance analysis with the same experiment configuration as the second experiment in Section 4.1, except that the training set in the analysis experiment is FBIS corpus, including 289k sentence pairs. In the following experiments, Moses employs the normal search algorithm without cube pruning.

For the decoders employ the linear distortion feature, the distortion limit will influence the translation accuracy. Besides, with different distortion limit, the proportion of ITG-legal translation generated by Moses will differ. The smaller the distortion limit is, the greater the proportion is. So we first compare the performance with different distortion limit.

We compare the shift-reduce decoder with Moses using different distortion limit. The results are shown in Figure 6. When distortion limit is set to 6, every decoder gets a peak value and SR has an improvement of 0.66 BLEU score over Moses. From the curves, we can see that the BLEU score of SR-same with distortion limit 8



(a) ITG set



(b) rest set

Figure 7: Accuracy comparison on the ITG set and rest set of NIST05. The ITG set includes the sentences the translations of which generated by Moses are ITG-legal, and the rest set contains the rest sentences. distortion limit $= 16$ denotes no distortion limit.

is lower than that of Mose with distortion limit 6. This is because the decoding speed of SR-same with distortion limit 8 is not faster than that of Moses with distortion limit 6. On the whole, compared to Moses, SR-same can improve the accuracy slightly with much faster decoding speed, and SR can obtain improvements on both the accuracy and the speed.

We split the test set into two sets: one contains the sentences, the translations of which generated by Moses are ITG-legal (denoted as ITG set) and the other contains the rest (denoted as rest set). From Figure 7, we can see that no matter on the ITG set or on the rest set, SR decoder can gain obvious accuracy improvements with all distortion

d	ITG						rest					
	Moses	SR-same	total	$<$	$=$	$>$	Moses	SR-same	total	$<$	$=$	$>$
4	28.67	28.68	1050	8	1042	0	25.61	25.82	32	0	0	32
6	31.34	31.42	758	51	705	2	25.78	25.72	324	32	2	290
8	32.59	32.93*	594	72	516	6	25.68	25.65	488	82	3	403
10	34.36	34.99**	456	80	365	11	26.04	26.50*	626	147	3	476
12	33.16	33.61**	454	63	380	11	27.01	27.13	628	165	1	462
14	35.98	36.25*	383	60	316	7	26.35	26.67*	699	203	1	495
-1	34.13	34.96**	351	39	308	4	26.17	26.78**	731	154	0	577

Table 2: Search ability comparison. The ITG set and the rest set of NIST05 were tested, respectively. On the ITG set, the following six factors are reported from left to right: BLEU score of Moses, BLEU score of SR-same, the number of sentences in the ITG set, the number of sentences the translation probabilities of which computed by Moses, compared to that computed by SR, is lower, equal and greater. The rest set goes similarly. *: significance at the .05 level, **: significance at the .01 level.

limit. While SR-same decoder only gets better results on the ITG set with all distortion limit. This may result from the use of the linear distortion feature. Moses may generate hypotheses the distortion of which is forbidden in the shift-reduce decoder. This especially sharpens on the rest set. So SR-same may suffer from an improper linear distortion parameter.

The search ability of Moses and the shift-reduce decoder are evaluated, too. The translation must be produced with the same set of parameters. In our experiments, we employed the parameters tuned by Moses. The test was done on the ITG and the rest set, respectively. The results are shown in Table 2. As the distortion limit becomes greater, the number of the ITG-legal translation generated by Moses becomes smaller. On the ITG set, translation probabilities from the shift-reduce decoder is either greater or equal to that from Moses on most sentences, and BLEU scores of shift-reduce decoder is greater than that of Moses with all distortion limit. Although the search space of shift-reduce decoder is smaller than that of Moses, shift-reduce decoder can give the translation that Moses can not reach. On the rest set, for most sentences, the translation probabilities from Moses is greater than that from shift-reduce decoder. But only when distortion limit is 6 and 8, the BLEU score of Moses is greater than that of the shift-reduce decoder. We may conclude that greater score does not certainly lead to greater BLEU score.

5 Conclusions and Future Work

In this paper, we present a shift-reduce decoding algorithm for phrase-based translation model that can generate the ITG-legal translation in linear time. The algorithm need not consider shift-reduce divergence and performs *reduce* operation as soon as possible. We compare the performance of the shift-reduce decoder with the state-of-the-art decoder Moses. Experiment results show that the shift-reduce algorithm can improve both the accuracy and the speed significantly on different test sets. We further analyze the performance and find that on the ITG set, the shift-reduce decoder is superior over Moses in terms of accuracy, speed and search ability, while on the rest set, it does not display advantage, suffering from improper parameters.

Next, we will extend the shift-reduce algorithm to syntax-based translation models, to see whether it works.

6 Acknowledgement

The authors were supported by National Natural Science Foundation of China Contract 60736014, National Natural Science Foundation of China Contract 60873167 and High Technology R&D Program Project No. 2006AA010108. We are grateful to the anonymous reviewers for their valuable comments.

References

Collins, Michael, Philipp Koehn, and Ivona Kucerova. 2005. Clause restructuring for statistical machine translation. In *Proc. of ACL*, pages 531–540.

Galley, Michel and Christopher D. Manning. 2008. A simple and effective hierarchical phrase reordering model. In *Proc. of EMNLP*, pages 848–856.

Huang, Liang and David Chiang. 2005. Better k-best parsing. In *Proceedings of the Ninth International Workshop on Parsing Technologies (IWPT)*, pages 53–64.

Knight, Kevin. 1999. Decoding complexity in word-replacement translation models. *Computational Linguistics*, 25:607–615.

Koehn, Philipp, Hieu Hoang, Alexandra Birch Mayne, Christopher Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proc. of the 45th ACL, Demonstration Session*.

Koehn, Philipp. 2004. Pharaoh: A beam search decoder for phrased-based statistical machine translation. In *Proc. of AMTA*, pages 115–124.

Och, Frans J., Christoph Tillmann, and Hermann Ney. 1999. Improved alignment models for statistical machine translation. In *Proc. of EMNLP*, pages 20–28.

Och, Frans J. 2003. Minimum error rate training in statistical machine translation. In *Proc. of ACL*, pages 160–167.

Tillmann, Chirstoph and Hermann Ney. 2003. Word reordering and a dynamic programming beam search algorithm for statistical machine translation. *Computational Linguistics*, 29:97–133.

Wu, Dekai. 1996. A polynomial-time algorithm for statistical machine translation. In *Proc. of ACL*, pages 152–158.

Wu, Dekai. 1997. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational Linguistics*, 23:377–403.

Xiong, Deyi, Qun Liu, and Shouxun Lin. 2006. Maximum entropy based phrase reordering model for statistical machine translation. In *Proc. of ACL*, pages 521–528.

Zens, Richard and Hermann Ney. 2003. A comparative study on reordering constraints in statistical machine translation. In *Proc. of ACL*, pages 144–151.

Zens, Richard, Hermann Ney, Taro Watanable, and Eiichiro Sumita. 2004. Reordering constraints for phrase-based statistical machine translation. In *Proc. of COLING*, pages 205–211.

A Novel Method for Bilingual Web Page Acquisition from Search Engine Web Records

Yanhui Feng, Yu Hong, Zhenxiang Yan, Jianmin Yao, Qiaoming Zhu
School of Computer Science & Technology, Soochow University
{20094227002, hongy, 20074227065071, jyao, qmzhu}@suda.edu.cn

Abstract

A new approach has been developed for acquiring bilingual web pages from the result pages of search engines, which is composed of two challenging tasks. The first task is to detect web records embedded in the result pages automatically via a clustering method of a sample page. Identifying these useful records through the clustering method allows the generation of highly effective features for the next task which is high-quality bilingual web page acquisition. The task of high-quality bilingual web page acquisition is a classification problem. One advantage of our approach is that it is search engine and domain independent. The test is based on 2516 records extracted from six search engines automatically and annotated manually, which gets a high precision of 81.3% and a recall of 94.93%. The experimental results indicate that our approach is very effective.

1 Introduction

There have been extensive studies on parallel resource extraction from parallel monolingual web pages of some bilingual web sites (Chen and Nie, 2000; Resnik and Smith, 2003; Zhang et al., 2006; Shi et al., 2006). Candidate parallel web pages are acquired by making use of URL strings or HTML tags, then the translation equivalence of the candidate pairs are verified via content-based features.

However, we observe that bilingual resources may exist not only in two parallel monolingual web pages, but also in single bilingual web pages. For example, many news web pages and English learning pages are bilingual. Based on this observation, researchers have proposed methods to improve parallel sentences extraction within a bilingual web page. Jiang (2009) uses an adaptive pattern-based method to mine interesting bilingual data based on the observation that bilingual data usually appears collectively following similar patterns. Because the World Wide Web is composed of billions of pages, it is a challenging task to locate valuable bilingual pages.

To acquire bilingual web pages automatically, a novel and effective method is proposed in this paper by making use of search engines, such as Baidu (http://www.baidu.com). By submitting parallel sentence pairs to the given search engine, lots of result pages with web records are returned, most of which are linked to bilingual web pages. We first identify and extract all result records automatically by selecting and analyzing a sample page with a clustering method, and then select high-quality bilingual web pages from candidates with classification algorithms.

Our method has the following advantages:

1. Former researchers extract parallel corpus from specific bilingual web sites. Since search engines index amounts of web pages, and we aim to acquire bilingual pages based on them, our method expands the corpus source greatly.

2. For one search engine, only one sample result page is used to generate the record wrapper. Then the wrapper is used to identify web records from other result pages of the same search engine. Compared with existing data record extraction technologies, such as MDR (Liu et al., 2003; Zhai and Liu, 2006), our method is more effective and efficient.

Coling 2008: Poster Volume, pages 294–302,
Beijing, August 2010

3. We model the issue of verification bilingual pages as a binary-class classification problem. The records acquired automatically and annotated manually are utilized to train and test the classifier. This work is domain and search engine independent. That is to say, the records acquired from any search engine in any domain are used indiscriminately as training and testing dataset.

The rest of the paper is organized as follows. Related works are introduced in section 2. Section 3 provides an overview of our solution. The work about bilingual page acquisition and verification is introduced in section 4 and 5. Section 6 presents the experiments and results. Finally section 7 concludes the paper.

2 Related Work

As far as we know, there is no publication available on acquiring bilingual web pages. Most existing studies, such as Nie (1999), Resnik and Smith (2003) and Shi (2006), mine parallel web documents within bilingual web sites first and then extract bilingual sentences from mined parallel documents using sentence alignment method.

In this paper, the candidate bilingual web pages are acquired by analyzing web records embedded in the search engines' result pages. Therefore, record extraction from result pages is a critical technique in our method. Many researches, such as Laender (2002), have been developed various solutions in web information extraction from kinds of perspectives.

Earlier web information extraction systems (Baumgartner et al., 2001; Liu et al., 2000; Zhai and Liu, 2005) require users to provide labeled data so that the extraction rules could be learned. Yet such semi-automatic methods are not scalable enough to the whole Web which changes at any time. That's why more and more researchers focus on fully or nearly fully automatic solutions.

Structured data objects are normally database records retrieved from underlying web databases and displayed on the web pages with some fixed templates, so automatic extraction methods try to find such patterns and use them to extract more data. Several approaches have succeeded to address the problem automatically without human assistance. IEPAD (Chang and Lui, 2001) identifies sub-strings that appear many times in a document. By traversing the DOM tree of the Web page, MDR extracts the data-rich sub-tree indirectly by detecting the existence of multiple similar generalized-nodes. The key limitation is its greedy manner of identifying a data region. DEPTA (Zhai and Liu, 2005) uses visual information (locations on the screen at which the tags are rendered) to infer the structural relationship among tags and to construct a tag tree. NET (Liu and Zhai, 2005) extracts flat or nested data records by post-order or pre-order traversal of the tag tree. ViNTs (Zhao et al., 2005) considers the web page as a tag tree, and utilizes both visual content features as well as tag tree structures. It assumes that data records are located in a minimum data-rich sub-tree and separated by separators of tag forests. Zhao (2006) explicitly aims at extracting all dynamic sections from web pages, and extracting records in each section, whereas ViNTs focuses on record extraction from a single section. Miao (2009) figures out how tag paths format the whole page. Compared with the previous method, it compares pairs of tag path occurrence patterns to estimate how likely these tag paths represent the same list of objects instead of comparing one pair of individual sub-trees in the record. It brings some noise. We follow this method and make appropriate improvement for our task.

3 Basic Concepts and Overview

3.1 Basic Concepts

Some basic concepts are introduced below.

Figure 1. An example of search engine return

Tag Path: The path of a tag consists of all nodes from the tree root <html> to itself. We use tag path to specify the location of the tag. The tag paths are classified into two types: text tag paths and non-text tag paths.

Data Record: When a page is considered as strings of tokens, data records are enwrapped by one or more tag paths, which compose the visually repeating pattern in a page. This paper aims to extract such structured data records that are produced by computer programs following some fixed templates, while whose contents are usually retrieved from backend databases. For example, there are four records in Figure 1.

3.2 Method Overview

We can get much more bilingual web pages by submitting parallel sentence pairs to the search engine than submitting monolingual queries. Based on this observation, our work is as shown in Figure 2. The algorithm consists of two steps: 1) Record wrapper generation. By submitting parallel sentence pairs to search engines, result pages containing lots of web records are returned. In order to generate record wrappers, we select and analyze a sample page and then apply clustering method to tag paths with similar patterns. We apply these wrappers to extract more records, which are linked to candidate bilingual web pages. 2) High-quality bilingual page acquisition. In order to acquire high-quality bilingual pages from candidates, a binary classifier is constructed to decide whether the candidate pages are bilingual or not. In order to improve the classifier, some useful resources are used, such as a dictionary and translation equivalents.

However, a result page often contains some information irrelevant to the query, such as information related to the hosting site of the search engine, which increases the difficulty of record extraction. Besides, there are also many irrelevant records irrelevant to the query. So our focus is to acquire plenty of features to filter out the irrelevant pages from the candidates.

In this paper, the first result page is chosen as the sample page and Affinity Propagation (AP) clustering is used. The reason lies in Frey and Dueck (2007), which proves that to produce the groups of tag paths; the AP algorithm does not require the three restrictions: 1) the samples must be of a specific kind, 2) the similarity values must be in a specific range, and 3) the similarity matrix must be symmetric. In order to decide the type of a page, the Support Vector Machines (SVM) (Cortes and Vapnik, 1995) classifier on Fuzzy C-means is constructed combining with word-overlap, length and frequency measures. SVM is well-fitted to treat such classification problems that involve interrelated features likes ours, while most probabilistic classifiers, such as Naïve Bayes classifier, strongly assume feature independence (DuVerle and Prendinger, 2009).

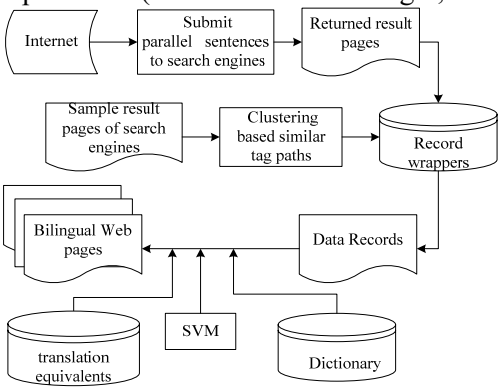


Figure 2. Overview of the method

4 Bilingual Page Acquisition

4.1 Result Page Extraction

The result pages of a search engine consist of a ranked list of document summaries linked to the actual documents or web pages. A web document summary typically contains the title and URL of the web page, links to live and cached versions of the page and, most importantly, a short text summary, or a snippet, to convey the contents of the page. Such snippets embedded in result pages of search engines are query-dependent summaries. White (2001) finds the result pages are sensitive to the content and language of the query. If the query is monolingual, the returned search results are mostly monolingual, while the result pages are bilingual if the query is bilingual. In order to acquire more bilingual web pages, we submit parallel translation pairs. Figure 1 gives an example result page from Baidu, in which the snapshot consists of four records related to the query, which consists of "I see." and its translation "我明白了。". The results have

more effective advantages than submitting the query "I see." or "我明白了。" respectively.

4.2 Clustering With Path Similarity

Given a web page, we get the occurrence positions of each tag path the same as the sequence in the preorder traversal of the page's DOM tree. Certainly, there are many tag paths which appear several times in the whole page. So an inverted mapping from HTML tag paths to their positions is built easily. For example, there are 599 tag paths formatting the sample page in Figure 1, and after the inverted mapping, we acquire 86 unique tag paths in all. Only tick off one part of the results as shown in Table 1, where P_i represents the *ith* unique tag path, and the vector S_i is defined to store the occurrence positions of P_i in the third column.

As introduced above, detecting visually repeating tag paths is a clustering problem. Above all, a factor in determining the clustering performance is the choice of similarity functions, which captures the likelihood that two data samples belong to the same cluster. In our case, the similarity scores between two tag paths aim to capture how their positions are close to each other and how they interleave each other.

With the purpose of characterizing how close two tag paths appear, we only acquire the distance between paths' average positions, which is easy to obtain by the acquired occurrence vectors. For example, the average position of P_{11} and P_{15} in Table 1 is 227 and 215, so the distance between them is 12.

i	Unique Tag Path (P_i)	Occurrences (S_i) of P_i
1	\html	1
3	\html\head\#text	3,4,7,8,9
9	\html\body\table	84,93,115,146,180, 217,258,292,335,372, 406,437
11	\html\body\table\tr	15,85,94,116,147,181, 218,259,293,336,373, 407,438
14	\html\body\table\tr \td\#text	18,21,24,27,55,79,87, 91,97,111,113
15	\html\body\table\tr \td\a	19,88,118,149,183, 220,261,295,338,375, 409,440

Table 1. Unique tag paths of the sample page

However, the most difficult problem is how to capture the interleaving characteristic between two tag paths. Before doing that, another vector O_i is produced. $O_i(k)$ indicates whether the tag path P_i occurs in the position k or not by its value. In addition, the value is binary that 0 or 1, and 0 shows P_i doesn't occur in the position k, while 1 shows the opposite. Of particular note, the length of each O_i is equal to the total number of HTML tags that formatting the whole web page. Take the tag path P_3 ("\html\head\#text") in Table 1 as an example, whose position vector O_3 is (0, 0, 1, 1, 0, 0, 1, 1, 1, 0… 0), and the vector's length is 599, because there are totally 599 tag paths formatting the sample page in Figure 1.

Based on the position vectors, we capture how tag path P_i and P_j interleave each other by a segment D_{O_i/O_j} of O_i divided by O_j. We aim to find such tag paths that divide each other in average. In other words if the variance of counts in the segment D_{O_i/O_j} is stable, they are likely to be grouped in the same cluster. So, we define the interleaving measure μ in terms of the variances of D_{O_i/O_j} and D_{O_j/O_i} as:

$$\mu(O_i,O_j)=\max\{Var(D_{O_i/O_j}),Var(D_{O_j/O_i})\} \quad (1)$$

where D_{O_i/O_j} is acquired by O_j as follows: if value of $O_j(k)$ is 1, $O_i(k)$ is a separator to segment itself into several regions. The value of every element in the segment is the count of P_i that occurs in every region, which is the number of 1 in the region.

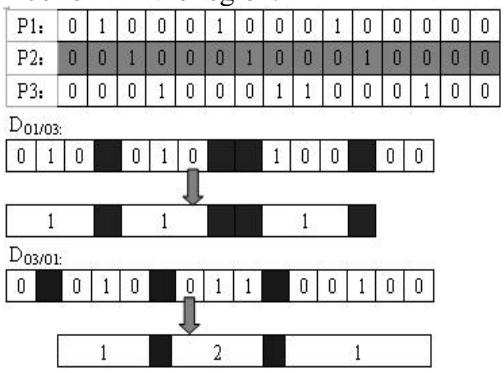


Figure 3. An Example of tag paths

In addition, there may be many consecutive separators in O_i, and we integrate them into one. Besides, the segment is a non-empty set. So if there is no occurrence of P_i in one region, we

will ignore this special region. Figure 3 shows three tag paths. P_1 and P_2 are likely to belong to the same cluster because of their regular occurrences, whereas the occurrences of P_3 are comparatively irregular. By our method, D_{O_1/O_3} = {1, 1, 1} and D_{O_3/O_1} = {1, 2, 1}. We integrate separators once and ignore an empty region in the process of getting D_{O_1/O_3}.

Both the score of the closeness measure and the interleaving measure for any two tag paths are non-negative real numbers. And a smaller value of either measure indicates a high frequency that the two tag paths appear regularly. The measure $\sigma(P_i, P_j)$ defined below is inversely proportional of these two measures.

$$\sigma(P_i, P_j) = \frac{\varepsilon}{c(S_i, S_j) \times \mu(O_i, O_j) + \varepsilon} \quad (2)$$

where ε is a non-negative term that avoids dividing by 0 and normalizes the similarity value so that it falls into the range (0, 1]. In our experiment, we choose ε = 10. By Equation 2, we calculate the similarity value of any pair of tag paths. As expected, the pairwise similarity matrix is fed into the AP clustering algorithm directly, and each cluster acquired from AP clustering contains *n* tag paths, which indicates that those *n* paths appear repeatedly together with high frequency, and the tag paths that have no remarkable relation are spilt into different clusters. For the given sample page in Figure 1, the number of identified clusters is 16.

We observe that HTML code of most data records contain more than three HTML tags, so we only examine the clusters containing four or more visual signals. In the clustering result of sample page in Figure 1, there are three clusters' sizes less than four. Meanwhile, we also note that:

1. The feature page of a common search engine usually contains 10 or more web records with similar layout pattern. So we define a threshold T=3. If an ancestor tag path doesn't occur more than T times, we believe these tag path dose not lead a record.
2. Usually the content of the result pages returned by search engines is completely related to the queries, which means the data records that we are interested in are distributed in the whole page as main component. So the occurrence position of valuable tag paths must be global optimization. In this paper, the scope between beginning and ending occurrence must be wider than three quarters of the length of the web page.

Thus, we get essential clusters fit with above observations, which is denoted by C= {C_1, C_2…C_M}. Once we have the essential clusters, we apply them in new web page of the same search engine to identify data records.

4.3 Data Record Extraction

Based on the essential clusters, we extract the exact data records from the real content of text tag path that follow the ancestor tag path.

In order to describe the extraction process in details, we firstly define D_{aI} as the child tag paths of an ancestor tag path P_a, and suppose that (Pos_1… Pos_i… Pos_m) is the occurrence vector of P_a, which means at each position Pos_i the tag path P_i occurs. $D_{a(i)}$ is such a tag path set that the position Pos of every path in it is $Pos_i < Pos < Pos_{i+1}$. In the meantime, such path strings must begin with the same prefix of P_a. Such as in Table 2, $D_{a\ (i)}$ contains tag paths from Pos_i to Pos_{i+1}-1, and we obtain the *ith* records embedded in the result pages by acquiring the real content of all text tag paths in $D_{a(i)}$.

Occurrence of P_a	D_{aI} of P_a	Child tag path
Pos_1		P_a:\html\body\table\tr
Pos_1+1	$D_{a(1)}$	P_t:\html\body\table\tr\…
……		……
Pos_2-1		P_k: ……
⋮	⋮	⋮
Pos_i		P_a:\html\body\table\tr
Pos_{i+1}	$D_{a(i)}$	P_t:\html\body\table\tr\…
……		……
Pos_{i+1}-1		P_n: ……
⋮	⋮	⋮
Pos_m	$D_{a(m)}$	P_a:\html\body\table\tr
……		……

Table 2. Collection of child tag paths for ancestor tag path

5 Bilingual Web Page Verification

Based on the previous work, we capture a list of records based on a holistic analysis of a result page, and each record contains snippets and URLs related to the query. In this section, we aim to decide whether the candidate pages that returned records are linked to are bilingual or not by putting some statistical features (collected from snippets) into an effective SVM classification.

To the acquired snippets, some necessary preprocessing is made before we acquire useful features. We remove most of the noise that affect the precision and robustness of the entire system by such methods as recovery of abbreviation words, deletion of noisy words, amendment for half or full punctuations and simplified or traditional characters, and so on.

The snippet is described with more regular contents after preprocessing. We cut the snippet into several segments by its language. Each segment of the snippet is just represented in one language, which is either English or Chinese in this paper and different from its adjacent segments. So the source snippets are transferred into such language strings that consist of C and E, where C stands for Chinese and E stands for English. It is unlikely that continuous C or E exists in the same language string. We store the real text T_c (T_e) that each C (E) stands for. We take the snippet "I see. 我明白了。I quit! 我不干了！" as example, its language string is "ECEC" and real text string is $T_eT_cT_eT_c$, where the two T_e stand for "I see" and "I quit", the two T_c stand for "我明白了" and "我不干了".

Note that different feature functions for the classifier will lead to different results, it is important to choose feature functions that will help to discriminate different classes. In this paper, the SVM classifier involves word-overlap, length and frequency features. We define these three features based on the snippet itself as follows:

(1) Word-Overlap measure

Word overlap judges the similarity of Chinese term and English term. In this paper, we acquire the word-overlap score between any two adjacent language segments. The similarity *Score(c_res,e_res)* of Chinese term and English term is based on word-overlap as following:

$$Score(c\_res, e\_res) = \frac{\sum_{i=1}^{p} Max_{1\le j\le q}(Sim(c_i, e_j))}{\phi} \quad (3)$$

where the denominator is normalization factor, and in our experiment we select $p+q$ as its value, where p stands for the length of Chinese term and q stands for the length of English term. In addition, c_i stands for the *ith* word of Chinese term and e_j stands for the *jth* word of English term. $Sim(c_i,e_j)$ in Liu (2003) and Deng (2004) stands for the similarity of Chinese word c_i and English word e_j.

In our experiment, the Chinese and English sub-snippets are equivalent to Chinese and English sentences of the bilingual pages. In the segmented snippet, with regard to each sub-snippet *T*, which is at even position in the language string, we separately evaluate the intermediate score for snippet *T* with its left and right neighbors by Equation 3. Especially when *T* doesn't have right or left neighbor, the score for *T* with its null neighbor is 0. So for every sub-snippet that needs to be scored the word-overlap score, there are two candidate scores with its adjacent neighbors. Then we choose the higher value as one item of an intermediate result vector. Either the length of the language string is $2 \times n$ or $2 \times n+1$, the length of intermediate vector is n, and the final score is computed as follows:

$$Score(s) = \frac{\sum_{k=1}^{n} InV_k}{n \times m} \quad (4)$$

where *Score(s)* stands for the final score of snippet s on the word-overlap measure, and vector *InV* is the intermediate result vector as mentioned before. The length of the vector *InV* is n, and m is the number of its items that is not equal to zero. *m/n* is used as a useful measure of length, because it indicates how many parallel pairs are there in the same snippet.

(2) Length-Based measure

We acquire three scores about length measure. Take the language string "ECECEC" as example, we use "$E_1C_1E_2C_2E_3C_3$" to replace it for simple description. We acquire one score of the length measure as follows:

$$Score\ (s) = \frac{\sum_{i=1}^{m} (Len\ (c) + Len\ (e))}{Len\ (s)} \quad (5)$$

where *s* and *m* stand for the same as in Equation 4. In addition, *c* and *e* stands for such sub-snippet that *Score(c,e)* contributes to $\sum_{k=1}^{n} InV_k$. The function *Len(s)* is to compute the number of words in the sentence.

We acquire the length of language string. If the length is too long or too short, the associated web page is unlikely to be a bilingual page. At the same time, we are not interested in some language strings although the lengths of them are appropriate. So we also store the variances of lengths about each sub-snippet.

(3) Frequency-Based measure

According to the result pages, queries often occur in the title, snippet, or advertisements. They are highlighted to make them easier to identify. Hence we aim to acquire the frequency of the query in one whole snippet as a feature.

Based on the three measures above, a number of records (containing snippets and URLs) for training and testing can be converted them into a 6-dimensional feature space. In our experiments, nonlinear SVM with Gaussian Radial Basis Function (RBF) kernel is used. The performance of the SVM classifier indicates that it is a reliable way to verify whether the page is bilingual or not by the content of snippet.

6 Experiments and Results

6.1 The Data Set

To acquire enough experimental data, we collect from Google, Baidu, Yahoo, Youdao, Bing and Tecent Soso, and the effectiveness of our algorithm is evaluated based on the data set from these six search engines.

Result records of search engines are collected by program and by human beings with submitting different queries respectively. They are used for checking the performance of record extraction. When evaluating the method of verification bilingual web pages, 2300 records (60% are positive instances) are chosen for training the SVM classifier, and other 230 are selected randomly as test records from the whole record set.

The training data is annotated by human in two methods. The first method is motivated by the content of each source snippet. The annotators assign the type of web pages by scanning the text of every snippet. If the snippet contains many parallel term pairs, we annotate the page as bilingual or monolingual if not parallel. We also use another annotation method, which is to reach the URL by the Internet Explorer. By checking the content of the real web page, annotators decide the type of the candidate pages. And the biggest difference between the two public hand-classified dataset appears when some snippets of candidate pages have no clues in their content to predict classifications.

6.2 Evaluation On Bilingual Page Acquisition

The entire system is evaluated by measuring the performance of the binary SVM classifier. And how the classifier performance changes with three features is shown in Table 3, where W, L and F separately stand for the word-overlap, length and frequency measures.

In order to improve the performance of word-overlap measure, we use not only the bilingual dictionary but also translation equivalents, which are extracted from parallel corpora. Because the bilingual dictionary doesn't contain all necessary entries, the classifier with only word-overlap measure accepts many wrong pairs.

Feature	W	W +L	W +L+F
Precision	70.2%	81.02%	85.10%

Table 3. SVM Classifier Performance changes with more features added to the classifier

Table 3 shows that the length feature and the frequency feature have a significant effect on bilingual web page verification because of the natural relationship among queries, snippets and true web pages.

N	#1		#2	
	P(%)	R(%)	P(%)	R(%)
1	85.1	92.3	75%	84.8
2	80.7	95.1	72.8	85.7
3	78.1	97.4	71.0	93.0
aver	81.3	94.93	72.93	87.83

Table 4. Performance versus training data types

Three experiments of verification bilingual web pages based on two different training datasets are conducted whose results are shown in Table 4. #1 stands for the data set annotated by snippets, and #2 stands for the training data annotated by URLs. Precision and recall are used to evaluate our method. The average precision based on training dataset #2 is 73%, which is lower than the precision of 81.3% resulting from the dataset #1, because in many cases, some snippets are weakly related with real text in the real pages introduced by search engine summarization algorithm. From the table, we also see that the recalls in dataset #1 and #2 are both relatively high, which means our classifier can select high-quality bilingual pages with high accuracy.

6.3 Evaluation On Web Record Extraction

Record extraction has significant effect on bilingual web page collection. A useful intermediate evaluation of the whole scheme is conducted by measuring the performance of record extraction.

We built a prototype system to test the algorithm of record extraction based on the clustering of similar records. On a laptop with a Pentium M 1.7G processor, the process of constructing records wrapper for a given search engine is done in 10 to 30 seconds. Once the wrapper is built, the record extraction from a new result page is done in a small fraction of a second.

In order to test the robustness of the generated wrapper, we compare the records extracted by our method with the test records acquired manually. The precision and recall measures are used to evaluate the result. 98% of all the records are extracted by program, with a precision of 99%. The precision indicates that the generated wrappers in our experiment are quite robust to acquire records. The recall is lower than the precision, which indicates that it sometimes misses a few records. The reason for this is that in the extraction step, the records different from more common ones are eliminated.

We compare our performance with the work in Zhao (2006), which addresses the issue of differentiating dynamic sections and records based on the sample result pages. It generates section wrappers by identifying section boundary markers in nine steps. It is more complicated in computation than ours because it renders each result page and extracts its content lines by a traversal of the DOM tree, while we use tag structure of a page. The accordance is making full use of the sample pages for given search engines. The method also gets a high precision of 98.8% and a recall of 98.7%.

7 Conclusion

The paper presents a novel method to acquire bilingual web pages automatically via search engines. In order to improve the efficiency and effectiveness, the snippets of search engines rather than the contents of the massive pages are analyzed to locate bilingual pages. Bilingual web page verification is modeled as a classification problem with word-overlap, length and frequency measures. Based on the similarity of HTML structures, AP clustering is used to extract web records from result pages of search engines. Experiments show that our algorithm has good performance in precision and recall.

As a valuable resource for up-to-date bilingual terms and sentences, bilingual web pages are counterpart to parallel monolingual web pages. Our method brings an efficient and effective solution to bilingual language engineering.

References

Adelberg B., NoDoSE. 1998. A tool for semi-Automatically extracting structured and sem-istructured data from text documents. In: *Proc.ACM SIGMOD Conference on man-agement of Data*, Seattle, WA (1998).

Baumgartner R., S. Flesca and G. Gottlob.2001. Visual Web Information Extraction with Lixto. *Proceedings of the 27th International Conference on Very Large Data Bases*, pp.119-128, September 11-14, 2001

Chang C., S. Lui. 2001. Information Extraction based on Pattern Discovery. In *Proceedings of the 10th international conference on World Wide Web*. pp.681-688, May 01-05, 2001, Hong Kong.

Chen Jiang and Jian-Yun Nie. 2000. Web

Parallel text mining for Chinese-English cross-language information retrieval. *Proceedings of RIAO2000 Content-Based Multimedia Information Access, CID, Paris*

Cortes, C. and V. Vapnik. 1995. Support-vector network. *Machine Learning 20*, pp.273-297.

Deng Dan. 2004. Research on Chinese-English word alignment. *Institute of Computing Technology Chinese Academy of Sciences*, Master Thesis. (in Chinese).

DuVerle David, Helmut Prendinger. 2009. A Novel Discourse Parser Based on Support Vector Machine Classification. *The 47th Annual Meeting of the Association for Computational Linguistics*. pp. 665-673

Frey B. J. and D. Dueck. 2007. Clustering by passing messages between data points. *Science*, 315(5814):972-976.

Laender A, B. Ribeiro-Neto, A. da Silva, J. Teixeira. 2002. A Brief Survey of Web Data Extraction Tools. *ACM SIGMOD Record.* Volume 31, Number 2.

Liu B. and Y. Zhai. 2005. System for extracting Web data from flat and nested data records. In *Proceedings of the Conference on Web Information Systems Engineering*, pp.487-495.

Liu B., R. Grossman and Y. Zhai. 2003. Mining Data Records in Web Pages. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge Discovery and Data mining*, Washington, D.C, pp.601-606.

Liu Feifan, Jun Zhao, Bo Xu. 2003. Building Large-Scale Domain Independent Chinese-English Bilingual Corpus and the Researches on Sentence Alignment. *Joint Symposium on Computational Linguistics*.

Liu L., C. Pu and W. Han. 2000. An XML-Enabled Wrapper Construction System for Web Information Sources. *Proceedings of the 16th International Conference on Data Engineering*, pp.611.

Long Jiang, Shiquan Yang, Ming Zhou, Xiaohua Liu and Qingsheng Zhou. 2009. Mining Bilingual Data from the Web with Adaptively Learnt Patterns. *The 47th Annual Meeting of the Association for Computational Linguistics*. pp. 870-878 (2009)

Miao Gengxin, Junichi Tatemura, Wang-Pin Hsiung, Arsany Sawires, Louise E. Moser. 2009. Extracting data records from the web using tag path clustering. In *Proceedings of the 18th International Conference on World Wide Web*, Spain, Madrid.

Nie Jian-Yun, Michel Simard, Pierre Isabelle, Richard Durand 1999. Cross-Language Information Retrieval based on Parallel Texts and Automatic Mining of Parallel Texts in the Web. *SIGIR-1999*; 74-81.

Resnik Philip and Noah A. Smith. 2003. The web as a Parallel Corpus. *Computational Linguistics*.

Shi Lei, Cheng Niu, Ming Zhou, and Jianfeng Gao. 2006. A DOM Tree Alignment Model for Mining Parallel Data from the Web. In *Joint Proceedings of the Association for Computational Linguistics and the International Conference on Computational Linguistics*, Sydney, Australia.

White, R., Jose, J. & Ruthven, R. 2001.Query-biased web page summarisation: a task-oriented evaluation. In Proceedings *of the 24th ACM SIGIR Conference on Research and Development of Information Retrieval*. New Orleans, Louisiana, United States, pp. 412-413.

Zhai Y., B. Liu. 2005. Extracting Web Data Using Instance-Based Learning. *Web Information Systems Engineering*.

Zhai Y., B. Liu. 2005. Web Data Extraction Based on Partial Tree Alignment. In *Proceedings of the 14th international conference on World Wide Web.* May 10-14, 2005, Chiba, Japan.

Zhang Ying, Ke Wu, Jianfeng Gao, Phil Vines. 2006. Automatic Acquisition of Chinese-English Parallel Corpus from the web. In *Proceedings of 28th European Conference on Information Retrieval*.

Zhao H., W. Meng, Z. Wu, V. Raghavan, C. Yu. 2006. Automatic Extraction of Dynamic Record Sections From Search Engine Result Pages. In *Proceedings of the 32nd International conference on Very large databases*.

Building Systematic Reviews Using Automatic Text Classification Techniques

Oana Frunza, **Diana Inkpen**, and **Stan Matwin**
School of Information Technology and Engineering
University of Ottawa
{ofrunza,diana,stan}@site.uottawa.ca

Abstract

The amount of information in medical publications continues to increase at a tremendous rate. Systematic reviews help to process this growing body of information. They are fundamental tools for evidence-based medicine. In this paper, we show that automatic text classification can be useful in building systematic reviews for medical topics to speed up the reviewing process. We propose a per-question classification method that uses an ensemble of classifiers that exploit the particular protocol of a systematic review. We also show that when integrating the classifier in the human workflow of building a review the per-question method is superior to the global method. We test several evaluation measures on a real dataset.

1 Introduction

Systematic reviews are the result of a tedious process which involves human reviewers to manually screen references of papers to determine their relevance to the review. This process often entails reading thousands or even tens of thousands of abstracts from prospective articles. As the body of available articles continues to grow, this process is becoming increasingly difficult.

Common systematic review practices stipulate that two reviewers are used at the screening phases of a systematic review to review each abstract of the documents retrieved after a simple query-based search. After a final decision is made for each abstract (the two reviewers decide if the abstract is relevant or not to the topic of review), in the next phase further analysis (more strict screening steps) on the entire article is done. A systematic review has to be complete, articles that are published on a certain topic and are clinically relevant need to be part of the review. This requires near-perfect recall since the accidental exclusion of a potentially relevant abstract can have a significantly negative impact on the validity of the overall systematic review (Cohen *et al*., 2006). Our goal in this paper is to propose an automatic system that can help human judges in the process of triaging articles by looking only at abstracts and not the entire documents. This decision step is known as the initial screening phase in the protocol of building systematic reviews, only the abstracts are used as source of information.

One reviewer will still read the entire collection of abstracts while the other will benefit from the help of the system; this reviewer will have to label only the articles that will be used to train the classifier (ideally a small proportion for workload reduction), the rest of the articles will be labeled by the classifier.

In the systematic review preparation, if at least one reviewer agrees to include an abstract, the abstract will have the labeled included and it will pass to the next screening phase; otherwise, it will be discarded. Therefore, the benefit of doubt plays an important role in the decision process. When we replace one reviewer with the automatic classifier, because we keep one human judge in the process, the confidence and reliability of the review is still higher while the overall workload is reduced. The reduction is from the time required for two passes through the collection (for the two humans) to only one pass and the smaller part labeled by the reviewer which is assisted by the classifier. Figure 1 presents on overview of our proposed workflow.

Coling 2008: Poster Volume, pages 303–311,
Beijing, August 2010

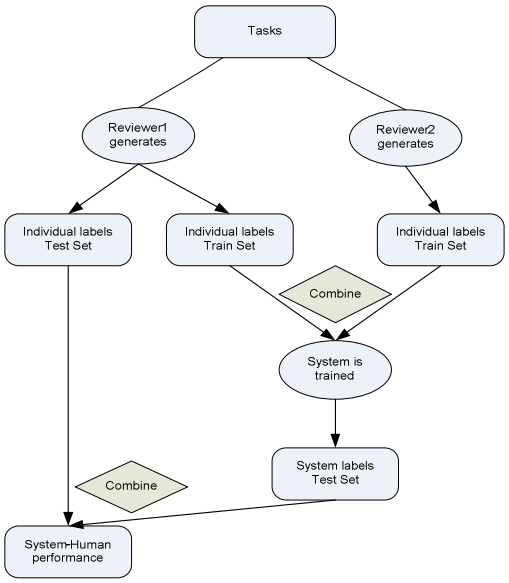


Figure 1. Embedding automatic text classification in the process of building a systematic review.

The task that needs to be solved in order to help the systematic review process is a text classification task intended to classify an abstract as relevant or not relevant to the topic of review.

The hypothesis that guides our research is that it is possible to save time for the human reviewers and obtain good performance levels, similar to the ones obtained by humans. In this current study we show that we can achieve this by building a classification model that is based on the natural human workflow used for building systematic reviews. We show, on a real data set, that a human-machine system obtains the best results when an ensemble of classifiers is used as the classification model.

2 Related Work

The traditional way to collect and triage the abstracts from a systematic review consists in using simple query search techniques based on MeSH[1] or keywords terms. The queries are usual Boolean-based and are optimized either for precision or for recall. The studies done by Haynes *et al.* (1994) show that it is difficult to obtain high performance for both measures.

The research done by Aphinyanaphongs and Aliferis (2005) is probably the first application of automatic text classification to the task of creating systematic reviews. In that paper the authors experimented with a variety of text classification techniques using the data derived from the ACP Journal Club as their corpus. They found that support vector machine (SVM) was the best classifier according to a variety of measures. Further work for systematic reviews was done by Cohen et al. (2006). Their work is mostly focused on the elimination of non relevant documents. As their main goal is to save work for the reviewers involved in systematic review preparation, they define a measure, called work saved over sampling (WSS) that captures the amount of work that the reviewers will save with respect to a baseline of just sampling for a given value of recall. The idea is that a classifier returns, with high recall, a set of abstracts, and only those abstracts need to be read to weed out the non-relevant ones. The savings are measured with respect to the number of abstracts that would have to be read if a random baseline classifier was used. Such baseline corresponds to uniformly sampling a given percentage of abstracts (equal to the desired recall) from the entire set. In Cohen *et al.* (2006), the WSS measure is applied to report the reduction in reviewer's work when retrieving 95% of the relevant documents; the precision was very low.

We focus on developing a classifier for systematic review preparation, relying on characteristics of the data that were not included in the Cohen et *al.'s* (2006), because the questions asked in the preparation of the reviews are not available, Therefore we cannot perform a direct comparison of results here. Also, the data sets that they used in their experiments are significantly smaller than the one that we used.

3 The Data Set

A set of 47,274 abstracts with titles were collected from MEDLINE[2] as part of a systematic review done by the McMaster University's Evidence-Based Practice Center using TrialStat Corporation's Systematic Review System[3], a web-based software platform used to conduct systematic reviews.

The initial set of abstracts was collected using a set of Boolean search queries that were run for

[1] http://www.nlm.nih.gov/mesh/

[2] http://medline.cos.com

[3] http://www.trialstat.com/

the specific topic of the systematic review: "*the dissemination strategy of health care services for elderly people of age 65 and over*".

In the protocol applied, two reviewers work in parallel. They read the entire collection of 47,274 abstracts and answer a set of questions to determine if an abstract is relevant or not to the topic of review. Examples of questions present in the protocol: *Is this article about a dissemination strategy or a behavioral intervention?; Is this a primary study?; Is this a review?; etc.* An abstract is not considered to pass to the next screening phase, when the entire article is available, if the two reviewers respond negative to the same question for a certain abstract. All other cases of possible responses suggest that the abstract will be part of the next screening phase. In this paper we focus on the initial screening phase, the only source of information is the abstract and the title of the article, with the main goal to achieve an acceptable level of recall not to mistakenly exclude relevant abstracts.

From the entire collection of labeled abstracts only 7,173 are relevant. Usually in the process of building systematic reviews the number of non-relevant documents is much higher than the number of relevant ones. The initial retrieval query is purposefully very broad, so as not to miss any relevant papers.

4 Methods

The machine learning techniques that could be used in the process of automating the creation of systematic reviews need to take into account some issues that can arise when dealing with such tasks. *Imbalanced data* sets are usually what we deal with when building reviews, the proportion of relevant articles that end up being present in the review is significantly lower compared with the original data set. The benefit of doubt will affect the quality of the data used to train the classifier, since a certain amount of *noise* is introduced: abstracts that are in fact non-relevant can be labeled as being relevant in the first screening process. The relatively high number of abstracts involved in the process will make the classification algorithms deal with a high number of features and the *representation* technique should try to capture aspects pertaining of the medical domain.

4.1 Representation Techniques

In our current research, we use three representation techniques: bag-of-words (BOW), concepts from the Unified Medical Language System (UMLS), and a combination of both.

The **bag-of-words** representation is commonly used for text classification and we have chosen to use binary feature values. Binary feature values were shown to out-perform weighted values for text classification tasks in the medical domain as shown by Cohen et *al.* (2006) and binary values tend to be more stable in results than frequency values for a task similar to ours, as shown by Ma (2007).

We considered feature words delimitated by space and simple punctuation marks that appeared at least three times in the training data, were not part of a stop words list[4], and had a length greater than three characters. 30,000 word features were extracted. No stemming was used.

UMLS concepts which are part of the U.S. National Library of Medicine[5] (NLM) knowledge repository are identified and extracted form the collection of abstracts using the MetaMap[6] system. This conceptual representation helped us overcome some of the shortcomings of BOW representation, and allowed us to use multi-word features, medical knowledge, and higher-level meanings of words in context. As Cohen (2008) shows, multi-word and medical concept representations are suitable to use.

4.2 Classification Algorithms

As a classification algorithm we have chosen to use the complement naive Bayes (CNB) (Frank and Bouckaert, 2006) classifier from the Weka[7] tool. The reason for this choice is that the CNB classifier implements state-of-the-art modifications of the standard multinomial naïve Bayes (MNB) classifier for a classification task with highly skewed class distribution (Drummond and Holte, 2003). As the systematic reviews data usually contain a large majority of not relevant abstracts, resulting in a skewness reaching even below 1%, it is important to use appropriate classifiers. Other classifiers, such as decision tress,

[4] http://www.site.uottawa.ca/~diana/csi5180/StopWords
[5] http://www.nlm.nih.gov/pubs/factsheets/umls.html
[6] http://mmtx.nlm.nih.gov/
[7] www.cs.waikato.ac.nz/machine learning/weka/

support vector machine, instance-based learning, and boosting, were used but the results obtained with CNB were always better.

4.3 Global Text Classification Method

The first method that we propose in order to solve the text classification task that is intended to help a systematic review process is a straightforward machine learning approach. We trained a classifier, CNB, on a collection of abstracts and then evaluated the classifier's performance on a separate test data set. The power of this classification technique stands in the ability to use a suitable classification algorithm and a good representation for the text classification task; Cohen *et al.* (2006) also used this approach. We randomly split the data set described in Section 3, into a training set and a test set. The two possible classes are **Included (relevant)** or **Excluded (non relevant)**. We decided to work with a training set smaller than the test set because ideally good results need to be obtained without using too much training data. We have to take into consideration that training a classifier for a particular topic, human effort is required for annotation.

Table 1 presents a summary of the data along with the class distribution in the training and test data sets. We randomly sampled the data to build the training and test data sets, and the original distribution of 1:5.6 between the two classes holds in both sets.

Data set	No. of abstracts	Class distribution Included : Excluded (ratio)
Training	20,000	3,056 : 16,944 (1:5.6)
Testing	27,274	4,117 : 23,157 (1:5.6)

Table 1. Training and test data sets.

4.3.1 Feature Selection

Using the global method, we performed experiments with several feature selection algorithms. We used only the BOW representation.

Chi^2 is a measure that evaluates the worth of an attribute by computing the value of the chi-squared statistic with respect to the class. We selected the top k_1 CHI^2 features that are exclusively included (appeared only in the training abstracts that are classified as **Included**) and the top k_2 CHI^2 features that are exclusively excluded (appeared only in the training abstracts that are classified as **Excluded**) and used them as a representation for our data set. We varied the k_1 parameter from 10 to 150 and k2 from 5 to 150 We used a minimum of 20 features and a maximum of 300.

InfoGain evaluates the worth of an attribute by measuring the information gain with respect to the class. We run experiments when we varied the number of selected features from 50 to 500. We used a number of 50, 100, 150, 250, 300 and 500 top features.

Bi-Normal Separation (**BNS**) is a feature selection technique that measures the separation between the threshold occurrences of a feature in one of the two classes. The latter measure is described in detail in Forman (2002). We used a ratio of features that varies from 10 to 150 for the most representative features for the **Included** class and from 5 to 150 for the **Excluded** class. For some experiments the number of features for the **Included** class is higher than the number of features for the **Excluded** class. We have chosen to do so because we wanted to re-balance the imbalance of classes in the training data set. After selecting the number of **Included** and **Excluded** features, we used the combination to represent our entire collection of abstracts.

We used the implementation from the Weka package for the **Chi^2** and **InfoGain** and the BNS implementation done by Ma (2007).

4.4 Per-Question Classification Method

The second method that we propose for solving the task takes into account the specifics of the systematic review process. It takes advantage of the set of questions the reviewers use in the process of deciding if an abstract is relevant or not. These questions are created in the design step of the systematic review and almost all systematic reviews have them. By using these questions we better emulate how the human judges work when building systematic reviews.

We have chosen to use only the questions that have inclusion/exclusion criteria, there were also some opened answer questions involved in the review, because they are the ones that are important for reviewers to make a decision. To collect training data for each question, we used the same training and test data set as in the previous method (but note that not all the abstracts

have answers for all the questions; therefore the training set sizes differ for each question). Table 2 presents the questions and data sets used.

When we created a training data set for each question we removed the abstracts for which we had a disagreement between the human experts – two different answers for a specific question, they represent noise in the training data. For each of the questions from Table 2, we trained a CNB classifier on the corresponding data set.

Question (Training : Included class : Excluded class)
Q1 - Is this article about a dissemination strategy or a behavioural intervention? **(14,057:1,145:12,912)**
Q2 - Is the population in this article made of individuals 65-year old or older or does it comprise individuals who serve the elderly population needs (i.e. health care providers, policy makers, organizations, community)? **(15,005:7,360:7,645)**
Q3 - Is this a primary study? **(8,825:6,895:1,930)**
Q4 - Is this a review? **(6,429:5,640:789)**

Table 2. Data sets for the per-question classification method.

We used the same representation for the per-question classifiers as we did for the global classifier: BOW, UMLS (the concepts that appeared only in the new question-oriented training data sets), and the combination BOW+UMLS. We used each trained model to obtain a prediction for each instance from the test set; therefore each test instance was assigned four prediction values of 0 or 1. To assign a final class for each test instance, from the prediction of all four classifiers, the class of a test instance is decided according to one of the following four schemes:

1. If any one vote is **Excluded**, the final class of a test instance is **Excluded**. This is a 1-vote scheme.

2. If any two votes are **Excluded**, the final class of a test instance is **Excluded**. This is a 2-vote scheme.

3. If any three votes are **Excluded**, the final class of a test instance is **Excluded**. This is a 3-vote scheme.

4. If all four votes are **Excluded**, the final class of a test instance is **Excluded.** This is a 4-vote scheme.

When we combined of the classifiers, we gave each classifier an equal importance.

5 Evaluation Measures and Results

When performing the evaluation for the task of classifying an abstract into one of the two classes **Included (relevant)** or **Excluded (non relevant)**, two objectives are of great importance: **Objective 1** - ensure the completeness of the systematic review (maximize the number of relevant documents included); **Objective 2** - reduce the reviewers' workload (maximize the number of irrelevant documents excluded).

We observe that objective 1 is more important than objective 2 and this is why we decided to report recall and precision for the **Included** class. We also report F-measure, since we are dealing with imbalanced data sets.

Besides the standard evaluation measures, we report WSS[8] measure as well in order to give a clearer view of the results we obtain.

As baseline for our methods we consider: two extreme baselines and a random-baseline classifier that takes into account the distribution of the two classes in the training data set. The baselines results are: *Include_All* – a baseline that classifies everything in the majority class: Recall = 100%, Precision = 15%, F-measure = 26.2%; WSS = 0% *Exclude_All* – a baseline that classifies everything as **Excluded**: Recall = 0%, Precision = 100%, F-measure = 64.2%; WSS = 0% *Random baseline*: Recall = 8.9%, Precision = 15.4%, F-measure = 67.8%; WSS = 0.23%.

5.1 Results for the Global Method

In this subsection, we present the results obtained using our global method with the three representation techniques and CNB as classification algorithm. To get a clear image of the results we show the confusion matrix in Table 3 for the reader to better understand the workload reduction when using classifiers to help the process of building systematic reviews.

BOW features were identified following the guidelines presented in Section 3.4 and a number of 23,906 features were selected. UMLS concepts were identified using the MetaMap system.

[8] WSS = (TE + FE)/(TE + FE + TI + FI) – 1+ TI/(TI + FE) where T stands for true; F – false I – Included class; E- Excluded class.

	BOW	UMLS	BOW+UMLS
True Inc.	*2,692*	*2,793*	*2,715*
False Inc.	*5,022*	*8,922*	*5,086*
True Exc.	*18,135*	*14,235*	*18,071*
False Exc.	*1,425*	*1,324*	*1,402*
Recall	65.3%	67.8%	65.9%
Precision	34.9%	23.8%	34.8%
F-measure	45.5%	35.2%	45.5%
WSS	37.1%	24.9%	37.3%

Table 3. Results for the global method.

From the whole training abstracts collection, a number of 459 UMLS features were identified. Analyzing the results from Table 5, in terms of recall, the UMLS representation obtained the best recall results, 67.8% for the global method but much lower precision, 23.8% than BOW representation, 34.9%. The hybrid representation, BOW+ UMLS features had similar results with the BOW alone. Recall increased a bit for the hybrid representation compared to BOW alone, 0.6% but its value is still not acceptable. We conclude that the levels of recall, our main objective for this task, were not acceptable for a classifier to be used as replacement of a human judge in the workflow of building a systematic review. The levels of precision that we obtained with the global method are acceptable but they cannot substitute the low level of recall. Since our major focus is recall, we investigated more and we further improved our precision scores with the per-question classification method.

5.1.1 Results for Feature Selection

Table 4 presents the results obtained with our feature selection techniques. We decided to report only representative results using CNB as a classifier and a specific representation setting. The number of features used in the experiment is presented in the round brackets. The first number represents the number of features extracted from the **Included** class data set while the second from the **Excluded** class data set.

Similar experiments were performed when using Naïve Bayes as classifier. The results obtained were opposite to ones obtained for CNB, all abstracts were classified as **Excluded**. We believe that this is the case because the CNB classifier tries to compensate for the class imbalance and gives more credit to the minority class,

	Chi^2 (150:150)	InfoGain (300)	BNS (10:8)
True Inc.	*3,819*	*3,875*	*2,690*
False Inc.	*19,233*	*19,638*	*13,905*
True Exc.	*3,924*	*3,518*	*9,253*
False Exc.	*298*	*242*	*1,427*
Recall	92.8%	94.1%	65.3%
Precision	16.6%	16.5%	16.2%
F-measure	28%	28%	25%
WSS	8.2%	7.9%	4.5%

Table 4. Representative results obtained for various feature selection techniques.

while the Naïve Bayes classifier will let the majority class overwhelm the classifier.

Besides the results presented in Table 4, we also tried to boost the representative features for the **Included** class hoping to re-balance the imbalance present in the training data set. To perform these experiments we selected the top k CHI^2 word features and then added to this set of features the top k_1 CHI^2 representative features only for the **Included** class. The parameter k varied from 50 to 100 and the parameter k_1 from 30 to 70. We performed experiments when using the original imbalanced training data set and using a balanced data set as well, with both CNB and Naïve Bayes classifier. The results obtained for these experiments were similar to the ones when we used the previous feature selection techniques. There was no significant difference in the results compared to the ones in Table 5.

5.2 Results for the Per-Question Method

The results for our second method using the four voting schemes are presented in Table 5.

Compared with the global method the results obtained by the per-question method, especially the ones for 2 votes are the best so far in terms of the balance between the two objectives. A large number of abstracts that should be excluded are classified as **Excluded** whereas wrongly excluding very few abstracts that should have been included (a lot fewer than in the case of the global classification method).

The 2-votes scheme performs better than the 1-vote schemes because of potential classification errors. When the classifiers for two different questions (that look at two different aspects of the systematic review topic) are confident that the abstract is not relevant, the chance of correct

prediction is higher; a balance between excluding an article and keeping it as relevant is achieved. When using the classifiers for 3 or 4 questions the performance goes down in terms of precision; a higher number of abstracts get classified as **Included** - some abstracts do not address all target question of the review topic.

1-Vote	BOW	UMLS	BOW+UMLS
True Inc.	*1,262*	*1,222*	*1,264*
False Inc.	*745*	*2,266*	*741*
True Exc.	*22,412*	*20,891*	*22,416*
False Exc.	*2,855*	*2,895*	*2,853*
Recall	30.6%	29.6%	30.7%
Precision	62.8%	35.0%	63.0%
F-measure	41.2%	32.1%	41.2%
WSS	23.2%	16.8%	23.3%
2-Vote	**BOW**	**UMLS**	**BOW+UMLS**
True Inc.	*3,181*	*2,603*	*3,283*
False Inc.	*9,976*	*9,505*	*10,720*
True Exc.	*13,181*	*13,652*	*12,437*
False Exc.	*936*	*1,514*	*834*
Recall	77.2%	63.2%	79.7%
Precision	24.1%	21.5%	23.4%
F-measure	36.8%	32.0%	36.2%
WSS	29.0%	18.8%	28.4%
3-Vote	**BOW**	**UMLS**	**BOW+UMLS**
True Inc.	*3,898*	*3,480*	*3,890*
False Inc.	*18,915*	*16,472*	*18,881*
True Exc.	*4,242*	*6,685*	*4,276*
False Exc.	*219*	*637*	*227*
Recall	94.6%	84.5%	94.4%
Precision	17.0%	17.4%	17.0%
F-measure	28.9%	28.9%	28.9%
WSS	11.0%	11.3%	11.0%
4-Vote	**BOW**	**UMLS**	**BOW+UMLS**
True Inc.	*4,085*	*3,947*	*4,086*
False Inc.	*21,946*	*20,869*	*21,964*
True Exc.	*1,211*	*2,288*	*1,193*
False Exc.	*32*	*170*	*31*
Recall	99.2%	95.8%	99.2%
Precision	15.6%	15.9%	15.6%
F-measure	27.1%	27.2%	27.0%
WSS	3.7%	4.8%	3.7%

Table 5. Results for the per-question method for the **Included** class.

For the per-question technique the recall value peaked at 99.2% with the 4-vote method BOW and BOW+UMLS representation technique. In the same time the lowest values of precision for the per-question technique, 15.6% is obtained with the same experimental setting. It is important to aim for a high recall but not to dismiss the precision values. The difference of even less than 2% in precision values can cause the reviewers to read additional thousands of documents, as observed in the confusion matrices for 2-vote, 3-vote and 4-vote methods in Table 5.

From the confusion matrix in Table 5 for the 2-vote method and the 3- and 4-vote method we observe the high difference in the number of documents a reviewer will have to read (the falsely included documents). The difference in precision from 24.1% for the 2-vote method to 15.6% for the 4-vote method makes the reviewer go through 11,988 additional abstracts.

The best value for the WSS measure for the per-question method is achieved by the 2-vote scheme. The result is lower than the one obtained by the global method but the recall level is higher therefore, we still keep as a potential winner the 2-vote scheme.

5.3 Results for Human-Machine Workflow

In Figure 1, we envisioned the way we can use the automatic classifier in the workflow of building a systematic review. In order to determine the performance of the human-machine workflow that we propose we computed the recall values when the human reviewer's labels are combined with the labels obtained from the classifier. The same labeling technique is applied as for the human-human workflow: if at least one decision for an abstract is to include it in the systematic review, then the final label is **Included**.

We also calculated the evaluation measures for the two reviewers. The evaluation measures for the human judge that is kept in the human-machine workflow, Reviewer 1 in Figure 1, are 64.29% for recall and 15.20% for precision. The evaluation measures for the reviewer that is to be replaced in the human-machine classification, Reviewer 2 in Figure 1 are 59.66% for recall and 15.09% for precision. The recall value for the two human judges combined is 85.26% and the precision value is 100%. As we can observe the recall value for the second reviewer, the one that is replaced in the human-classifier workflow is low. In Table 6 we present precision and recall results for the symbiotic model for both our methods. In these results we can clearly see that the 2-vote technique is superior to the other voting techniques and to the global method. For almost the same level of precision the level or recall it is much higher. These observations support the fact

that the extra effort spent in identifying the most suitable methodology pays off.

The fact that we keep a human in the loop makes our method acceptable as a workflow for building a systematic review.

Method	BOW	UMLS	BOW+ UMLS
Global	17.9/87.7%	17.0/88.6%	17.9/87.7%
1-Vote	17.1/75.3%	16.5/74.8%	17.1/75.4%
2-Vote	17.1/91.6%	16.4/86.6%	17.1/92.7%
3-Vote	15.8/97.9%	15.8/94.2%	15.8/97.8%
4-Vote	15.3/99.6%	15.4/98.3%	15.3/99.6%

Table 6. Precision/recall results for the human-classifier workflow for the **Included** class.

6 Discussion

The global method achieves good results in terms of precision while the best recall is obtained by the per-question method.

The best results for the task were obtained using the per-question method with the 2-vote scheme with or without UMLS features. The 3-vote scheme with UMLS representation is close to the 2-vote scheme but looking at F-measure and WSS results the 2-vote scheme is better. The clear distinction between the methods comes when we combined the classifiers with the human judge in the workflow of building reviews.

The per-question technique is more robust and it offers the possibility to choose the desired type of performance. If the reviewers are willing to read almost the entire collection of documents, knowing that the recall is high, then a 3 or 4-vote scheme can be the set-up (though the 3 or 4-vote method is not likely to achieve 100% recall because it is very rare that an abstract contain answers to three or four of the questions associated with the systematic review). If the reviewers will like to read a small collection being confident that almost all the abstracts are relevant, then a 1-vote scheme can be the set-up required. The per-question method confirms the fact that an ensemble of classifiers is better than one classifier; (Dietterich, 1997).

When we combine the human and the system results we obtain a major improved in terms of recall. We base our discussion for the human-machine results for the experiment that obtained the best results, the 2-vote scheme with a BOW+UMLS representation technique. When combining the human and classifier decisions, the precision level decreased a bit compared to the one that the machine obtained. We believe that this is the case because some of the abstracts that the classifier excluded were included by the first human reviewer and, with this decision process in place, the level of precision dropped.

Our goal of improving the recall level from the first level of screening is achieved, since when both the classifier and the human judge are integrated in the workflow, the recall level jumps from 79.7% to 92.7%.

We believe that the low level of precision that is obtained for the human reviewer, for the human-classifier workflow, and for the classifier, is due to the fact that we are running experiments for the first screening phase when we use only the abstracts as source of information and not the entire articles.

We believe that further investigations are required to fully replace a human reviewer with an automatic classifier but the results obtained with the per-question method encourage us to believe that this is a suitable solution for reaching our final goal.

7 Conclusions and Future Work

In this paper, we looked at two methods by which we envision the way automatic text classification techniques could help the workflow of building systematic reviews.

The first method is a straight-forward application of the representations and learning algorithms that capture the specifics of the data: medical domain, huge number of features, misclassification, and imbalanced classes.

We showed that the specifics of the human protocol in which systematic reviews are built have a positive effect when deployed in an automatic way. We believe that the tedious process that is currently used for building systematic reviews can be lightened by the use of a classifier in combination with only one human judge. By having a human judge in the loop, we ensure that the workflow is reliable and that the system can be easily integrated in the workflow.

In future work we would like to look into ways of improving the results by the way we chose the training data set and by integrating more domain specific knowledge. We would also like to investigate ways by witch we can update systematic reviews.

References

Aphinyanaphongs Y. and Aliferis C. *Text Categorization Models for Retrieval of High Quality Articles.* Journal of the American Medical Informatics Association 2005; 12:207-216.

Cohen A.M. *Optimizing Feature Representation for Automated Systematic Review Work Prioritization.* Proceedings of the AMIA Annual Symposium 2008; 6:121-126.

Cohen A.M., Hersh W.R., Peterson K., Yen P.Y. *Reducing Workload in Systematic Review Preparation Using Automated Citation Classification.* Journal of the American Medical Informatics Association 2006; 13:206-219.

Dietterich, T. *Machine-Learning Research: Four Current Directions*. Artificial Intelligence Magazine. 18(4): 97-136 (1997)

Drummond C. and Holte R.C. *C4.5, Class Imbalance, and Cost Sensitivity: Why Under-Sampling beats Over-Sampling.* Proceedings of the Twentieth International Conference on Machine Learning: Workshop on Learning from Imbalanced Data Sets (II), 2003.

Forman G. *Choose Your Words Carefully: An Empirical Study of Feature Selection Metrics for Text Classification.* In the Joint Proceedings of the 13th European Conference on Machine Learning and the 6th European Conference on Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD), 2002.

Frank E. and Bouckaert R.R. *Naive Bayes for Text Classification with Unbalanced Classes.* In the Proceedings of the 10th European Conference on Principles and Practice of Knowledge Discovery in Databases, Berlin, Germany, 2006, pp. 503-510.

Haynes R.B., Wilczynski N., McKibbon K.A., Walker C.J., Sinclair J.C. *Developing optimal search strategies for detecting clinically sound studies in MEDLINE.* Journal of the American Medical Informatics Association 1994; 1:447-58.

Kohavi R. and Provost F. *Glossary of Terms*. Editorial for the Special Issue on Applications of Machine Learning and the Knowledge Discovery Process 1998; 30:271-274.

Ma Y. 2007. Text classification on imbalanced data: Application to Systematic Reviews Automation. M.Sc. Thesis. University of Ottawa.

Chinese Sentence-Level Sentiment Classification Based on Fuzzy Sets

Guohong Fu and Xin Wang
School of Computer Science and Technology, Heilongjiang University
ghfu@hlju.edu.cn,wangxincs@hotmail.com

Abstract

This paper presents a fuzzy set theory based approach to Chinese sentence-level sentiment classification. Compared with traditional topic-based text classification techniques, the fuzzy set theory provides a straightforward way to model the intrinsic fuzziness between sentiment polarity classes. To approach fuzzy sentiment classification, we first propose a fine-to-coarse strategy to estimate sentence sentiment intensity. Then, we define three fuzzy sets to represent the respective sentiment polarity classes, namely positive, negative and neutral sentiments. Based on sentence sentiment intensities, we further build membership functions to indicate the degrees of an opinionated sentence in different fuzzy sets. Finally, we determine sentence-level polarity under maximum membership principle. We show that our approach can achieve promising performance on the test set for Chinese opinion analysis pilot task at NTCIR-6.

1 Introduction

With the explosive growth of the user-generated content on the web over the past years, opinion mining has been attracting an ever-increasing amount of attention from the natural language processing community. As a key issue in opinions mining, sentiment classification aims to classify opinionated documents or sentences as expressing positive, negative or neutral opinions, and plays a critical role in many opinion mining applications such as opinion summarization and opinion question answering.

Although recent years have seen a great progress in sentiment analysis, it is still challenging to develop a practical sentiment classifier for open applications. This is largely due to the particularities of subjective languages. Unlike factual text, opinion text is usually expressed in a more subtle or arbitrary manner (Pang and Lee, 2008). Moreover, the sentiment orientation of a subjective expression is often context, domain and/or even order-dependent (Pang and Lee, 2008). This makes it hard to explore informative cues for sentiment classification. In particular, the final semantic orientation of an opinionated sentence often depends on the synthetic effects of all sentiment units (e.g. sentiment words or phrases) within it. Therefore, sentiment granularity selection and polarity aggregation are two important factors that affect sentiment classification performance.

In addition, real opinion texts do not contain precisely-defined criteria of membership with respect to polarity classes. Most current work employs supervised machine learning techniques like naive Bayesian models and support vector machines to perform sentiment classification. While they have shown a good performance in traditional topic-based text classification tasks (Wang, 2006), their applications in sentiment classification are far from satisfactory (Pang *et al.*, 2002). The reason might be the intrinsic fuzziness between sentiment polarity classes. Relative to the concept of objective topics like *sports* and *politics* in traditional text classification, the division between *positive sentiments* and *negative sentiments* is rather vague, which does not make clear boundary between their conceptual extensions. Such vague conceptual extension in sentiment polarity inevitably raises another challenge to sentiment classification.

Coling 2008: Poster Volume, pages 312–319,
Beijing, August 2010

To address the above problems, in this paper we exploit fuzzy set theory to perform Chinese sentiment classification at sentence level. To approach this task, we first consider multiple sentiment granularities, including sentiment morphemes, sentiment words and sentiment phrases, and develop a fine-to-coarse strategy for computing sentence sentiment intensity. Then, we reformulate the three classes of sentiment orientations, namely positive, negative and neutral sentiments, as three fuzzy sets, respectively. To describe the membership of an opinion sentence in a special sentiment fuzzy set, we further construct membership functions based on sentence sentiment intensity, and thus determine the final semantic orientation of a given opinionated sentence under the principle of maximum membership. We show that the proposed approach can achieve a promising performance on the test set for Chinese opinion analysis pilot task at NTCIR-6.

The remainder of the paper is organized as follows: Section 2 provides a brief review of the literature on sentiment classification. In Section 3, we describe the fine-to-coarse strategy for estimating sentiment intensity of opinionated sentences. Section 4 details how to apply fuzzy set theory in sentiment classification. Section 5 reports our experimental results on NTCIR-6 Chinese opinion data. Finally, section 6 concludes our work and discusses some possible directions for future research.

2 Related Work

Sentiment classification has been extensively studied at different granularity levels. At lexical level, Andreevskaia and Bergler (2006) exploit an algorithm for extracting sentiment-bearing adjectives from the WordNet based on fuzzy logic. Following (Turney, 2002), Yuen *et al.* (2004) investigate the association between polarity words and some strongly-polarized morphemes in Chinese, and present a method for inferring sentiment orientations of Chinese words. More recently, Ku *et al.* (2009) consider eight morphological types that constitute Chinese opinion words, and develop a machine learning based classifier for Chinese word-level sentiment classification. They show that using word structural features can improve performance in word-level polarity classification. At phase level, Turney (2002) presents a technique for inferring the orientation and intensity of a phrase according to its PMI-IR statistical association with a set of strongly-polarized seed words. More recently, Wilson *et al.* (2009) distinguish prior and contextual polarity, and thus describe a method to phrase-level sentiment analysis. At sentence level, Yu and Hatzivassiloglou (2003) propose to classify opinion sentences as positive or negative in terms of the main perspective being expressed in opinionated sentences. Kim and Hovy (2004) try to determine the final sentiment orientation of a given sentence by combining sentiment words within it. However, their system is prone to produce error sentiment classification because they only consider sentiment words near opinion holders and ignore some important words like adversative conjunctions. To compute sentiment intensity of opinionated sentences, in this study we propose a fine-to-coarse strategy, which take into account multiple granularity sentiments, from sentiment morphemes, sentiment words to sentiment phrases, and can thus handle both unknown lexical sentiments and contextual sentiments in sentiment classification.

Most recent studies apply machine learning techniques to perform sentiment classification. Pang *et al.* (2002) attempt three machine learning methods, namely naive Bayesian models, maximum entropy and support vector machines in sentiment classification. They conclude that the traditional machine learning methods do not perform well enough in sentiment analysis. Wilson *et al.* (2009) further employ several machine learning algorithms to explore important features for contextual polarity identification. Different from most existing works that focus on traditional text classification techniques, in this study we attempt to resolve sentiment classification problems under the framework of fuzzy set theory. We choose fuzzy set theory because it provides a more straightforward way to represent the intrinsic fuzziness in sentiment.

3 Sentence-Level Sentiment Intensity

In this section, we describe a fine-to-coarse strategy to compute sentence-level sentiment intensity. After a brief discussion of the relationship between Chinese sentiment words and their component morphemes in Section 3.1,

we extract a dictionary of sentiment morphemes from a sentiment lexicon, and compute their opinion scores using a modified chi-square technique. Then, we develop two rule-based strategies for word-level and phrase-level polarity identification, respectively. Finally, we calculate the final sentiment intensity of an opinionated sentence by summing the opinion score of all phrases within it.

3.1 Sentiment words and morphemes

As shown in Table 1, Chinese sentiment words can be categorized into static polar words and dynamic polar words. The polarity of a static polar word remains unchanged while a dynamic polar word may have different polarity in different contexts or domains.

Type		Example
Static polar word	Positive	美丽 'beautiful', 温柔 'gentle'
	Negative	卑劣 'beggary', 错误'wrong'
	Neutral	还可以 'acceptable'
Dynamic polar words		大'big', 高'high'

Table 1. Types of Chinese sentiment words

For a static polar word, its polarity can be easily determined by referring to a sentiment lexicon. However, a precompiled dictionary cannot cover all sentiment words in real text, which raises an issue of predicting the polarity of out-of-vocabulary (OOV) sentiment words. To address this problem, we introduce sentiment morphemes. As Table 2 shows, here we consider two types of sentiment morphemes, namely positive morphemes and negative morphemes.

Morpheme types	Sentiment morphemes	Sentiment words composed by sentiment morphemes
Positive morphemes	美'beauty'	精美 'exquisite'
		优美 'graceful'
	爱'love'	喜爱 'like'
		爱慕 'adoration'
Negative morphemes	污'dirty'	污染 'pollution'
		贪污 'corruption'
	败'fail'	腐败 'corruption'
		败坏 'undermine'

Table 2. Types of Chinese sentiment morphemes

In most cases, the polarity of a sentiment word is closely related to the semantic orientation of its component morphemes. In other words, word-level polarity can often be determined by some key component sentiment morphemes within sentiment words. Take the following three sentiment words for example, 败坏 'undermine', 腐败 'corruption', and 败类 'degenerate'. They share a same negative sentiment morpheme 败 'fail', and thus have the same negative orientation. Based on this observation, here we use morpheme-level polarity, rather than a sentiment lexicon, to predict the polarity of static sentiment words, particularly the OOV sentiment words in real text.

As for dynamic sentiment words, traditional lexicon-based methods do not work for their real polarity changes with contexts. We will discuss the problem of dynamic polarity identification in Section 3.4.

3.2 Identifying morpheme-level polarity

Sentiment morphemes prove to be helpful in dealing with OOV polarity (Ku et al, 2009). However, there is not a dictionary of sentiment morphemes available for sentiment analysis. To avoid this, we propose to automatically extract sentiment morphemes from some existing sentiment lexicon using chi-square (χ^2) technique. Formula (1) presents the χ^2 of a morpheme m within a sentiment word of category c.

$$\chi^2(m,c) = \frac{n \times (n_{11} \times n_{22} - n_{12} \times n_{21})^2}{(n_{11}+n_{12})(n_{21}+n_{22})(n_{11}+n_{21})(n_{12}+n_{22})} \quad (1)$$

where m denotes a sentiment morpheme. $c \in$ {positive, negative} denotes the polarity of a certain sentiment word w that contain m. n is the total number of sentiment words in the lexicon. To calculate χ^2, we need to construct a 2×2 contingency table from the sentiment lexicon. As shown in Table 3, n_{11}, n_{12}, n_{21} and n_{22} denote the observed frequencies, respectively.

Polar word w	belong to c	not belong to c
contain m	n_{11}	n_{12}
not contain m	n_{21}	n_{22}

Table 3. The 2×2 contingency table for χ^2

The traditional χ^2 statistics in Formula (1) can demonstrate the degree of contributions that a sentiment morpheme forms a special group of sentiment words. However, it cannot indicate whether the morpheme and the sentiment category are either positively- or anti-correlated.

Such information is very important for inferring word-level polarity from sentiment morphemes. To compensate for this deficiency, we modify the traditional χ^2 by injecting positive correlation and anti-correlation. Following (Wang, 2006), we introduce the following two rules in determining the sign of correlation between the sentiment category of words and their component sentiment morphemes.

- If $n_{11}\times n_{22}-n_{12}\times n_{21}>0$, the morpheme and the sentiment category are positively correlated. In this case, a larger χ^2 implies a higher likelihood that the morpheme belongs to the sentiment category.
- If $n_{11}\times n_{22}-n_{12}\times n_{21}<0$, the morpheme and the sentiment category are anti-correlated. In this case, a larger χ^2 value implies a higher likelihood that the morpheme does not belong to the sentiment category.

Thus, we obtain a modified χ^2 statistics as follows.

$$\chi^{2\prime}=sign(n_{11}\times n_{22}-n_{12}\times n_{21})\frac{n\times(n_{11}\times n_{22}-n_{12}\times n_{21})^2}{(n_{11}+n_{12})(n_{21}+n_{22})(n_{11}+n_{21})(n_{12}+n_{22})} \quad (2)$$

With the $\chi^{2\prime}$ statistic, we can build a dictionary of sentiment morphemes from a source sentiment lexicon, and further determine the polarity of each sentiment morpheme using the two rules as shown in Definitions 1 and 2.

Definition 1 (positive sentiment morphemes). If the $\chi^{2\prime}$ statistic between a morpheme *m* and positive sentiment words is greater than zero, then *m* can be identified as positive.

Definition 2 (negative sentiment morphemes). If the $\chi^{2\prime}$ statistic between a morpheme *m* and positive sentiment words is smaller than zero, then *m* can be identified as is negative.

Table 4 illustrates some extracted sentiment morphemes and their $\chi^{2\prime}$ values.

Types of morphemes	Examples	$\chi^{2\prime}$
Positive morphemes	美'beautiful'	111.78
	爱'love'	65.88
	喜'happy'	40.72
Negative morphemes	死'die'	-104.97
	败'failed'	-45.28
	恶'evil'	-72.37

Table 4. $\chi^{2\prime}$ values of sentiment morphemes

3.3 Identifying word-level polarity

To determine word-level polarity, we employ morpheme-based rules. First of all, we normalize the $\chi^{2\prime}$ value of each sentiment morpheme *m* into [-1, 1] by dividing it with the maximum absolute value. Such normalized chi-square, denoted by *chi*(*m*), is further viewed as the opinion score of the sentiment morpheme *m*. Thus, we can determine whether a word is a sentiment or not using a simple rule: if a word contains sentiment morphemes, it is a sentiment word. Finally, we can calculate the opinion score of a word *w* consisting of morphemes m_i, $(1\le i\le 2)$[1], using the following two rules.

- If m_1 is a negation, e.g. 不 'not' and 非 'non-', then $Score(w)= -1\times chi(m_2)$.
- If m_1 is not a negation morpheme, then $Score(w)=Sign(chi(m_i))\times Max(|chi(m_i)|)$. Where, $Max(|chi(m_i)|)$ is the largest absolute value among the opinion scores of morphemes within a word *w*, $Sign(chi(m_i))$ denotes the positive or negative sign of *m*, namely '-' and '+'.

3.4 Identifying phrase-level polarity

To handle contextual polarity, we apply lexical polarity to determine the sentiment orientation of phrases within an opinionated sentence. Based on (Hatzivassiloglou and Wiebe, 2000) and (Turney, 2002), we consider four types of structures (as shown in Table 5) during sentiment phrase extraction. To simplify the process, we reduce some function words like 的 ''s' and 与 'and' from the input sentences before extraction in that they have no influence on sentiment orientation determination, and focus on extracting two consecutive words. Different from (Turney, 2002), we consider phrases with negations as their initial words. In this way, we can handle the local negation that may reverse polarity.

Phrase structures	Examples
Phases containing a adjective	成功率高 'high success rate'
Phrases containing a verb	详细讨论'carefully discuss'
Phrase containing an idiom	企图掩人耳目/'intent to deceive the public'
Phrases beginning with a negation	没有证据'no evidence'

Table 5. Structures of opinion phrases

[1] For words that contain three or more characters, particularly the four-character idioms, their polarity can be determined using the second rule.

After opinion phrase extraction, we continue to calculate the opinion score of the extracted phrases using rules that are similar to (Hu and Liu, 2004). Before going to the details of phrase-level opinion score calculation, we need to give some definitions in advance.

Definition 3 (increased dynamic polar words). An increased dynamic polarity word can increase the orientation strength of sentiment words that it modifies without changing their polarity. For example, the word 大 'serious' in the phrase 污染大 'serious pollution' and the word 高 'high' in the phrase 效益高 'high benefit'.

Definition 4 (decreased dynamic polar word). A decreased dynamic polarity word can decrease the orientation strength of sentiment words that it modifies and at the same time, reverse their polarity. For example, the word "小" 'little' in the phrase 污染小 'little pollution' and the word 低 'low' in the phrase 效益低 'low benefit'.

To calculate phrase-level opinion scores, we construct a dictionary of dynamic polar words by extracting adjectives and verbs that contain a single-character seed morpheme like 少 'little' from the training corpus. Table 6 illustrates some increased and decreased dynamic polar words and their signs for changing polarity.

Dynamic polar word	Example	Polarity sign
Increased	高 'high' 增加 'increase' 提升 'upgrade' 下降 'down'	*Sign*(increased)=1
Decreased	减少 'reduce' 缩小 'diminish'	*Sign*(decreased)=−1

Table 6. Dynamic words and their polarity sign

With these dynamic polar words, we can then calculate the opinion score of a given opinion phrase p_i that consists of two words (denoted by w_j, $j \in \{1,2\}$), using three rules as follows.

- If w_1 is a negation, e.g. 不 'no' and 没有 'without', then $Score(p_i) = -1 \times Score(w_2)$.
- If p_i involves a dynamic word w_d, then $Score(p_i) = Sign(w_d) \times Score(w_j)$. Where, $Sign(w_d)$ denotes the polarity sign of dynamic words shown in Table 6.
- Otherwise, $Score(p_i) = Sign(w_j) \times Max(|Score(w_j)|)$. *Where* $Max(|Score(word_j)|)$ is the largest absolute value among the word-level opinion scores.

4 Sentence Sentiment Classification

4.1 Sentiment fuzzy sets and membership functions

As we have mentioned above, sentiment polarity is vague with regard to its conceptual extension. There is not a clear boundary between the concepts of "positive", "neutral" and "negative". To better handle such intrinsic fuzziness in sentiment polarity, we apply the fuzzy set theory by (Zadeh, 1965) to sentiment classification. To do so, we first redefine sentiment classes as three fuzzy sets, and then apply existing fuzzy distributions to construct membership functions for the three sentiment fuzzy sets.

In our formulation, all the opinionated sentences under discussion are represented as a sorted set, denoted by X, in terms of their opinion scores. Thus, we have $X = [Min(Opinion\ Score(S_i)), \ldots, Max(Opinion\ Score(S_i))]$. Where, $i=\{1,\ldots,n\}$, $Min(Opinion\ Score(S_i))$ and $Max(Opinion\ Score(S_i))$ denotes the respective minimum and maximum opinion scores. The details of the fuzzy sets and their membership functions are given in Definitions 5, 6 and 7, respectively.

Definition 5 (positive sentiment fuzzy set). if X is a collection of sentiment opinions (denoted by x), then a positive sentiment fuzzy set $\tilde{P}$ in X can be defined as a set of ordered pairs, namely

$$\tilde{P} = \{(x, \mu_{\tilde{P}}(x)) \mid x \in X\},$$

where $\mu_{\tilde{P}}(x)$ denotes the membership function of x in $\tilde{P}$ that maps X to the membership space M.

We choose the rise semi-trapezoid distribution (Zimmermann, 2001) as the membership function of the positive sentiment fuzzy set, namely

$$\mu_{\tilde{P}}(x) = \begin{cases} 0, & x < a \\ \dfrac{x-a}{b-a}, & a \le x \le b \\ 1, & x > b \end{cases} \quad (3)$$

where x denotes the opinion score of a sentence under discussion. The adjustable parameters a and b can be defined as $a = Min(x_i) + \lambda_1(Max(x_i) - Min(x_i)/k)$ and $b = Min(x_i) + \lambda_2(Max(x_i) - Min(x_i)/k)$, respectively. $Max(x_i)$ and $Min(x_i)$

denote the respective minimum and maximum values within X. λ_1, λ_2 and k are parameters. Here we set λ_1= 5.2, λ_2 = 5.4, and k = 10.

Definition 6 (neutral sentiment fuzzy set). if X is a collection of sentiment opinions (denoted by x), then a neutral sentiment fuzzy set $\tilde{E}$ in X can be defined as a set of ordered pairs, namely

$$\tilde{E} = \{(x, \mu_{\tilde{E}}(x)) \mid x \in X\},$$

where $\mu_{\tilde{E}}(x)$ denotes the membership function of x in $\tilde{E}$ that maps X to the membership space M.

As shown in Formula (4), we also select the semi-trapezoid distribution (Zimmermann, 2001) as the membership function of the neutral sentiment fuzzy set.

$$\mu_{\tilde{E}}(x) = \begin{cases} 0, & x < a \\ \dfrac{x-a}{b-a}, & a \le x < b \\ 1, & b \le x < c \\ \dfrac{d-x}{d-c}, & c \le x < d \\ 0, & x \ge d \end{cases} \quad (4)$$

where x denotes the opinion score of a sentence under test. a, b, c and d are adjustable parameters that can be defined as a = Min(xi) + λ1(Max(xi)-Min(xi)/k), b=Min(xi) +m1(Max(xi) - Min(xi)/k), c = Min(xi) + m2(Max(xi) - Min(xi)/k) and d= Min(xi) + λ2(Max(xi) - Min(xi)/k), respectively. Max(xi) and Min(xi) denotes the respective minimum and maximum values within X. λ1, λ2, m1, m2 and k are parameters, Here we set λ1 = 5.2, λ2 = 5.5, m1 = 5.26, m2 = 5.33, and k = 10.

Definition 7 (negative sentiment fuzzy set). if X is a collection of sentiment opinions (denoted by x), then a negative sentiment fuzzy set $\tilde{N}$ in X can be defined as a set of ordered pairs, namely

$$\tilde{N} = \{(x, \mu_{\tilde{N}}(x)) \mid x \in X\},$$

where $\mu_{\tilde{N}}(x)$ denotes the membership function of x in $\tilde{N}$ that maps X to membership space M.

To represent the membership function of the negative sentiment fuzzy set, we employ the drop semi-trapezoid distribution (Zimmermann, 2001), namely

$$\mu_{\tilde{N}}(x) = \begin{cases} 1, & x < a \\ \dfrac{b-x}{b-a}, & a \le x \le b \\ 0, & x > b \end{cases} \quad (5)$$

where x denotes the opinion score of a subjective sentence under discussion. The adjustable parameters a and b can be defined as $a = Min(x_i) + \lambda_1(Max(x_i) - Min(x_i)/k)$ and $b = Min(x_i) + \lambda_2(Max(x_i) - Min(x_i)/k)$, respectively. $Max(x_i)$ and $Min(x_i)$ refer to the corresponding minimum and maximum values in X. λ_1, λ_2, and k are parameters. Here we set λ_1=5.2, λ_2=5.3 and k=10.

4.2 Determining sentence polarity

Based on the above membership functions, we can now calculate the grade of membership of a given opinionated sentence in each sentiment fuzzy set, and thus determine its polarity under the principle of maximum membership. The basic idea is as follows: Let $\tilde{A}_1, \tilde{A}_2, \ldots, \tilde{A}_n$ be the fuzzy sets of X. $\exists x_0 \in X$, if

$$\tilde{A}_k(x_0) = \max_{1 \le i \le n} \{\tilde{A}_i(x_0)\}$$

then x_0 is a membership of the fuzzy set $\tilde{A}_k$.

5 Experiments and Results

To assess the effectiveness of our approach, we implemented a classification system for Chinese sentence-level sentiment analysis. The system involves three main modules, namely a lexical analysis module, a subjectivity detection module and a sentiment classification module. To explore lexical cues for sentiment analysis, the morpheme-based chunking technique by (Fu, Kit and Webster, 2008) is employed in the lexical analysis module to carry out word segmentation and part-of-speech tagging tasks. To conform to the NTCIR-6 evaluation, a sentiment density-based naive Bayesian classifier is also embedded in the second module to perform opinionated sentence detection. The details of this classifier can be seen in (Wang and Fu, 2010). To evaluate our system, we conducted experiments on the NTCIR-6 Chinese opinion data. This section reports the experimental results.

5.1 Experimental setup

In our experiments, we use the same test set for the Chinese opinion analysis tasks at NTCIR-6. The basic statistics is presented in Table 7. For comparison, the performance is reported in terms of the same metrics as used in NTCIR-6. They are F-score (F), recall (R), precision (P) under the LWK evaluation with lenient standard.

Item	Number
Topics	32
Documents	843
Sentences	11907
Opinionated sentences under the lenient standard	62%

Table 7. Basic statistics of the test set for Chinese opinion tasks at NTCIR-6

The basic sentiment lexicon used in our system contains a total of 17138 sentiment words, which is built from the CUHK and NTU sentiment lexica by excluding some derived opinion words like 不美丽 'not beautiful'. In addition, we also construct a list of 95 dynamic polarity words using the method described in Section 3.4.

5.2 Experimental results

The experiments are designed to examine the following two issues:

(1) As we have discussed above, it is a key issue to select a proper granularity for sentiment classification. To determine the sentiment orientation of an opinionated sentence, we use a fine-to-coarse strategy that considers three types of sentiment units, namely sentiment morphemes, sentiment words and sentiment phrases. Therefore, the first intention of our experiments is to investigate how the use of different sentiment granularity affects the performance of Chinese sentence-level sentiment classification. To do this, we take the above three sentiment granularity as the basic units for computing sentence-level sentiment intensity, respectively, and examine the relevant sentiment classification results.

(2) To the best of our knowledge, this study may be the first attempt to apply the fuzzy set theory in Chinese sentiment classification. Therefore, our second motivation is to examine whether it is feasible to apply fuzzy set theory in sentiment classification by comparing our system with other public systems for Chinese opinion analysis pilot task at NTCIR-6.

Table 8 presents the experimental results with different sentiment granularities. It can be observed that the system with word as the basic sentiment units slightly performs better than the system based on sentiment morphemes. But a prominent improvement of performance can be obtained after using sentiment phrases. This reason may be that under the fine-to-coarse framework, sentiment classification based on sentiment phrases can handle both internal and external contextual sentiment information, and can thus result in performance improvement.

Granularity	P	R	F
Morpheme	0.389	0.480	0.430
Word	0.393	0.485	0.434
Phrase	0.415	0.512	0.458

Table 8. Performance on sentiment classification with different sentiment granularity

Table 9 illustrates the comparison of our system with the best system for Chinese opinion analysis pilot task at NTCIR-6, namely the CUHK system (Seki *et al.*, 2007; Xu, Wong and Xia, 2007). As can be seen from Table 9, our system outperforms the CUHK system by 5 percents with regard to F-score, showing the feasibility of using fuzzy set theory in sentiment classification.

System	P	R	F
CUHK	0.522	0.331	0.405
Our system	0.415	0.512	0.458

Table 9. Comparison of our system with the best system at NTCIR-6 under lenient standard

6 Conclusion and Future Work

In this paper, we have described a fuzzy set theory based framework for Chinese sentence-level sentiment classification. To handle unknown polarity and contextual polarity as well, we consider three types of sentiment granularities, namely sentiment morphemes, words and phrases in calculating sentiment intensity of opinionated sentenced. Furthermore, we define three fuzzy sets to represent polarity classes and construct the relevant membership functions, respectively. Compared with most existing work, the proposed approach provides a straightforward way to model the vagueness in conceptual division of sentiment polarity. The experimental results show that our system outperforms the best system for Chinese opinion analysis pilot task at NTCIR-6 under the lenient evaluation standard.

The encouraging results of the fuzzy set-based approach suggest several possibilities for future

research. Our experiments demonstrate that the incorporation of multiple granularity polarity has a positive effect on sentiment classification performance. To further enhance our system, in future we intend to exploit more tailored techniques for aggregating multiple-granularity polarity within opinionated sentences. Moreover, we plan to optimize the proposed membership functions for fuzzy sentiment classification.

Acknowledgments

The authors would like to thank Chinese University of Hong Kong, National Taiwan University and NTCIR for their data. This study was supported by National Natural Science Foundation of China under Grant No.60973081, the Returned Scholar Foundation of Educational Department of Heilongjiang Province under Grant No.1154hz26, and Harbin Innovative Foundation for Returnees under Grant No.2009RFLXG007, respectively.

References

Alina Andreevskaia, and Sabine Bergler. 2006. Mining WordNet for a fuzzy sentiment: Sentiment tag extraction from WordNet glosses. In *Proceedings of EACL-06*, pages 209-216.

Xiaowen Ding, Bing Liu, and Philip S. Yu. 2008. A holistic lexicon-based approach to opinion mining. In *Proceedings of the International Conference on Web Search and Web Data Mining*, pages 231-240.

Guohong Fu, Chunyu Kit, and Jonathan J. Webster. 2008. Chinese word segmentation as morpheme-based lexical chunking. *Information Sciences*, 7(1):2282–2296.

Vasileios Hatzivassiloglou, and Janyce Wiebe. 2000. Effects of adjective orientation and gradability on sentence subjectivity. In *Proceedings of ACL-00*, pages 299-305.

Soo-Min Kim, and Eduard Hovy. 2004. Determining the sentiment of opinions. In *Proceedings of COLING-04*, pages 1267-1373.

Lun-Wei Ku, Ting-Hao Huang, and Hsin-Hsi Chen. 2009. Using morphological and syntactic structures for Chinese opinion. In *Proceedings of EMNLP-09*, pages 1260-1269.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumps up? Sentiment classification using machine learning techniques. In *Proceedings of EMNLP-02*, pages 79-86.

Bo Pang, and Lillian Lee. 2008. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1-2): 1-135.

Yohei Seki, David Kirk Evans, Lun-Wei Ku, Hsin-Hsi Chen, Noriko Kando, and Chin-Yew Lin. 2007. Overview of opinion analysis pilot task at NTCIR-6. In *Proceedings of NTCIR-6 Workshop Meeting*, pages 265-278.

Peter D. Turney. 2002. Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In *Proceedings of ACL-03*, pages 417-424.

Xin Wang, and Guohong Fu. 2010. Chinese subjectivity detection using a sentiment density-based naïve Bayesian classifier. In *Proceedings of IWWIP-10*.

Yu Wang. 2006. *Research on text categorization based on decision tree and K-nearest neighbors*. PhD thesis, Tianjin University.

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2009. Recognizing contextual polarity: An exploration of features for phrase-level sentiment analysis. *Computational Linguistics*, 35(3):99-433.

Ruifeng Xu, Kam-Fai Wong, and Yunqing Xia. 2007. Opinmine: Opinion mining system by CUHK for NTCIR-06 Pilot Task. In *Proceedings of NTCIR-6 Workshop Meeting*, pages 350-357.

Hong Yu, and Vasileios Hatzivassiloglou. 2003. Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. In *Proceedings of EMNLP-03*, pages 129-136.

Raymond W.M. Yuen, Terence Y.W. Chan, Tom B.Y. Lai, O.Y. Kwong, and Benjamin K.Y. T'sou. 2004. Morpheme-based Derivation of Bipolar Semantic Orientation of Chinese Words. In *Proceedings of COLING-04*, pages 1008-1014.

Lotfi A. Zadeh. 1965. Fuzzy sets. *Information and Control*, 8:338-353.

Hans-Jürgen Zimmermann. 2001. Fuzzy set theory and its applications. Kluwer Academic Publishers, Norwell, MA, USA.

Monolingual Distributional Profiles for Word Substitution in Machine Translation

Rashmi Gangadharaiah rgangadh@cs.cmu.edu
Ralf D. Brown ralf@cs.cmu.edu
Jaime Carbonell jgc@cs.cmu.edu

Language Technologies Institute,
Carnegie Mellon University

Abstract

Out-of-vocabulary (OOV) words present a significant challenge for Machine Translation. For low-resource languages, limited training data increases the frequency of OOV words and this degrades the quality of the translations. Past approaches have suggested using stems or synonyms for OOV words. Unlike the previous methods, we show how to handle not just the OOV words but *rare* words as well in an Example-based Machine Translation (EBMT) paradigm. Presence of OOV words and rare words in the input sentence prevents the system from finding longer phrasal matches and produces low quality translations due to less reliable language model estimates. The proposed method requires only a monolingual corpus of the source language to find candidate replacements. A new framework is introduced to score and rank the replacements by efficiently combining features extracted for the candidate replacements. A lattice representation scheme allows the decoder to select from a beam of possible replacement candidates. The new framework gives statistically significant improvements in English-Chinese and English-Haitian translation systems.

1 Introduction

An EBMT system makes use of a parallel corpus to translate new sentences. Each input sentence is matched against the source side of a training corpus. When matches are found, the corresponding translations in the target language are obtained through sub-sentential alignment. In our EBMT system, the final translation is obtained by combining the partial target translations using a statistical target Language Model. EBMT systems, like other data-driven approaches, require large amounts of data to function well (Brown, 2000).

Having more training data is beneficial resulting in log-linear improvement in translation quality for corpus-based methods (EBMT, SMT). Koehn (2002) shows translation scores for a number of language pairs with different training sizes translated using the Pharaoh SMT toolkit (Koehn et al., 2003). However, obtaining sizable parallel corpora for many languages is time-consuming and expensive. For rare languages, finding bilingual speakers becomes especially difficult.

One of the main reasons for low quality translations is the presence of large number of OOV and rare words (low frequency words in the training corpus). Variation in domain and errors in spelling increase the number of OOV words. Many of the present translation systems either ignore these unknown words or leave them untranslated in the final target translation. When data is limited, the number of OOV words increases, leading to the poor performance of the translation models and the language models due to the absence of longer sequences of source word matches and less reliable language model estimates.

Approaches in the past have suggested using stems or synonyms for OOV words as replacements (Yang and Kirchhoff, 2006). Similarity measures have been used to find words that are closely related (Marton et al., 2009). For morpho-

Coling 2008: Poster Volume, pages 320–328,
Beijing, August 2010

logically rich languages, the OOV word is morphologically analyzed and the stem is used as its replacement (Popović and Ney, 2004).

This paper presents a simpler method inspired by the Context-based MT approach (Carbonell et al., 2006) to improve translation quality. The method requires a large source language monolingual corpus and does not require any other language dependent resources to obtain replacements. Approaches suggested in the past only concentrated on finding replacements for the OOV words and not the rare words. This paper proposes a unified method to find possible replacements for OOV words as well as rare words based on the context in which these words appear. In the case of rare words, the translated sentence is traced back to find the origin of the translations and the target translations of the replacements are replaced with the translations of the rare words. In the case of OOV words, the target translations are replaced by the OOV word itself. The main idea for adopting this approach is the belief that the EBMT system will be able to find longer phrasal matches and that the language model will be able to give better probability estimates while decoding if it is not forced to fragment text at OOV and rare-word boundaries. This method is highly beneficial for low-resource languages that do not have morphological analysers or Part-of-Speech (POS) taggers and in cases where the similarity measures proposed in the past do not find closely related words for certain OOV words.

The rest of the paper is organized as follows. The next section (Section 2) discusses related work in handling OOV words. Section 3 describes the method adopted in this paper. Section 4 describes the experimental setup. Section 5 reports the results obtained with the new framework for English-Chinese and English-Haitian translation systems. Section 6 concludes and suggests possible future work.

2 Related Work

Orthographic and morpho-syntactic techniques for preprocessing training and test data have been shown to reduce OOV word rates. Popović and Ney (2004) demonstrated this on rich morphological languages in an SMT system. They introduced different types of transformations to the verbs to reduce the number of unseen word forms. Habash (2008) addresses spelling, name-transliteration OOVs and morphological OOVs in an Arabic-English Machine Translation system. Phrases with the OOV replacements in the phrase table of a phrase-based SMT system were "recycled" to create new phrases in which the replacements were replaced by the OOV words.

Yang and Kirchhoff (2006) proposed a back-off model for phrase-based SMT that translated word forms in the source language by hierarchical morphological phrase level abstractions. If an unknown word was found, the word was first stemmed and the phrase table entries for words sharing the same stem were modified by replacing the words with their stems. If a phrase entry or a single word phrase was found, the corresponding translation was used, otherwise the model backed off to the next level and applied compound splitting to the unknown word. The phrase table included phrasal entries based on full word forms as well as stemmed and split counterparts.

Vilar et al. (2007) performed the translation process treating both the source and target sentences as a string of letters. Hence, there are no unknown words when carrying out the actual translation of a test corpus. The word-based system did most of the translation work and the letter-based system translated the OOV words.

The method proposed in this work to handle OOV and rare words is very similar to the method adopted by Carbonell et al. (2006) to generate word and phrasal synonyms in their Context-based MT system. Context-based MT does not require parallel text but requires a large monolingual target language corpus and a fullform bilingual dictionary. The main principle is to find those n-gram candidate translations from a large target corpus that contain as many potential word and phrase translations of the source text from the dictionary and fewer spurious content words. The overlap decoder combines the target n-gram translation candidates by finding maximal left and right overlaps with the translation candidates of the previous and following n-grams. When the overlap decoder does not find coherent sequences of overlapping target n-grams, more candidate transla-

tions are obtained by substituting words or phrases in the target n-grams by their synonyms.

Barzilay and McKeown (2001) and Callison-Burch et al. (2006) extracted paraphrases from monolingual parallel corpus where multiple translations were present for the same source. The synonym generation in Carbonell et al. (2006) differs from the above in that it does not require parallel resources containing multiple translations for the same source language. In Carbonell et al. (2006), a list of paired left and right contexts that contain the desired word or phrase are extracted from the monolingual corpus. The same corpus is used to find other words and phrases that fit the paired contexts in the list. The idea is based on the distributional hypothesis which states that words with similar meanings tend to appear in similar contexts (Harris, 1954). Hence, their approach performed synonym generation on the target language to find translation candidates that would provide maximal overlap during decoding.

Marton et al. (2009) proposed an approach similar to Carbonell et al. (2006) to obtain replacements for OOV words, where monolingual distributional profiles for OOV words were constructed. Hence, the approach was applied on the source language side as opposed to Carbonell et al. (2006) which worked on the target language. Only similarity scores and no other features were used to rank the paraphrases (or replacements) that occured in similar contexts. The high ranking paraphrases were used to augment the phrase table of phrase-based SMT.

All of the previously suggested methods only handle OOV words (except Carbonell et al. (2006) which handles low frequency target phrases) and no attempt is made to handle rare words. Many of the methods explained above directly modify the training corpus (or phrase table in phrase-based SMT) increasing the size of the corpus. Our method clusters words and phrases based on their context as described by Carbonell et al. (2006) but uses the clustered words as replacements for not just the OOV words but also for the rare words on the source language side. Our method does not make use of any morphological analysers, POS taggers or manually created dictionaries as they may not be available for many rare or low-resource languages. The translation of the replacements in the final decoded target sentence is replaced by the translation of the original word (or the source word itself in the OOV case), hence, we do not specifically look for synonyms. The only condition for a word to be a candidate replacement is that its left and right context need to match with that of the OOV/rare-word. Hence, the clustered words could have different semantic relations. For example,

$(cluster1)$:"laugh, giggle, chuckle, cry, weep"
where "laugh, giggle, chuckle" are synonyms and "cry, weep" are antonyms of "laugh".

Clusters can also contain hypernyms (or hyponyms), meronyms (or holonyms), troponyms and coordinate terms along with synonyms and antonyms. For example,

$(cluster2)$:"country, region, place, area, district, state, zone, United States, Canada, Korea, Malaysia".
where "country" is a hypernym of "United States/Canada/Korea/Malaysia". "district" is a meronym of "state". "United States, Canada, Korea, Malaysia" are coordinate terms sharing "country" as their hypernym.

The contributions made by the paper are threefold: first, replacements are found for not just the OOV words but for the *rare* words as well. Second, the framework used allows scoring replacements based on multiple features to permit optimization. Third, instead of directly modifying the training corpus by replacing the candidate replacements by the OOV words, a new representation scheme is used for the test sentences to efficiently handle a beam of possible replacements.

3 Proposed Method

Like Marton et al. (2009), only a large monolingual corpus is required to extract candidate replacements. To retrieve more replacements, the monolingual corpus is pre-processed by first generalizing numbers, months and years by NUMBER, MONTH and YEAR tags, respectively.

3.1 OOV and Rare words

Words in the test sentence (new source sentence to be translated) that do not appear in the training corpus are called OOV words. Words in the test sentence that appear less than K times in the training corpus are considered as rare words (in this paper $K = 3$). The method presented in the following sections holds for both OOV as well as rare words. In the case of rare words, the final translation is postprocessed (Section 3.7) to include the translation of the rare word.

The procedure adopted will be explained with a real example T (the rest of the sentence is removed for the sake of clarity) encountered in the test data with "hawks" as the OOV word,

T:a mobile base , *hitting three* **hawks** *with one arrow* over the past few years ...

3.2 Context

As the goal is to obtain longer target phrasal translations for the $test\ sentence$ before decoding, only words that fit the left and right context of the OOV/rare-word in the test sentence are extracted. Unlike Marton et al. (2009) where a context list for each OOV is generated from the contexts of their replacements, this paper uses only the left and right context of the OOV/rare-word. The default window size for the context is five words (two words to the left and two words to the right of the OOV/rare-word). If the windowed words contain only function words, the window is incremented until at least one content word is present in the resulting context. This enables one to find sensible replacements that fit the context well. The contexts for T are:

Left-context (L): hitting three
Right-context (R): with one arrow

The above contexts are further processed to generalize the numbers by a $NUMBER$ tag to produce more candidate replacements. The resulting contexts are now:

Left-context (L): hitting $NUMBER$
Right-context (R): with $NUMBER$ arrow

As a single $L - R$ context is used, a far smaller number of replacements are extracted.

3.3 Finding Candidate replacements

The monolingual corpus (ML) of the source language is used to find words and phrases (X_k) that fit LX_kR i.e., with L as its left context and/or R as its right context. The maximum length for X_k is set to 3 currently. The replacements are further filtered to obtain only those replacements that contain at least one content word. As illustrated earlier, the resulting replacement candidates are not necessarily synonyms.

3.4 Features

A local context of two to three words to the left of an OOV/rare-word ($word_i$) and two to three words to the right of $word_i$ contain sufficient clues for the word, $word_i$. Hence, local contextual features are used to score each of the replacement candidates ($X_{i,k}$) of $word_i$. Each $X_{i,k}$ extracted in the previous step is converted to a feature vector containing 11 contextual features. Certainly more features can be extracted with additional knowledge sources. The framework allows adding more features, but for the present results, only these 11 features were used.

As our aim is to assist the translation system in finding longer target phrasal matches, the features are constructed from the occurrence statistics of $X_{i,k}$ from the bilingual training corpus (BL). If a candidate replacement does not occur in the BL, then it is removed from the list of possible replacement candidates.

Frequency counts for the features of a particular replacement, $X_{i,k}$, extracted in the context of $L_{i,-2}L_{i,-1}$ (two preceding words of $word_i$) and $R_{i,+1}R_{i,+2}$ (two following words of $word_i$) (the remaining words in the left and right context of $word_i$ are not used for feature extraction) are obtained as follows:

f_1: frequency of $X_{i,k}R_{i,+1}$
f_2: frequency of $L_{i,-1}X_{i,k}$
f_3: frequency of $L_{i,-1}X_{i,k}R_{i,+1}$
f_4: frequency of $L_{i,-2}L_{i,-1}X_{i,k}$
f_5: frequency of $X_{i,k}R_{i,+1}R_{i,+2}$
f_6: frequency of $L_{i,-2}L_{i,-1}X_{i,k}R_{i,+1}$

f_7: frequency of $L_{i,-1}X_{i,k}R_{i,+1}R_{i,+2}$
f_8: frequency of $L_{i,-2}L_{i,-1}X_{i,k}R_{i,+1}R_{i,+2}$
f_9: frequency of $X_{i,k}$ in ML
f_{10}: frequency of $X_{i,k}$ in BL
f_{11}: number of feature values $(f_1, ..f_{10}) > 0$

f_{11} is a vote feature which counts the number of features (f_1 ... f_{10}) that have a value greater than zero. The features are normalized to fall within $[0, 1]$. The sentences in ML, BL and test data are padded with two begin markers and two end markers for obtaining counts for OOV/rare-words that appear at the beginning or end of a test sentence.

3.5 Representation

Before we go on to explaining the lattice representation, we would like to make a small clarification in the terminalogy used. In the MT community, a lattice usually refers to the list of possible partially-overlapping target translations for each possible source $n-$gram phrase in the input sentence. Since we are using the term lattice to also refer to the possible paths through the input sentence, we will call the lattice used by the decoder, the "*decoding lattice*". The lattice obtained from the input sentence representing possible replacement candidates will be called the "*input lattice*".

An input lattice (Figure 1) is constructed with a beam of replacements for the OOV and rare words. Each replacement candidate is given a score (Eqn 1) indicating the confidence that a suitable replacement is found. The numbers in Figure 1 indicate the start and end indices (based on character counts) of the words in the test sentence. In T, two replacements were found for the word *"hawks"*: *"homers"* and *"birds"*. However, "homers" was not found in the BL and hence, it was removed from the replacement list.

The input lattice also includes the OOV word with a low score (Eqn 2). This allows the EBMT system to also include the OOV/rare-word during decoding. In the Translation Model of the EBMT system, this test lattice is matched against the source sentences in the bilingual training corpus. The matching process would now also look for phrases with "birds" and not just "hawks". When a match is found, the corresponding translation in the target language is obtained through sub-sentential alignment (Section 3.7). The scores on the input lattice are later used by the decoder (Section 3.7). Each replacement $X_{i,k}$ for the OOV/rare-word ($word_i$) is scored with a logistic function (Bishop, 2006) to convert the dot product of the features and weights ($\vec{\lambda} \cdot \vec{f_{i,k}}$) to a score between 0 and 1 (Eqn 1 and Eqn 2).

T : a mobile base , hitting three hawks with one arrow .....

input lattice:

0	0	(" a ")
1	6	(" mobile ")
7	10	(" base ")
11	11	(",")
12	18	(" hitting ")
13	17	(" three ")
18	22	(" *hawks* " 0.0026)
18	22	(" **birds** " 0.9974)
23	26	(" with ")
27	29	(" one ")
30	34	(" arrow ")

Figure 1: Lattice of the input sentence T containing replacements for OOV words.

OOV/Rare word	Candidate Replacements
Spelling errors	
krygyzstan	kyrgyzstan,...
yusukuni	yasukuni,..
kilomaters	kilometers, miles, km, ...
	Coordinate terms
somoa	india, turkey, germany, russia, japan,...
ear	body, arms, hands, feet, mind, car, ...
buyers	dealer, inspector, the experts, smuggler,.
	Synonyms
plummet	drop, dropped, fell, ...
	Synonyms and Antonyms
optimal	worse, better, minimal,....

Figure 2: Sample English candidate replacements obtained.

$$p_\lambda(X_{i,k}|word_i) = \frac{exp(\vec{\lambda}\vec{f_{i,k}})}{1+\sum_{j=1...S} exp(\vec{\lambda}\vec{f_{i,j}})} \quad (1)$$

$$p_\lambda(word_i) = \frac{1}{1 + \sum_{j=1...S} exp(\vec{\lambda} \cdot \vec{f_{i,j}})} \quad (2)$$

where, $\vec{f_{i,j}}$ is the feature vector for the j^{th} replacement candidate of $word_i$, S is the number of replacements, $\vec{\lambda}$ is the weight vector indicating the importance of the corresponding features.

3.6 Tuning feature weights

We would like to select those feature weights ($\vec{\lambda}$) which would lead to the least expected loss in translation quality (Eqn 3). $-log(BLEU)$ (Papineni et al., 2002) is used to calculate the expected loss over a development set. As this objective function has many local minima and is piecewise constant, the surface is smoothed using the L2-norm regularization. Powell's algorithm (Powell, 1964) with grid-based line optimization is used to find the best weights. 7 different random guesses are used to initialize the algorithm.

$$\min_{\lambda} E_\lambda[L(t_{tune})] + \tau * ||\lambda||^2 \quad (3)$$

The algorithm assumes that partial derivates of the function are not available. Approximations of the weights ($\lambda_1, ..\lambda_N$) are generated successively along each of the N standard base vectors. The procedure is iterated with a stopping criteria based on the amount of change in the weights and the change in the loss. A cross-validation set (in addition to the regularization term) is used to prevent overfitting at the end of each iteration of the Powell's algorithm. This process is repeated with different values of τ, as in Deterministic Annealing (Rose, 1998). τ is initialized with a high value and is halved after each process.

3.7 System Description

The EBMT system finds phrasal matches for the test (or input) sentence from the source side of the bilingual corpus. The corresponding target phrasal translations are obtained through subsentential alignment. When an *input lattice* is given instead of an input sentence, the system performs the same matching process for all possible phrases obtained from the input lattice. Hence, the system also finds matches for source phrases that contain the replacements for the OOV/rare-word. Only the top C ranking replacement candidates for every OOV/rare word are used in building the input lattice. The optimal value of C was empirically found to be 2. On examining the obtained input lattices, the proposed method found replacements for at the most 3 OOV/rare words in each test sentence (Section 4). Hence, the number of possible paths through the input lattice is not substantially large.

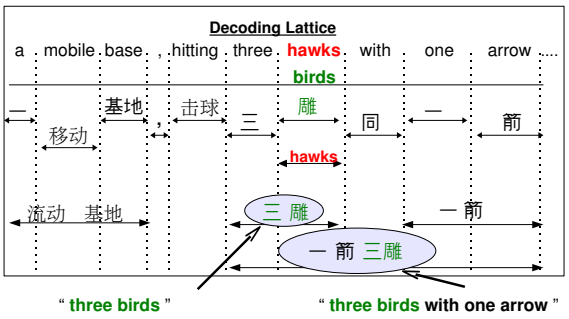


Figure 3: Lattice containing possible phrasal target translations for the test sentence T.

The target translations of all the source phrases are placed on a common decoding lattice. An example of a decoding lattice for example T is given in Figure 3. The system is now able to find longer matches (*" three birds with one arrow "* and *" three birds "*) which was not possible earlier with the OOV word, *"hawks"*. The local ordering information between the translations of "three birds" and "with one arrow" is well captured due to the retrieval of the longer source phrasal match, "three birds with one arrow". Our ultimate goal is to obtain translations for such longer $n-$gram source phrases boosting the confidence of both the translation model and the language model.

The decoder used in this paper (Brown, 2003) works on this *decoding lattice* of possible phrasal target translations (or fragments) for source phrases present in the *input lattice* to generate the target translation. Similar to Pharaoh (Koehn et al., 2003), the decoder uses multi-level beam search with a priority queue formed based on the number of source words translated. Bonuses are given for paths that have overlapping fragments. The total score (TS) for a path (Eqn 4) through the translation lattice is the arithmetic average of the scores for each target word in the

path. The EBMT engine assigns each candidate phrasal translation a quality score computed as a log-linear combination of alignment score and translation probability. The alignment score indicates the engine's confidence that the right target translation has been chosen for a source phrase. The translation probability is the proportion of times each distinct alternative translation was encountered out of all the translations. If the path includes a candidate replacement, the log of the score, $p_\lambda(w_i)$, given for a candidate replacement is incorporated into TS as an additional term with a weight wt_5.

$$TS = \frac{1}{t}\sum_{i=1}^{t}[wt_1 \log(b_i) + wt_2 \log(pen_i) + wt_3 \log(q_i) + wt_4 \log(P(w_i|w_{i-2}, w_{i-1})) + \mathbb{1}_{(w_i=replacement)} wt_5 \log(p_\lambda(w_i)) \ ] \quad (4)$$

where, t is the number of target words in the path, wt_j indicates the importance of each score, b_i is the bonus factor given for long phrasal matches, pen_i is the penalty factor for source and target phrasal-length mismatches, q_i is the quality score and $P(w_i|w_{i-2}, w_{i-1})$ is the LM score. The parameters of the EBMT system (wt_j) are tuned on a development set.

The target translation is postprocessed to include the translation of the OOV/rare-word with the help of the best path information from the decoder. In the case of OOV words, since the translation is not available, the OOV word is put back into the final output translation in place of the translation of its replacement. In the output translation of the test example T, the translation of *"birds"* is replaced by the word, *"hawks"*. For rare words, knowing that the translation of the rare word may not be correct (due to poor alignment statistics), the target translation of the replacement is replaced by the translation of the rare word obtained from the dictionary. If the rare word has multiple translations, the translation with the highest score is chosen.

4 Experimental Setup

As we are interested in improving the performance of low-resource EBMT, the English-Haitian (Eng-Hai) newswire data (Haitian Creole, CMU, 2010) containing 15,136 sentence-pairs was used. To test the performance in other languages, we simulated sparsity by choosing less training data for English-Chinese (Eng-Chi). For the Eng-Chi experiments, we extracted 30k training sentence pairs from the FBIS (NIST, 2003) corpus. The data was segmented using the Stanford segmenter (Tseng et al., 2005). Although we are only interested in small data sets, we also performed experiments with a larger data set of 200k. 5-gram Language Models were built from the target half of the training data with Kneser-Ney smoothing. For the monolingual English corpus, 9 million sentences were collected from the Hansard Corpus (LDC, 1997) and FBIS data.

EBMT system without OOV/rare-word handling is chosen as the Baseline system. The parameters of the EBMT system are tuned with 200 sentence pairs for both Eng-Chi and Eng-Hai. The tuned EBMT parameters are used for the Baseline system and the system with OOV/rare-word handling. The feature weights for the proposed method are then tuned on a seperate development set of 200 sentence-pairs with source sentences containing at least 1 OOV/rare-word. The cross-validation set for this purpose is made up of 100 sentence-pairs. In the OOV case, 500 sentence pairs containing at least 1 OOV word are used for testing. For the rare word handling experiments, 500 sentence pairs containing at least 1 rare word are used for testing.

To assess the translation quality, 4-gram word-based BLEU is used for Eng-Hai and 3-gram word-based BLEU is used for Eng-Chi. Since BLEU scores have a few limitations, the NIST and TER metrics are also used. The test data used for comparing the system handling OOV words and the Baseline (without OOV word handling) is different from the test data used for comparing the system handling rare words and the Baseline system (without rare word handling). In the former case, the test data handles only OOV words and in the latter, the test data only handles rare words. Hence, the test data for both the cases do not completely overlap. As we are interested in determining whether handling rare words in test sentences is useful, we keep both the test data sets seperate and assess the improvements obtained by only

OOV/Rare	system	TER	BLEU	NIST
OOV	Baseline	77.89	18.61	4.8525
	Handling OOV	76.95	19.32	4.9664
Rare	Baseline	74.23	22.84	5.3803
	Handling Rare	74.02	23.12	5.4406

Table 1: Comparison of translation scores of the Baseline system and system handling OOV and Rare words for Eng-Hai.

handling OOV words and by only handling rare words over their corresponding Baselines. As future work, it would be interesting to create one test data set to handle both OOV and rare words to see the overall gain.

The test set is further split into 5 files and the Wilcoxon (Wilcoxon, 1945) Signed-Rank test is used to find the statistical significance.

5 Results

Sample replacements found are given in Figure 2. For both Eng-Chi and Eng-Hai experiments, only the top C ranking replacement candidates were used. The value of C was tuned on the development set and the optimal value was found to be 2. Translation quality scores obtained on the test data with 30k and 200k Eng-Chi training data sets are given in Table 2. Table 1 shows the results obtained on Eng-Hai. Statistically significant improvements ($p < 0.0001$) were seen by handling OOV words as well as rare words over their corresponding baselines.

As the goal of the approach was to obtain longer target phrasal matches, we counted the number of n-grams for each value of n present on the decoding lattice in the 30k Eng-Chi case. The subplots: A and B in Figure 4, shows the frequency of n-grams for higher values of n (for $n > 5$) when handling OOV and rare words. The plots clearly show the increase in number of longer target phrases when compared to the phrases obtained by the baseline systems.

Since the BLEU and NIST scores were computed only up to 3-grams, we further found the number of n-gram matches (for $n > 3$) in the final translation of the test data with respect to the reference translations (subplots: C and D). As expected, a larger number of longer $n-$gram matches were found. For the OOV case, matches

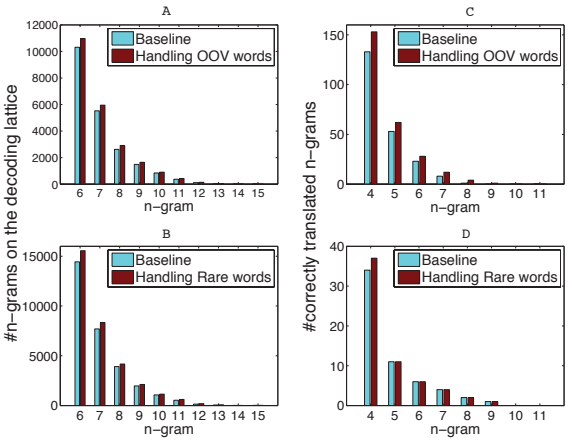


Figure 4: A, B: number of n-grams found for increasing values of n on the decoding lattice. C, D: number of target n-gram matches for increasing values of n with respect to the reference translations.

OOV/Rare	Training data size	system	TER	BLEU	NIST
OOV	30k	Baseline	82.03	14.12	4.1186
	30k	Handling OOV	80.97	14.78	4.1798
	200k	Baseline	79.41	19.90	4.6822
	200k	Handling OOV	77.66	20.50	4.7654
Rare	30k	Baseline	82.09	15.36	4.3626
	30k	Handling Rare	80.02	16.03	4.4314
	200k	Baseline	78.04	20.96	4.9647
	200k	Handling Rare	77.35	21.17	5.0122

Table 2: Comparison of translation scores of the Baseline system and system handling OOV and Rare words for Eng-Chi.

up to 9-grams were found where the baseline only found matches up to 8-grams.

6 Conclusion and Future Work

A simple approach to improve translation quality by handling both OOV and rare words was proposed. The framework allowed scoring and ranking each replacement candidate efficiently.

The method was tested on two language pairs and statistically significant improvements were seen in both cases. The results showed that rare words also need to be handled to see improvements in translation quality.

In this paper, the proposed method was only applied on words, as future work we would like to extend it to OOV and rare-phrases as well.

References

R. Barzilay and K. McKeown 2001. Extracting paraphrases from a parallel corpus. In *Proceedings of the 39th Annual Meeting of the Association for Computaional Linguistics,* pp. 50-57.

C. M. Bishop 2006. Pattern Recognition and Machine Learning, Springer.

R. D. Brown, R. Hutchinson, P. N. Bennett, J. G. Carbonell, P. Jansen. 2003. Reducing Boundary Friction Using Translation-Fragment Overlap. In *Proceedings of The Ninth Machine Translation Summit,* pp. 24-31.

R. D. Brown. 2000. Automated Generalization of Translation Examples. In *Proceedings of The International Conference on Computational Linguistics,* pp. 125-131.

C. Callison-Burch, P. Koehn and M. Osborne. 2006. Improved Statistical Machine Translation Using Paraphrases. In *Proceedings of The North American Chapter of the Association for Computational Linguistics,* pp. 17-24.

J. Carbonell, S. Klien, D. Miller, M. Steinbaum, T. Grassiany and J. Frey. 2006. Context-Based Machine Translation Using Paraphrases. In *Proceedings of The Association for Machine Translation in the Americas,* pp. 8-12.

N. Habash. 2008. Four Techniques for Online Handling of Out-of-Vocabulary Words in Arabic-English Statistical Machine Translation. In *Proceedings of Association for Computational Linguistics-08: HLT,* pp. 57-60.

Public release of Haitian Creole language data by Carnegie Mellon, 2010. *http://www.speech.cs.cmu.edu/haitian/*

Z. Harris. 1954. Distributional structure. *Word,* 10(23): 146-162.

P. Koehn. 2004. Pharaoh: a Beam Search Decoder for Phrase-Based Statistical Machine Translation Models. *The Association for Machine Translation*.

P. Koehn, F. J. Och and D. Marcu. 2003. Statistical Phrase-Based Translation. In *Proceedings of HLT:The North American Chapter of the Association for Computational Linguistics*.

P. Koehn 2002 Europarl: A multilingual corpus for evaluation of machine translation. Unpublished, *http://www.isi.edu/koehn/publications/europarl/*

Linguistic Data Consortium. 1997 Hansard Corpus of Parallel English and French. Linguistic Data Consortium, December. *http://www.ldc.upenn.edu/*

Y. Marton, C. Callison-Burch and P. Resnik. 2009. Improved Statistical Machine Translation Using Monolingually-derived Paraphrases. In *Proceeding of The Empirical Methods in Natural Language Processing,* pp. 381-390.

NIST. 2003. Machine translation evaluation. *http://nist.gov/speech/tests/mt/*

K. Papineni, S. Roukos, T. Ward, and W. Zhu. 2002. BLEU: A Method for Automatic Evaluation of Machine Translation. In *Proceedings of The Association for Computational Linguistics*. pp. 311-318.

M. Popović and H. Ney. 2004. Towards the use of Word Stems and Suffixes for Statistical Machine Translation. In *Proceedings of The International Conference on Language Resources and Evaluation*.

M. J. D. Powell. 1964. An efficient method for finding the minimum of a function of several variables without calculating derivatives *Computer Journal*. Volume 7, pp. 152-162.

K. Rose. 1998. Deterministic annealing for clustering, compression, classification, regression, and related optimization problems. In *Proceedings of The Institute of Electrical and Electronics Engineers,* pp. 2210-2239.

H. Tseng, P. Chang, G. Andrew, D. Jurafsky and C. Manning. 2005. A Conditional Random Field Word Segmenter. *Fourth SIGHAN Workshop on Chinese Language Processing*.

D. Vilar, J. Peter, and H. Ney. 2007. Can we translate letters? In *Proceedings of Association Computational Linguistics Workshop on SMT,* pp. 33-39.

M. Yang and K. Kirchhoff. 2006. Phrase-based back-off models for machine translation of highly inflected languages. In *Proceedings of European Chapter of the ACL,* 41-48.

F. Wilcoxon. 1945. Individual comparisons by ranking methods. *Biometrics, 1,* 80-83. tool: *http://faculty.vassar.edu/lowry/wilcoxon.html*

Utilizing User-input Contextual Terms for Query Disambiguation

Byron J. Gao
Texas State University
bgao@txstate.edu

David C. Anastasiu
Texas State University
da1143@txstate.edu

Xing Jiang
Nanyang Technological University
jian0008@ntu.edu.sg

Abstract

Precision-oriented search results such as those typically returned by the major search engines are vulnerable to issues of polysemy. When the same term refers to different things, the dominant sense is preferred in the rankings of search results. In this paper, we propose a novel technique in the context of web search that utilizes contextual terms provided by users for query disambiguation, making it possible to prefer other senses without altering the original query.

1 Introduction

World Wide Web and search engines have become an indispensable part of everyone's everyday life. While web search has come a long way over the past 10 years, it still has a long way to go to respond to the ever-increasing size of the web and needs of web surfers. Today, web search is under intensive and active research, drawing unparalleled attention from both industry and academia.

Need of disambiguation. One of the major challenges in web search lies in unsatisfactory relevance of results caused by ambiguity. Query terms are inherently ambiguous due to polysemy, and most queries are short containing 1 to 3 terms only (Jansen et al., 2000). Thus queries are in general prone to ambiguity of user intent or information needs, resulting in retrieval of enormous irrelevant pages. As the web increases in size at an increasing rate, ambiguity becomes ubiquitous and users are in increasing need of effective means of disambiguation. The ambiguity issue and its consequences are demonstrated in Example 1.

Example 1 *There are 17 entries in Wikipedia for different renown individuals under the same name of "Jim Gray", including a computer scientist, a sportscaster, a zoologist, a politician, a film director, a cricketer, and so on. Suppose we intend to find information about* `Jim Gray`*, the Turing award winner, we can issue a query of "Jim Gray" in Yahoo!For this extremely famous name in computer science, only 3 are relevant in the top 10 results. They are his Wikipedia entry, homepage at Microsoft Research, and DBLP entry.*

Straightforward approach. One intuitive way of disambiguation would be to apply available domain knowledge and refine the query by adding some confining contextual terms. This would generally improve precision. However, there are several inevitable problems in this approach. First, the improvement on precision is at the sacrifice of recall. For example, many `Jim Gray` pages may not contain the added contextual terms and are thus excluded from the search results.

Second, the query is altered, leading to unfavorable ranking of results. Term proximity matters significantly in ranking (Manning et al., 2008). Some good pages w.r.t. the original query may be ranked low in the new search results because of worsened term proximity and relevance w.r.t. the new query. Thus, with this straightforward approach only limited success can be expected at best, as demonstrated in Example 2.

Example 2 *Suppose we know that* `Jim Gray` *is a computer scientist, we can issue a query of "Jim Gray computer". All the top 10 results are about* `Jim Gray` *and relevant. However, many of them are trivial pages, failing to include 2 of the 3 most important ones. His DBLP entry appears as the*

Coling 2008: Poster Volume, pages 329–337,
Beijing, August 2010

27th result, and his homepage at Microsoft Research appears as the 51st result.

This limited success is achieved by using a carefully selected contextual term. "Computer" is a very general term appearing on most of the `Jim Gray` pages. Also, there are no other competitively known computer people with the same name. Most other contextual terms would perform much worse. Thus a third problem of this straightforward query refinement approach is that only few contextual terms, which may not be available to users, would possibly achieve the limited success. Often, much of our domain knowledge would cause more damage than repair and is practically unusable, as demonstrated in Example 3.

Example 3 *Suppose we know that* `Jim Gray` *has David DeWitt as a close friend and colleague, we can issue a query of "Jim Gray David DeWitt". Again, all the top 10 results are about* `Jim Gray` *and relevant. However, the theme of the query is almost completely altered. Evidently, the 1st result "Database Pioneer Joins Microsoft to Start New Database Research Lab", among many others, talks about David DeWitt. It is relevant to* `Jim Gray` *only because the lab is named "Jim Gray Systems Lab" in honor of him.*

The `Bobo` approach. Can we freely apply our domain knowledge to effectively disambiguate search intent and improve relevance of results without altering the original query? For this purpose, we propose and implement `Bobo`.[1]

For conceptual clarity, the `Bobo` interface features two boxes. Besides a regular query box, an additional box is used to take *contextual terms* from users that capture helpful domain knowledge. Contextual terms are used for disambiguation purposes. They do not alter the original query defined by query terms. Particularly, unlike in the straightforward approach, positive contextual terms are not required to be included in search results and negative contextual terms are not required to be excluded from search results. Contextual terms help estimate relevance of search results, routing them towards a user intended domain, filtering out those not-in-domain, or irrelevant, results.

`Bobo` works in two rounds. In round I, a query is issued using by default the combination of query terms and contextual terms, or just the contextual terms if the query returns too few results. Then from the results, some top-ranked high-quality pages are (automatically) selected as *seeds*. In round II, a query is issued using the query terms. Then the results are compared with the seeds and their similarities are computed. The similarity values reflect the degree of relevance of search results to the user intent, based on which the results are re-ranked.

Example 4 reports the `Bobo` experiment using the same contextual terms as in Example 3.

Example 4 *As in Example 3, suppose we know* `Jim Gray` *has David DeWitt as a colleague. Then with* `Bobo`*, we can enter "Jim Gray" in the query box and "David DeWitt" in the auxiliary box. As a result with default preferences, all the top 10 results are relevant including all the top 3 important* `Jim Gray` *pages. From the top 10, only 1 page, the DBLP entry, contains "David DeWitt" as they coauthored papers. The theme of the query is not altered whereas in Example 3, all the top 10 results contain "David DeWitt".*

In Example 4, the selected seeds are relevant to `Jim Gray`. Observe that seeds can be useful if they are relevant to the user-intended domain, not only the user-intended query. `Bobo` works effectively with such seeds and thus can utilize a much expanded range of domain knowledge. Helpful contextual terms do not even need to co-occur with query terms on any page. They only need to occur, possibly separately, on some pages of the same domain, as demonstrated in Example 5.

Example 5 *Using the criteria of being in the same community as* `Jim Gray` *but co-occuring on no web pages, we randomly chose a student name, Flavia Moser. In* `Bobo`*, we entered "Jim Gray" in the query box, "Flavia Moser" in the auxiliary box, and used only the contextual terms for the round I query. As a result, 11 of the top 12 results were relevant including all the top 3 important* `Jim Gray` *pages. Of course, none of the returned pages contains "Flavia Moser".*

[1] `Bobo` has been implemented using Yahoo! web search API and maintained at http://dmlab.cs.txstate.edu/bobo/.

2 Related Work

Disambiguating search intent, capturing information needs and improving search performance have been a fundamental research objective in information retrieval and studied from different perspectives. Voorhees (1993) shows that disambiguation cannot be easily resolved using thesauruses. The filtering problem (Baeza-Yates and Ribeiro-Neto, 1999; Schapire et al., 1998) views disambiguation as a binary text classification task assigning documents into one of the two categories, relevant and irrelevant. The routing problem (Schutze et al., 1995; Singhal et al., 1997) differs from text classification in that search results need to be ranked instead of just classified (Gkanogiannis and Kalamboukis, 2008).

Contextual search (Lawrence, 2000; Finkelstein et al., 2002; Kraft et al., 2006), personalized search (Haveliwala, 2002; Teevan et al., 2005; Zhu et al., 2008), and implicit relevance feedback (Kelly and Teevan, 2003; Joachims et al., 2005; White et al., 2004) generally utilize long-term search history to build user profiles. These profiles are used on a regular basis to guide *many* queries. Such approaches entail little extra user involvement in search, but need to manage profiles, face the privacy issue, and swallow the inflexibility in context switch.

Explicit and pseudo relevance feedback (RF) techniques (Ruthven and Lalmas, 2003; Baeza-Yates and Ribeiro-Neto, 1999; Manning et al., 2008) are more related to Bobo in the sense that they do not build long-term profiles. Instead, they construct a one-time search context that are used only once to guide a *single* query each time. Such approaches enjoy the flexibility to switch spontaneously from one domain to another in response to different information needs.

RF is regarded as the most popular query reformation strategy (Baeza-Yates and Ribeiro-Neto, 1999). It iterates in multiple rounds, typically two, to modify a query step by step. Explicit RF asks explicit feedback from users, whereas pseudo (or blind) RF assumes relevance of top-ranked results. The problem of explicit RF is that it requires too much user involvement. Users are often reluctant to provide explicit feedback, or do not wish to prolong the search interaction. Web search engines of today do not provide this facility. Excite.com initially included but dropped it due to the lack of use (Manning et al., 2008).

Pseudo RF, first suggested by Croft and Harper (1979) and since widely investigated, automates the manual part of RF, so that users get improved search performance without extended interactions. Psuedo RF has been found to improve performance in the TREC ad hoc task and Cornell SMART system at TREC 4 (Buckley et al., 1995). Unfortunately, pseudo RF suffers from a major flaw, the so-called *query drift* problem. Query drift occurs when the feedback documents contain few or no relevant ones. In this case, search results will be routed farther away from the search intent, resulting in even worse performance. Different approaches (Mitra et al., 1998; Yu et al., 2003; Lee et al., 2008)have been proposed to alleviate query drift but with little success. Some queries will be improved, others will be harmed (Ruthven and Lalmas, 2003).

Similarly to RF, Bobo works in two rounds. Similarly to pseudo RF, it makes use of top-ranked round I results. However, Bobo and RF differ fundamentally in various aspects.

Firstly, Bobo is not a query reformation technique as RF. In RF, the *automatically generated* additional terms become part of the reformed query to be issued in round II, while in Bobo, the *user-input* contextual terms are *not* used in round II. The terms generated by RF may work well as contextual terms for Bobo but not the other way around. In general, effective contextual terms form a much larger set.

In query reformation, it is often hard to understand why a particular document was retrieved after applying the technique (Manning et al., 2008). In Bobo, the original query is kept intact and only the ranking of search results is changed.

Secondly, in RF, only query terms are used in round I queries. In Bobo, by default the combination of query terms and contextual terms, both entered by users, is used, leading to much more relevant seeds that are comparable to explicit RF. In this sense, Bobo provides a novel and effective remedy for query drift.

Beyond that, Bobo can use contextual terms

only to obtain seeds that are relevant to the user-intended domain and not necessarily the user-intended query, leading to effective utilization of a largely expanded range of domain knowledge.

Thirdly, RF can have practical problems. The typically long queries (usually more than 20 terms) generated by RF techniques are inefficient for IR systems, resulting in high computing cost and long response time (Manning et al., 2008). In Bobo, however, both query terms (1 to 3) and contextual terms (1 to 2) are short. A round I query combining the two would typically contain 2 to 5 terms only.

3 Overview

Bobo uses the vector space model, where both documents and queries are represented as vectors in a discretized vector space. Documents used in similarity comparison can be in the form of either full pages or snippets. Documents are preprocessed and transformed into vectors based on a chosen term weighting scheme, e.g., TF-IDF.

The architecture of Bobo is shown in Figure 1. Without input of contextual terms, Bobo works exactly like a mainstream search engine and the dashed modules will not be executed. Input of contextual terms is optional in need of disambiguation of user intent. Domain knowledge, directly or indirectly associated with the query, can be used as "pilot light" to guide the search towards a user-intended domain.

With input of contextual terms, Bobo works in two rounds. In round I, a query is issued using by default the combination of query terms and contextual terms, or just the contextual terms if they are unlikely to co-occur much with the query terms. Then from the results, the top k documents (full pages or snippets) satisfying certain quality conditions, e.g., number of terms contained in each seed, are selected as seeds. Optionally, seeds can be *cleaned* by removing the contained query terms to reduce background noise of individual seeds, or *purified* by removing possibly irrelevant seeds to improve overall concentration. Contextual terms themselves can be used as an *elf seed*, which is a special document allowing negative terms, functioning as an explicit feedback.

In round II, a query is issued using the query

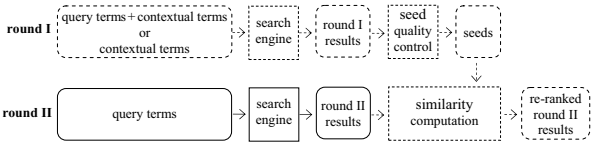


Figure 1: Architecture of Bobo.

terms. Then, each returned result (full page or snippet) is compared to the seeds to compute a similarity using a designated similarity measure, Jaccard coefficient or Cosine coefficient. In the computation, seeds can be *combined* to form a prototype as in Rocchio, or not combined using none generalization as in instance-based lazy learning to better capture locality and handle polymorphic domains. Based on the assumption that seeds are highly relevant, the similarity values estimate the closeness of search results to the user intent, based on which the results are re-ranked.

Bobo was implemented using Yahoo! web search API. For each query, Bobo retrieves 30 HTML pages from the API. If snippets are used for seeding and comparison, the response time of Bobo is sufficiently fast. If full pages are used, page downloading and preprocessing are prohibitively time-consuming. However, the goal of Bobo is to illustrate the promise of the novel disambiguation approach. If Bobo were implemented at the server (search engine) side, response time would not be an issue.

4 Principles and Preferences

In this section, we introduce in detail the design principles and preferences of Bobo regarding the various key issues. We also discuss possible improvements in these aspects.

4.1 Use of Contextual Terms

How to use contextual terms has a fundamental impact on the behavior and performance of Bobo.

In round I. By default, the combination of query terms and contextual terms are used in round I queries. This produces seeds that are relevant to the user-intended query. For instance, in Example 4, the seeds are relevant to Jim Gray. This usage of contextual terms actually provides a novel and effective remedy for query drift, thanks to the input of domain knowledge.

Generally, a large portion of domain knowledge cannot be utilized in a straightforward manner, due to the fact that contextual terms may co-occur with query terms in very few or none web pages. However, as shown in Example 5, Bobo allows using only contextual terms for round I queries, enabling utilization of indirectly associated domain knowledge.

As elf seed. Contextual terms can be considered forming a pseudo document, which can be optionally used as a seed. We call such a seed *elf seed* as it is actually a piece of explicit relevance feedback. Unlike normal seeds, an elf seed may contain positive as well as negative terms, providing a way of collecting positive as well as negative explicit feedback.

Discussion. The option of combing query terms and contextual terms in round I queries can be automated. The idea is to combine the terms first, then test the k^{th} result to see whether it contains all the terms. If not, only the contextual terms should be used in the query.

4.2 Quality of Seeds

As in pseudo relevance feedback, quality of seeds plays an critical role in search performance. The difference is that in Bobo, input of contextual terms is largely responsible for the much improved relevance of seeds. To provide further quality control, Bobo accepts several user-input thresholds, e.g., number of seeds and number of terms contained in each seed. Beyond that, Bobo also provides the following options.

Removing query terms. By default, Bobo uses a combination of contextual terms and query terms in round I queries. Thus usually all the seeds contain the query terms. Round II results contain the query terms as well. Then, in similarity computation against the seeds, those query terms contribute almost equally to each round II result. This amount of contribution then becomes background noise, reducing the sensitivity in differentiating round II results.

By default, Bobo removes query terms from seeds. Although a simple step, this option significantly improves performance in our experiments.

Purifying seeds. Different approaches have been proposed to alleviate query drift by improving relevance of pseudo feedback, but with limited success (Ruthven and Lalmas, 2003). In Bobo, due to the input of domain knowledge, we can well assume that the majority of seeds are relevant, based on which, we can design simple mechanisms to purify seeds. Briefly, we first calculate the centroid of seeds. Then, we compute the similarity of each seed against the centroid, and remove those outlying seeds with poor similarities.

Discussion. Current search engines take into account link-based popularity scores in ranking search results. In Bobo, round I search results are not used to directly meet information needs of users. They are never browsed by users. Thus, different search engines with alternative ranking schemes may be used to better fulfill the purpose of round I queries.

Round I queries do not need to be issued to the same region as round II queries either. Working in a more quality region may help avoid spamming and retrieve better candidates for seed selection.

4.3 Term Weighting

Bobo uses two term weighting schemes. The default one is the conventional TF-IDF. The other scheme, TF-IDF-TAI, uses term association to favor terms that show high co-occurrence with query terms. It is tailored to Bobo, where documents are not compared in isolation, but being "watched" by a query. While TF-IDF can be considered global weighting independent of queries, TF-IDF-TAI can be considered local weighting. Here we omit the details due to the page limit.

IDF estimation. To estimate the IDF values of terms, Bobo used 664, 103 documents in the Ad-hoc track of TREC dataset.[2] These documents can produce a reasonable approximation as they cover various domains such as newspapers, U.S. patents, financial reports, congressional records, federal registers, and computer related contents.

In particular, for a term A, $IDF(A) = \log_2 \frac{n}{DF(A)}$, where $DF(A)$ is the document frequency of A in the TREC data set and n = 664,103.

[2] http://trec.nist.gov/data/docs_eng.html.

4.4 Similarity Computation

By computing similarities between round II results and seeds, Bobo estimates how close different results are to the search intent.

Document type. Seeds can either be in type of snippets (including titles) or full pages. So it is with round II results. White et al. (2007) reported that snippets performed even better than full texts for the task of pseudo RF. In our experiments, snippets also performed comparably to full pages. Thus, Bobo uses "snippet" as the default option for fast response time.

Similarity measure. Bobo uses two standard similarity measures, Cosine coefficient (default) and Jaccard coefficient. Both performed very well in our experiments, with the default option slightly better.

Prototype-based similarity. Bobo implements two types of similarity computation methods, prototype-based or instance-based, with the latter as the default option.

The prototype-based method is actually a form of the well-known Rocchio algorithm (Rocchio, 1971; Salton and Buckley, 1997), which is efficient but would perform poorly in the presence of polymorphic domains. In this method, the seeds are combined and the centroid of seeds is used in similarity computation. Given a set S of seeds, the centroid $\vec{u}$ is calculated as $\vec{u} = \frac{1}{|S|}\sum_{s\in S}\vec{s}$, where $\vec{s}$ is the vector space representation of seed $s \in S$.

Recall that the original Rocchio algorithm for query reformation is defined as follows,

$$\vec{q_e} = \alpha\vec{q} + \beta\frac{1}{|D_r|}\sum_{\vec{d_j}\in D_r}\vec{d_j} - \gamma\frac{1}{|D_{ir}|}\sum_{\vec{d_j}\in D_{ir}}\vec{d_j}$$

where q is the original query vector, q_e is the modified query vector, and D_r and D_{ir} represent the sets of known relevant and irrelevant document vectors respectively. α, β, and γ are empirically-chosen tuning parameters.

If we assign $\alpha = 0$ and $\gamma = 0$, the Rocchio formula agrees with our definition of centroid of seeds. We assign $\alpha = 0$ because Bobo does not target query reformation. We assign $\gamma = 0$ not because of the lack of negative feedback, which is not hard to identify from low-ranked round I search results. The reason is that even in explicit RF, there is no evidence that negative feedback improves performance (Schutze et al., 1995).

Figure 2: A Polymorphic Domain.

Instance-based similarity. Rocchio is simple and efficient. However, it over-generalizes training data and is inaccurate in the presence of polymorphic, or disjunctive, domains. In Figure 2, the 10 seeds labeled by "+" are split into two separate and rather distant sub-domains. The centroid of seeds labeled by "⊕" is not local to any sub-domain. Search result 1 is close to the centroid whereas result 2 is not. Rocchio would give high relevance score to result 1 and poor score to result 2. However, result 2 actually belongs to one of the two sub-domains whereas result 1 does not.

To handle polymorphic domains and capture locality, Bobo uses an instance-based approach, where the similarity of a document against each individual seed is computed, weighted, and aggregated. Let $sim(d, S)$ denote the similarity between a document d and a set S of seeds, then,

$$sim(d, S) = \sum_{s\in S} sim(d, s) \times sim(d, s)$$

Using this approach, result 2 will receive much higher relevance score than result 1 in Figure 2.

Note that, this approach resembles instance-based lazy learning such as k-nearest neighbor classification. Lazy learning generally has superior performance but would suffer from poor classification efficiency. This, however, is not a critical issue in our application because we do not have many seeds. The default number of seeds in Bobo is set to 10.

Discussion. While Bobo adopts rather standard approaches, we are aware of the many other approaches proposed in the literature for pairwise web page similarity computation. An interesting direction to investigate would be a link-based or hybrid approach. For example, Vassilvitskii and Brill (2006) uses web-graph distance for relevance feedback in web search.

5 Empirical Evaluation

We evaluated Bobo in comparison with regular Yahoo! search with and without using contextual terms. Results returned from Yahoo! may vary with time. This, however, will not change the general trends revealed by our empirical study. From these trends we conclude that, Bobo is a simple yet effective paradigm for query intent disambiguation without altering the original query and with maximized utilization of domain knowledge.

5.1 Experiment Setting and Methodology

Parameter setting. To emphasize the Bobo idea, unless otherwise specified, we used default options in the experiments that implement conventional approaches, e.g., TF-IDF for term weighting and Cosine coefficient for similarity computation. By default, number of seeds was set to 10 with each seed having at least 10 terms. The number of layers was set such that round II results were re-ranked in decreasing order of similarity. Cleaning seeds was set to yes. Purifying seeds, elf seed and weighting seeds were set to no.

Dataset. Finding information about people is one of the most common search activities. Around 30% of web queries include person names (Artiles et al., 2005). Person names, however, are highly ambiguous, e.g., only 90,000 different names are shared by 100 million people according to the U.S. Census Bureau (Guha and Garg, 2004).

To test the disambiguation effectiveness of Bobo, we constructed 60 ambiguous name queries and 180 test cases from the Wikipedia disambiguation pages.[3]

In Wikipedia, articles about two or more different topics could have the same natural page title. Disambiguation pages are then used to solve the conflicts. From the various categories, we used the human name category, containing disambiguation pages for multiple people of the same name. For each name, the disambiguation page lists all the different people together with their brief introductions. For example, an Alan Jackson is introduced as "born 1958, American country music singer and songwriter".

[3] en.wikipedia.org/wiki/Category:Disambiguation_pages.

Person names were chosen from the most common English first and last names for the year 2000 published on Wikipedia. The first 10 male and first 10 female given names were combined with the first 10 most common last names to make a list of 200 possible names. From this list, names were chosen based on the following criteria. For each name, there are at least 2 distinct people with the same name, each having at least 3 relevant pages in the returned 30 results.

In total 60 names were chosen as ambiguous queries. For each query, the actual information need was predetermined in a random manner. Then, for this predetermined person, 3 contextual terms were selected from her brief introduction, or her Wikipedia page in case the introduction was too short. For example, for the above Alan Jackson example, "music", "singer", or "songwriter" can be selected as contextual terms. Contextual terms were used one at a time, thus there are 3 test cases for each ambiguous query.

The identification of relevance of search results was done manually. For each query, let R_{30} be the set of relevant pages w.r.t. the information need contained in the 30 retrieved results. R_{30} can be considered containing the most important relevant pages for the original query.

Comparison partners and evaluation measures. To compare with Bobo, two types of regular search methods were used. The *Yahoo!* method uses the original query and performs the simplest Yahoo! web search, returning the same set of results as Bobo but without re-ranking.

To demonstrate the relevance improvement of Bobo over the *Yahoo!* method, we used a couple of standard ranking-aware evaluation measures, which were 11-point precision-recall graph, precision at k graph, Mean Average Precision (MAP) and R-precision.

The *Yahoo!-refined* method is the straightforward query refinement approach we previously discussed. It refines the original query by adding some contextual terms. The refined query alters the original query, leading to unfavorable ranking of results and failing to include many important relevant pages, i.e., R_{30} pages, in the top results.

To demonstrate this point, we used the recall at k evaluation measure, which measures the frac-

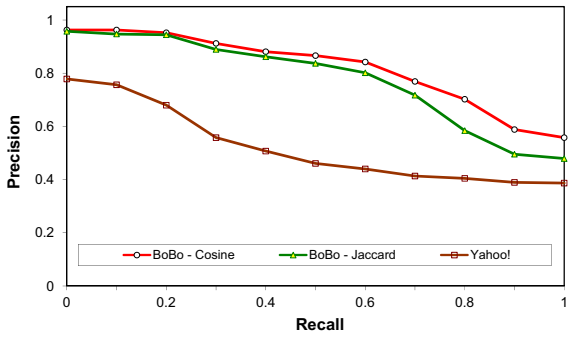


Figure 3: Bobo vs. Yahoo! on Averaged 11-point Precision-Recall.

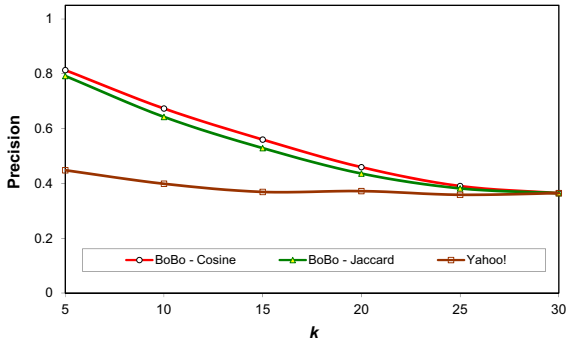


Figure 4: Bobo vs. Yahoo! on Averaged Precision at k.

tion of relevant pages (here, ones in R_{30}) contained in the top k results.

In the entire empirical study, the *Yahoo!* results were averaged over 60 queries, whereas all other results were averaged over 180 test cases.

5.2 Evaluation Results

In Figures 3, 4 and 5, Bobo results using both Cosine similarity and Jaccard coefficient are shown. The two performed similarly, with the former (default) slightly better.

Bobo vs. *Yahoo!*. The 11-point precision-recall graphs and precision at k graphs are presented in Figure 3 and Figure 4 respectively.

Web search users would typically browse a few top-ranked results. From Figure 4 we can see that for k =15, 10 and 5, the precision improvement of Bobo over *Yahoo!* is roughly 20% – 40%.

In addition, the MAP and R-precision values for Bobo are 0.812 and 0.740 respectively, whereas they are 0.479 and 0.405 for *Yahoo!* respectively. The improvement of Bobo over *Yahoo!* is about 33% for both measures.

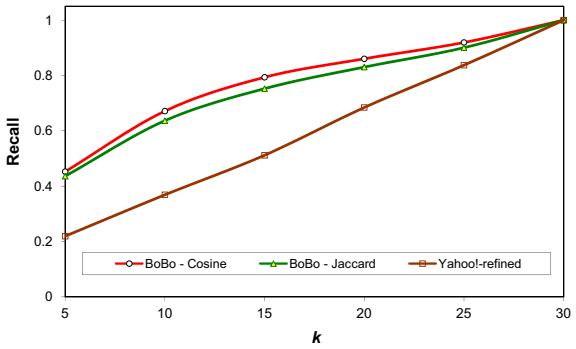


Figure 5: Bobo vs. Yahoo!-refined on Averaged Recall at k.

Bobo vs. *Yahoo!-refined*. The recall at k graphs are presented in Figure 5. From the figure we can see that for $k = 15$, $k = 10$ and $k = 5$, the recall (of important R_{30} pages) improvement of Bobo over *Yahoo!* is roughly 30%.

The results demonstrated that, although the straightforward query refinement approach can effectively improve relevance, it fails to rank those important relevant pages high, as it alters the original query and changes the query themes. Bobo, on the contrary, overcomes this problem by using the contextual terms "in the backstage", effectively improving relevance while keeping the original query intact.

Due to the page limit, here we omit other series of experiments that evaluated the flexibility of Bobo in choosing effective contextual terms and how the varied user preferences affect its performance. A user study was also conducted to test the usability and performance of Bobo.

6 Conclusions

As the web increases in size at an increasing rate, ambiguity becomes ubiquitous. In this paper, we introduced a novel Bobo approach to achieve simple yet effective search intent disambiguation without altering the original query and with maximized domain knowledge utilization.

Although we introduce Bobo in the context of web search, the idea can be applied to the settings of traditional archival information retrieval or multimedia information retrieval.

References

Artiles, Javier, Julio Gonzalo, and Felisa Verdejo. 2005. A testbed for people searching strategies in the WWW. In *SIGIR*.

Baeza-Yates, R. and B. Ribeiro-Neto. 1999. *Modern Information Retrieval.* Addison-Wesley.

Buckley, Chris, Singhal Amit, , and Mitra Mandar. 1995. New retrieval approaches using smart: Trec 4. In *TREC*.

Croft, W. and D. Harper. 1979. Using probabilistic models of information retrieval without relevance information. *Journal of Documentation*, 35(4):285–295.

Finkelstein, Lev, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. 2002. Placing search in context: the concept revisited. *ACM Transactions on Information Systems*, 20(1):116–131.

Gkanogiannis, Anestis and Theodore Kalamboukis. 2008. An algorithm for text categorization. In *SIGIR*.

Guha, R. V. and A. Garg. 2004. Disambiguating people in search. In *WWW*.

Haveliwala, Taher H. 2002. Topic-sensitive pagerank. In *WWW*.

Jansen, B. J., A. Spink, , and T. Saracevic. 2000. Real life, real users and real needs: A study and analysis of users queries on the web. *Information Processing and Management*, 36(2):207–227.

Joachims, Thorsten, Laura Granka, Bing Pan, Helene Hembrooke, and Geri Gay. 2005. Accurately interpreting clickthrough data as implicit feedback. In *SIGIR*.

Kelly, Diane and Jaime Teevan. 2003. Implicit feedback for inferring user preference: a bibliography. *SIGIR Forum*, 37(2):18–28.

Kraft, Reiner, Chi Chao Chang, Farzin Maghoul, and Ravi Kumar. 2006. Searching with context. In *WWW*.

Lawrence, Steve. 2000. Context in web search. *IEEE Data Engineering Bulletin*, 23(3):25–32.

Lee, Kyung Soon, W. Bruce Croft, and James Allan. 2008. A cluster-based resampling method for pseudo-relevance feedback. In *SIGIR*.

Manning, Christopher D., Prabhakar Raghavan, and Hinrich Schutze. 2008. *Introduction to Information Retrieval*. Cambridge University Press.

Mitra, M., A. Singhal, and C. Buckley. 1998. Improving automatic query expansion. In *SIGIR*.

Rocchio, J. 1971. *Relevance Feedback in Information Retrieval.* In The SMART Retrieval System – Experiments in Automatic Document Processing. Prentice-Hall.

Ruthven, Ian and Mounia Lalmas. 2003. A survey on the use of relevance feedback for information access systems. *Knowledge Engineering Review*, 18(1).

Salton, Gerard and Chris Buckley. 1997. *Improving retrieval performance by relevance feedback.* Morgan Kaufmann.

Schapire, Robert E., Yoram Singer, and Amit Singhal. 1998. Boosting and rocchio applied to text filtering. In *SIGIR*.

Schutze, Hinrich, David A. Hull, and Jan O. Pedersen. 1995. A comparison of classifiers and document representations for the routing problem. In *SIGIR*.

Singhal, Amit, Mandar Mitra, and Chris Buckley. 1997. Learning routing queries in a query zone. In *SIGIR*.

Teevan, Jaime, Susan T. Dumais, and Eric Horvitz. 2005. Personalizing search via automated analysis of interests and activities. In *SIGIR*.

Vassilvitskii, Sergei and Eric Brill. 2006. Using webgraph for relevance feedback in web search. In *SIGIR*.

Voorhees, Ellen M. 1993. Using wordnet to disambiguate word senses for text retrieval. In *SIGIR*.

White, Ryen W., Joemon M. Jose, C. J. Van Rijsbergen, and Ian Ruthven. 2004. A simulated study of implicit feedback models. In *ECIR*.

White, Ryen W., Charles L.A. Clarke, and Silviu Cucerzan. 2007. Comparing query logs and pseudo-relevance feedback for web search query refinement. In *SIGIR*.

Yu, Shipeng, Deng Cai, Ji-Rong Wen, and Wei-Ying Ma. 2003. Improving pseudo-relevance feedback in web information retrieval using web page segmentation. In *WWW*.

Zhu, Yangbo, Jamie Callan, and Jaime Carbonell. 2008. The impact of history length on personalized search. In *SIGIR*.

Comparing the performance of two TAG-based surface realisers using controlled grammar traversal

Claire Gardent
CNRS/LORIA
claire.gardent@loria.fr

Benjamin Gottesman
acrolinx GmbH
ben.gottesman@acrolinx.com

Laura Perez-Beltrachini
Université Henri Poincaré/LORIA
laura.perez@loria.fr

Abstract

We present GENSEM, a tool for generating input semantic representations for two sentence generators based on the same reversible Tree Adjoining Grammar. We then show how GENSEM can be used to produced large and controlled benchmarks and test the relative performance of these generators.

1 Introduction

Although computational grammars are mostly used for parsing, they can also be used to generate sentences. This has been done, for instance, to *detect overgeneration by the grammar* (Gardent and Kow, 2007). Sentences that are generated but are ungrammatical indicate flaws in the grammar. This has also been done to *test a parser* (Nederhof, 1996; Purdom, 1972). Using the sentences generated from the grammar ensures that the sentences given to the parser are in the language it defines. Hence a parse failure necessarily indicates a flaw in the parser's design as opposed to a lack of coverage by the grammar.

Here we investigate a third option, namely, the *focused benchmarking of sentence realisers* based on reversible grammars, i.e. on grammars that can be used both to produce sentences from a semantic representation and semantic representations from a sentence.

More specifically, we present a linguistically-controlled grammar traversal algorithm for Tree Adjoining Grammar (TAG) which, when applied to a reversible TAG, permits producing arbitrarily many of the semantic representations associated by this TAG with the sentences it generates. We then show that the semantic representations thus produced can be used to compare the relative performance of two sentence generators based on this grammar.

Although the present paper concentrates on Tree Adjoining Grammar realisers, it is worth pointing out that the semantic representations produced could potentially be used to evaluate any surface realiser whose input is a flat semantic formula.

Section 2 discusses related work and motivates the approach. Section 3 presents GENSEM, the DCG-based grammar traversal algorithm we developed. We show, in particular, that the use of a DCG permits controlling grammar traversal in such a way as to systematically generate sets of semantic representations covering certain computationally or linguistically interesting cases. Finally, Section 4 reports on the benchmarking of two surface realisers with respect to a GENSEM-produced benchmark.

2 Motivations

Previous work on benchmark construction for testing the performance of surface realisers falls into two camps depending on whether or not the realiser uses a reversible grammar, that is, a grammar that can be used for both parsing and generation.

To test a surface realiser based on a large reversible Head-Driven Phrase Structure Grammar (HPSG), Carroll et al. (1999) use a small test set of two hand-constructed and 40 parsing-derived cases to test the impact of intersective modifiers[1] on generation performance. More recently, Carroll and Oepen (2005) present a perfor-

[1] As first noted by Brew (1992) and Kay (1996), given a set of n modifiers all modifying the same structure, all possible intermediate structures will be constructed, i.e., 2^{n+1}.

Coling 2008: Poster Volume, pages 338–346,
Beijing, August 2010

mance evaluation which uses as a benchmark the set of semantic representations produced by parsing 130 sentences from the Penn Treebank and manually selecting the correct semantic representations. Finally, White (2004) profiles a CCG[2]-based sentence realiser using two domain-focused reversible CCGs to produce two test suites of 549 and 276 ⟨ semantic formula, target sentence ⟩ pairs, respectively.

For realisers that are not based on a reversible grammar, there are approaches which derive large sets of realiser input from the Penn Treebank (PTB). For example, Langkilde-Geary (2002) proposes to translate the PTB annotations into a format accepted by her sentence generator Halogen. The output of this generator can then be automatically compared with the PTB sentence from which the corresponding input was derived. Similarly, Callaway (2003) builds an evaluation benchmark by transforming PTB trees into a format suitable for the KPML realiser he uses.

In all of the above cases, the data is derived from real world sentences, thereby exemplifying "real world complexity". If the corpus is large enough (as in the case of the PTB), the data can furthermore be expected to cover a broad range of syntactic phenomena. Moreover, the data, being derived from real world sentences, is not biased towards system-specific capabilities. Nonetheless, there are also limits to these approaches.

First, they fail to support graduated performance testing on constructs such as intersective modifiers or lexical ambiguity, which are known to be problematic for surface realisation.

Second, the construction of the benchmark is in both cases time consuming. In the reversible approach, for each input sentence, the correct interpretation must be manually selected from among the semantic formulae produced by the parser. As a side effect, the constructed benchmarks remain relatively small (825 in the case of White (2004); 130 in Carroll and Oepen (2005)). In the case of a benchmark derived by transformation from a syntactically annotated corpus, the implementation of the converter is both time-intensive and corpus-bound. For instance, Callaway (2003) reports that the implementation of such a processor for the SURGE realiser was the most time-consuming part of the evaluation with the resulting component containing 4000 lines of code and 900 rules.

As we shall show in the following sections, the GENSEM approach to benchmark construction aims to address both of these shortcomings. By using a DCG to implement grammar traversal, it permits both a full automation of the benchmark creation and some control over the type and the distribution of the benchmark items.

3 GenSem

As mentioned above, GENSEM is a grammar traversal algorithm for TAG. We first present the specific TAG used for traversal, namely SEMXTAG (Alahverdzhieva, 2008) (section 3.1). We then show how to automatically derive a DCG that describes the derivation trees of this grammar (section 3.2). Finally, we show how this DCG encoding permits generating formulae while enabling control over the set of semantic representations to be produced (section 3.3).

3.1 SemXTAG

The SEMXTAG grammar used by GENSEM and by the two surface realisers is a Feature-Based Lexicalised Tree Adjoining Grammar augmented with a unification-based semantics as described by Gardent and Kallmeyer (2003). We briefly introduce each of these components and describe the grammar coverage.

FTAG. A Feature-based TAG (Vijay-Shanker and Joshi, 1988) consists of a set of (auxiliary or initial) elementary trees and of two tree-composition operations: substitution and adjunction. Initial trees are trees whose leaves are labelled with substitution nodes (marked with a downarrow) or terminal categories. Auxiliary trees are distinguished by a foot node (marked with a star) whose category must be the same as that of the root node. Substitution inserts a tree onto a substitution node of some other tree while adjunction inserts an auxiliary tree into a tree. In an FTAG, the tree nodes are furthermore decorated with two feature structures (called **top** and

[2]Combinatory Categorial Grammar

bottom) which are unified during derivation as follows. On substitution, the top of the substitution node is unified with the top of the root node of the tree being substituted in. On adjunction, the top of the root of the auxiliary tree is unified with the top of the node where adjunction takes place; and the bottom features of the foot node are unified with the bottom features of this node. At the end of a derivation, the top and bottom of all nodes in the derived tree are unified. Finally, each sentence derivation in an FTAG is associated with both a **derived tree** representing the phrase structure of the sentence and a **derivation tree** recording how the corresponding elementary trees were combined to form the derived tree.

FTAG with semantics. To associate semantic representations with natural language expressions, the FTAG is modified as proposed by Gardent and Kallmeyer (2003).

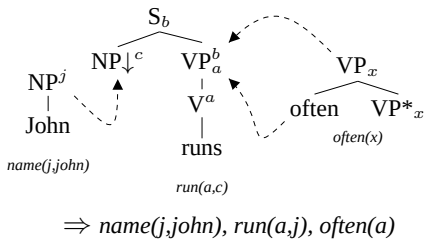


Figure 1: Flat semantics for "John often runs"

Each elementary tree is associated with a flat semantic representation. For instance, in Figure 1,[3] the trees for *John*, *runs*, and *often* are associated with the semantics *name(j,john)*, *run(a,c)*, and *often(x)*, respectively. Importantly, the arguments of a semantic functor are represented by unification variables which occur both in the semantic representation of this functor and on some nodes of the associated syntactic tree. For instance in Figure 1, the semantic index c occurring in the semantic representation of *runs* also occurs on the subject substitution node of the associated elementary tree. The value of semantic arguments is determined by the unifications resulting from adjunction and substitution. For instance, the semantic index c in the tree for *runs* is unified during substitution with the semantic index labelling the root node of the tree for *John*. As a result, the semantics of *John often runs* is {name(j,john),run(a,j),often(a)}.

SemXTAG. SEMXTAG is an FTAG for English augmented with a unification-based compositional semantics of the type described above. Its syntactic coverage approaches that of XTAG, the FTAG developed for English by the XTAG group (The XTAG Research Group, 2001). Like this grammar, it contains around 1300 elementary trees and covers auxiliaries, copula, raising and small clause constructions, topicalization, relative clauses, infinitives, gerunds, passives, adjuncts, ditransitives and datives, ergatives, it-clefts, wh-clefts, PRO constructions, noun-noun modification, extraposition, sentential adjuncts, imperatives and resultatives.

3.2 Converting SemXTAG to a DCG

We would like to be able to traverse SEMXTAG in order to generate semantic representations that are licensed by it. In the DCG formalism, a grammar is represented as a set of Prolog definite clauses, and Prolog's query mechanism provides built-in grammar traversal. We take advantage of this by deriving a DCG from SEMXTAG and then using Prolog queries to generate semantic representations that are associated with sentences in the language described by it.

Another advantage of the DCG formalism is that arbitrary Prolog goals can be inserted into a rule, to constrain when the rule applies or to bind variables occurring in it. We use this to ground derivations with lexical items, which are represented using Prolog assertions. We also use it to control Prolog's grammar traversal in such a way as to generate sets of semantic formulae covering certain computationally interesting cases (see section 3.3).

Our algorithm for converting SEMXTAG to a DCG is inspired by Schmitz and Le Roux (2008), who derive from an FTAG a feature-based regular tree grammar (RTG) whose language is the derivation trees of the FTAG. Indeed, in our implementation, we derive a DCG from such an RTG, thereby taking advantage of a SEMXTAG-

[3] C^x/C_x abbreviate a node with category C and a top/bottom feature structure including the feature-value pair { **index :** x}.

to-RTG converter previously implemented by Sylvain Schmitz.

TAG to RTG. In the conversion to RTG[4], each elementary tree in SEMXTAG is converted to a rule that models the contribution of the tree to a TAG derivation. A TAG derivation involves the selection of an initial tree, which has some nodes requiring substitution and some permitting adjunction. Let us think of the potential adjunction sites as requiring, rather than permitting, adjunction, but such that the requirement can be satisfied by 'null' adjunction. Inserting another tree into this initial tree satisfies one of the substitution or adjunction requirements, but introduces some new requirements into the resulting tree, in the form of its own substitution nodes and adjunction sites.

Thus, intuitively, the RTG representation of a SEMXTAG elementary tree is a rule that rewrites the satisfied requirement as a local tree whose root is a unique identifier of the tree and whose leaves are the introduced requirements. A requirement of a substitution or adjunction of a tree of root category X is written as X_S or X_A, respectively. Here, for example, is the translation to RTG of the TAG tree (minus semantics) for *runs* in Figure 1, using the word anchoring the tree as its identifier (the superscripts abbreviate feature structures: b/t refers to the bottom/top feature structure and the upper case letters to the semantic index value, so $[idx : X]$ is abbreviated to X):

$$S_S^{[t:T]} \rightarrow runs(S_A^{[t:T,b:B]}\ NP_S^{[t:C]}\ VP_A^{[t:B,b:A]}\ V_A^{[t:A]})$$

The semantics of the SEMXTAG tree are carried over as-is to the corresponding RTG rule. Further, the feature structures labelling the nodes of SEMXTAG trees are carried over to the RTG rules so as to correctly interact with substitution and adjunction (see Schmitz and Le Roux (2008) for more details on this part of the conversion process).

To account for the optionality of adjunction, there are additional rules allowing any adjunction requirement to be rewritten as the symbol ϵ, a terminal symbol of the RTG.

The terminal symbols of the RTG are thus the tree identifiers and the symbol ϵ, and its non-terminals are X_S and X_A for each terminal or non-terminal X of SEMXTAG.

RTG to DCG. Since the right-hand side of each RTG rule is a local tree – that is, a tree of depth no more than one – we can flatten each of them into a list consisting of the root node followed by the leaves without losing any structural information. This is the insight underlying the RTG-to-DCG conversion step. Each RTG rule is converted to a DCG rule that is essentially identical except for this flattening of the right-hand side. Here is the translation to DCG of the RTG rule above[5]:

```
rule(s,init,Top,Bot,Sem;S;N;VP;V)
  -->  [runs],
       {lexicon(runs,n0V,[run])},
       rule(s,aux,Top,[B],S),
       rule(np,init,[C],_,N),
       rule(vp,aux,[B],[A],VP),
       rule(v,aux,[A],_,V),
       {Sem =.. [run,A,C]}.
```

We represent non-terminals of the DCG using the `rule` predicate, whose five (non-hidden)[6] arguments, in order, are the category, the subscript (`init` for subscript S, `aux` for subscript A), the *top* and *bottom* feature values, and the semantics. Feature structures are represented using Prolog lists with a fixed argument position for each attribute in the grammar (in this example, only the index attribute). The semantics associated with the left-hand-side symbol (here, `Sem;S;N;VP;V`, with the semicolon representing semantic conjunction) are composed of the semantics associated with this rule and those associated with each of the right-hand-side symbols.

The language of the resulting DCG is neither the language of the RTG nor the language of SEMXTAG, and indeed the language of the DCG does not interest us but rather its derivation trees.

[4]For a more precise description of the FTAG to RTG conversion see Schmitz and Le Roux (2008).

[5]In practice, the lexicon is factored out, so there is no rule specifically for *runs*, but one for intransitive verbs (n0V) in general. Each rule hooks into the lexicon, so that a given invocation of a rule is grounded by a particular lexical item.

[6]The $-->$ notation is syntactic sugar for the usual Prolog $:-$ definite clause notation with two hidden arguments on each predicate. The hidden arguments jointly represent the list of terminals dominated by the symbol.

These are correlated one-to-one with the trees in the language described by the RTG, i.e. with the derivation trees of SEMXTAG, and the latter can be trivially reconstructed from the DCG derivations. From a SEMXTAG derivation tree, one can compose the semantic representation of the associated sentence, and in fact this semantic composition occurs as a side effect of a Prolog query against the DCG, allowing semantic representations licensed by SEMXTAG to be returned as query results.

We define a Prolog predicate for querying against the DCG, as follows. Its one input argument, `Cat`, is the label of the root node of the derivation tree (typically `s`), and its one output argument, `Sem`, is the semantic representation associated with that tree[7].

```
genSem(Cat,Sem) :-
    rule(Cat,init,_,_,Sem,_,[]).
```

3.3 Control parameters

In order to give the users some control over the sorts of semantic representations that they get back from a query against the DCG, we augment the DCG in such a way as to allow control over the TAG family[8] of the root tree in the derivation tree, over the number and type of adjunctions in the derivation, and over the depth of substitutions. To implement control over the family is quite simple: we need merely to index the DCG rules by family and modify the GENSEM call accordingly. For instance, the above DCG rule becomes :

```
rule(s,init,Top,Bot,n0V,Sem;S;NP;VP;V)
  -->  [runs],
       {lexicon(runs,n0V,[run])},
       ...
```

We implement restrictions on adjunctions by adding an additional argument to the grammar symbols, namely a vector of non-negative integers representing the number of non-null adjunctions of each type that are in the derivation subtree dominated by the symbol. By 'type' of adjunction, we mean the category of the adjunction site. In DCG terms, a non-null adjunction of a category `X` is represented as the expansion of an `x/aux` symbol other than as ϵ. So, for example, a DCG symbol associated with the vector `[1,0,0,0,0]`, where the five dimensions of the vector correspond to the `n`, `np`, `v`, `vp`, and `s` categories, respectively, dominates a subtree containing exactly one `n/aux` symbol expanded by a non-epsilon rule, and no other `aux` symbol expanded by a non-epsilon rule. We link the vector associated with the root of the derivation to the query predicate.

We define a special predicate to handle the divvying up of a mother node's vector among the daughters, taking advantage of the fact that the DCG formalism permits the insertion of arbitrary Prolog goals into a rule.

Finally, we add an additional argument to the DCG rule and to the GENSEM's call to control the traversal depth with respect to the number of substitutions applied. The overall depth of each derivation is therefore constrained both by the user defined adjunctions and substitution depth constraints.

Our query predicate now has four input arguments and one output argument:

```
genSem(Cat,Fam,[N,NP,V,VP,S],Dth,Sem):-
    rule(Cat,init,_,_,Fam,
         [N,NP,V,VP,S],Dth,Sem,_,[]).
```

4 Using GENSEM for benchmarking

We now show how GENSEM can be put to work for comparing two TAG-based surface realisers, namely GENI (Gardent and Kow, 2007) and RTGEN (Perez-Beltrachini, 2009). These two realisers follow globally similar algorithms but differ in several respects. We show how GENSEM can be used to produce benchmarks that are tailored to test hypotheses about how these differences might impact performance. We then use this GENSEM-generated benchmark to compare the performance of the two realisers.

4.1 GenI and RTGen

Both GENI and RTGEN use the SEMXTAG grammar described in section 3.1. Moreover, both realisers follow an algorithm pipelining three main phases. First, **lexical selection** selects from the

[7] The 6th and 7th arguments of the rule call are the hidden arguments needed by the DCG.

[8] TAG families group together trees which belong together, in particular, the trees associated with various realisation of a specific subcategorisation type. Thus, here the notion of TAG family is equivalent to that of subcategorisation type.

grammar those elementary trees whose semantics subsumes part of the input semantics. Second, the **tree combining** phase systematically tries to combine trees using substitution and adjunction. Third, the **retrieval phase** extracts the yields of the complete derived trees, thereby producing the generated sentence(s).

There are also differences however. We now spell these out and indicate how they might impact the relative performance of the two surface realisers.

Derived vs. derivation trees. While GENI constructs derived trees, RTGEN uses the RTG encoding of SEMXTAG sketched in the previous section to construct derivation trees. These are then unraveled into derived trees at the final retrieval stage. As noted by Koller and Striegnitz (2002), these trees are simpler than TAG elementary trees, which can favourably impact performance.

Interleaving of feature constraint solving and syntactic analysis. GENI integrates in the tree combining phase a filtering step in which the initial search space is pruned by eliminating from it all combinations of TAG elementary trees that cover the input semantics but cannot possibly lead to a valid derived tree. This filtering eliminates all combinations of trees such that either the category of a substitution node cannot be cancelled out by that of the root node of a different tree, or a root node fails to have a matching substitution site. Importantly, filtering ignores feature information and tree combining takes place after filtering. RTGEN, on the other hand, directly combines derivation trees decorated with full feature structure information.

Handling of intersective modifiers. GENI and RTGEN differ in their strategies for handling modification.

Adapting Carroll and Oepen's (2005) proposal to TAG, GENI adopts a two-step tree-combining process such that in the first step, only substitution applies, while in the second, only adjunction is used. Although the number of intermediate structures generated is still 2^n for n modifiers, this strategy has the effect of blocking these 2^n structures from multiplying out with other structures in the chart.

RTGEN, on the other hand, uses a standard Earley algorithm that includes sharing and packing. Sharing allows intermediate structures common to several derivations to be represented once only while packing groups together partial derivation trees with identical semantic coverage and similar combinatorics (same number and type of substitution and adjunction requirements), keeping only one representative of such groups in the chart. In this way, intermediate structures covering the same set of intersective modifiers in a different order are only represented once and the negative impact of intersective modifiers is lessened.

4.2 Two GENSEM benchmarks

We use GENSEM to construct two benchmarks designed to test the impact of the differences between the two realisers and, more specifically, to compare the relative performance of both realisers (i) on cases involving intersective modifiers and (ii) on cases of varying overall complexity.

The MODIFIERS benchmark focuses on intersective modifiers and contains semantic formulae corresponding to sentences involving an increasing number of modifiers. Recall that GENSEM calls are of the form *gensem(Cat,Family,[N,NP,V,VP,S],Dth,Sem)* where *N,NP,V,VP,S* indicates the number of required adjunctions in *N, NP, V, VP* and *S*, respectively, while *Family* constrains the subcategorisation type of the root tree in the derivations produced by GENSEM. To produce formulae involving the lexical selection of intersective modifiers, we set the following constraints. *Cat* is set to s and *Family* is set to either n0V (intransitive verbs) or n0Vn1 (transitive verbs). Furthermore, N and VP vary from 0 to 4 thereby requiring the adjunction of 0 to 4 N and/or VP modifiers. All other adjunction counters are set to null. To avoid producing formulae with identical derivation trees but distinct lemmas, we use a restricted lexicon containing one lemma of each syntactic type, e.g. one transitive verb, one intransitive verb, etc. Given these settings, GENSEM produces 1 789 formulae whose adjunction requirements vary from 1 to 6. For instance, the semantic formula

{*sleep(b,c),man(c),a(c),blue(c),sleep(i,c),carefully(b)*} (*A sleeping blue man sleeps carefully*) extracted from the MODIFIERS benchmark contains two NP adjunctions and one VP adjunction.

The MODIFIERS benchmark is tailored to focus on cases involving a varying number of intersective modifiers. To support a comparison of the realisers on this dimension, it displays little or no variation w.r.t. other dimensions, such as verb type and non-modifying adjunctions.

To measure the performance of the two realisers on cases of varying overall complexity, we construct a second benchmark (COMPLEXITY) displaying such variety. The GENSEM parameters for the construction of this suite are the following. The verb type (*Family*) is one of 28 possible verb types[9]. The number and type of required adjunctions vary from 0 to 4 for N adjunctions, 0 to 1 for NP, 0 to 4 for VP and 0 to 1 for S. The resulting benchmark contains 890 semantic formulae covering an extensive set of verb types and of adjunction requirements.

4.3 Results

Using the two GENSEM-generated benchmarks, we now compare GENI and RTGEN. We plot the average number of chart items against both the number of intersective modifiers present in the input (Figure 3) and the size of the Initial Search Space (ISS), i.e., the number of combinations of elementary TAG trees covering the input semantics to be explored after the **lexical selection** step (Figure 2). In our case, the ISS gives a more meaningful idea about the complexity than considering only the number of input literals. In an FTAG, the number of elementary trees selected by a given literal may vary considerably depending on the number and the size of the tree families selected by this literal. For instance, a literal selecting the n0Vn2n1 class will select many more trees than a literal selecting the n0V family because there are many more ways of realising the three arguments of a ditransitive verb than the single subject argument of an intransitive one. Chart items include all elementary trees selected by the lexical selection step as well as the intermediate and final structures produced by the tree combining phase. In RTGEN, we distinguish between the number of structures built before unpacking (packed chart) and the number of structures obtained after unpacking (unpacked chart).

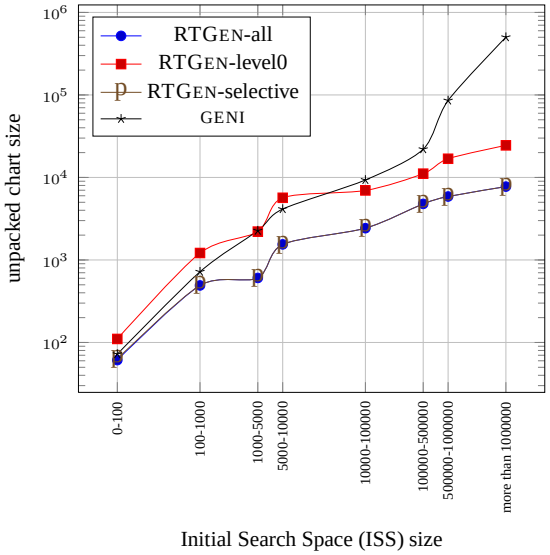


Figure 2: Performance of realisation approaches on the COMPLEXITY benchmark, average unpacked chart size as a function of the ISS complexity.

Both realisers are implemetned in different programming languages, GENI is implemented in Haskell whereas RTGEN in Prolog. As for the time results comparison, preliminary experiments show that GENI is faster is simple input cases. On the other hand, in the case of more complex cases, the point of producing much less intermediate results pays off compared to the overhead of the chart/agenda operations.

[9]The 28 verb types are En1V,n0BEn1,n0lVN1Pn2,n0V,n0Va1,n0VAN1,n0VAN1Pn2, n0VDAN1,n0VDAN1Pn2,n0VDN1,n0VDN1Pn2,n0Vn1, n0VN1,n0Vn1Pn2,n0VN1Pn2,n0Vn2n1,n0Vpl,n0Vpln1, n0Vpn1,n0VPn1,n0Vs1,REn1VA2,REn1VPn2,Rn0Vn1A2, Rn0Vn1Pn2,s0V,s0Vn1,s0Vton1. The notational convention for verb types is from XTAG and reads as follows. Subscripts indicate the thematic role of the verb argument. n indicates a nominal, Pn a PP and s a sentential argument. pl is a verbal particle. Upper case letters describe the syntactic functor type: V is a verb, E an ergative, R a resultative and BE the copula. Sequences of upper case letters such as VAN in n0VAN1 indicate a multiword functor with syntactic categories V, A, and N. For instance, n0Vn1 indicates a verb taking two nominal arguments (e.g., *like*) and n0VAN1 a verb locution such as *to cry bloody murder*.

Overall efficiency. The plot in Figure 2 shows the results obtained by running both realisers on the COMPLEXITY benchmark. Recall (cf. section 4.2) that the COMPLEXITY benchmark contains input with varying verb arity and a varying number of required adjunctions. Hence it provides cases of increasing complexity in terms of ISS to be explored. Furthermore, test cases in the benchmark trigger sentence realisation involving certain TAG families, which have a certain number of trees. Those trees within a family often have identical combinatorics but different features. Consequently, the COMPLEXITY benchmark also provides an appropriate testbed for testing the impact of feature structure information on the two approaches to tree combination.

The graphs show that as complexity increases, the performance delta between GENI and RTGEN increases. We conjecture that as complexity grows, the filtering used by GENI does not suffice to reduce the search space to a manageable size. Conversely, the overhead introduced by RTGEN's all-in-one, tree-combining Earley with packing strategy seems compensated throughout by the construction of a derivation rather than a derived tree and pays off increasingly as complexity increases.

Modifiers. Figure 3 plots the results obtained by running the realisers on the MODIFIERS benchmark. Here again, RTGEN outperforms GENI and the delta between the two realisers grows with the number of intersective modifiers to be handled. A closer look at the data shows that the global constraints set by GENSEM on the number of required adjunctions covers an important range of variation in the data complexity. For instance, there are cases where 4 modifiers modify the same NP (or VP) and cases where the modifiers are distributed over two NPs. Similarly, literals introduced into the formula by a GENSEM adjunction requirement vary in terms of the number of auxiliary trees whose selection they trigger. The steep curve in GENI's plot suggests that although the delayed adjunction mechanism helps in avoiding the proliferation of intermediate incomplete modifiers' structures, the lexical ambiguity of modifiers still poses a problem. In contrast, RTGEN's

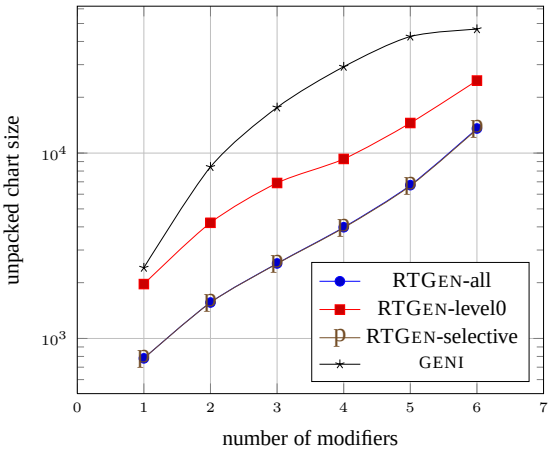


Figure 3: Performance of realisation approaches on the MODIFIERS benchmark, average unpacked chart size as a function of the number of modifiers.

packing uniformly applies to word order variations and to the cases of lexical ambiguity raised by intersective modifiers because the items have the same combinatoric potential and the same semantics.

5 Conclusion

Surface realisers are complex systems that need to handle diverse input and require complex computation. Testing raises among other things the issue of coverage – how can the potential input space be covered? – and of test data creation – should this data be hand tailored, created randomly, or derived from real world text?

In this paper, we presented an approach which permits automating the creation of test input for surface realisers whose input is a flat semantic formula. The approach differs from other existing evaluation schemes in two ways. First, it permits producing arbitrarily many inputs. Second, it supports the construction of grammar-controlled, linguistically focused benchmarks.

We are currently working on further extending GENSEM with more powerful (recursive) control restrictions on the grammar traversal; on combining GENSEM with tools for detecting grammar overgeneration; and on producing a benchmark that could be made available to the community for testing surface realisers whose input is either a dependency tree or a flat semantic formula.

References

Alahverdzhieva, K. 2008. XTAG using XMG. Master's thesis, U. Nancy 2. Erasmus Mundus Master "Language and Communication Technology".

Brew, C. 1992. Letting the cat out of the bag: Generation for shake-and-bake MT. In *Proceedings of COLING '92*, Nantes, France.

Callaway, Charles B. 2003. Evaluating coverage for large symbolic NLG grammars. In *18th IJCAI*, pages 811–817, Aug.

Carroll, John and Stephan Oepen. 2005. High efficiency realization for a wide-coverage unification grammar. *2nd IJCNLP*.

Carroll, John, A. Copestake, D. Flickinger, and V. Paznański. 1999. An efficient chart generator for (semi-)lexicalist grammars. In *Proceedings of EWNLG '99*.

Gardent, Claire and Laura Kallmeyer. 2003. Semantic construction in FTAG. In *10th EACL*, Budapest, Hungary.

Gardent, Claire and Eric Kow. 2007. Spotting overgeneration suspects. In *11th European Workshop on Natural Language Generation (ENLG)*.

Kay, Martin. 1996. Chart generation. In *Proceedings of the 34th annual meeting on Association for Computational Linguistics*, pages 200–204, Morristown, NJ, USA. Association for Computational Linguistics.

Koller, Alexander and Kristina Striegnitz. 2002. Generation as dependency parsing. In *Proceedings of the 40th ACL*, Philadelphia.

Langkilde-Geary, Irene. 2002. An empirical verification of coverage and correctness for a general-purpose sentence generator. In *Proceedings of the INLG*.

Nederhof, M.-J. 1996. Efficient generation of random sentences. *Natural Language Engineering*, 2(1):1–13.

Perez-Beltrachini, Laura. 2009. Using regular tree grammars to reduce the search space in surface realisation. Master's thesis, Erasmus Mundus Master Language and Communication Technology, Nancy/Bolzano.

Purdom, Paul. 1972. A sentence generator for testing parsers. *BIT*, 12(3):366–375.

Schmitz, S. and J. Le Roux. 2008. Feature unification in TAG derivation trees. In Gardent, C. and A. Sarkar, editors, *Proceedings of the 9th International Workshop on Tree Adjoining Grammars and Related Formalisms (TAG+'08)*, pages 141–148, Tübingen, Germany.

The XTAG Research Group. 2001. A lexicalised tree adjoining grammar for english. Technical report, Institute for Research in Cognitive Science, University of Pennsylvannia.

Vijay-Shanker, K. and AK Joshi. 1988. Feature Structures Based Tree Adjoining Grammars. *Proceedings of the 12th conference on Computational linguistics*, 55:v2.

White, Michael. 2004. Reining in CCG chart realization. In *INLG*, pages 182–191.

Verbs are where all the action lies: Experiences of Shallow Parsing of a Morphologically Rich Language

Harshada Gune **Mugdha Bapat** **Mitesh M. Khapra** **Pushpak Bhattacharyya**
Department of Computer Science and Engineering,
Indian Institute of Technology Bombay
{harshadag,mbapat,miteshk,pb}@cse.iitb.ac.in

Abstract

Verb suffixes and verb complexes of morphologically rich languages carry a lot of information. We show that this information if harnessed for the task of shallow parsing can lead to dramatic improvements in accuracy for a morphologically rich language- Marathi[1]. The crux of the approach is to use a powerful morphological analyzer backed by a high coverage lexicon to generate rich features for a CRF based sequence classifier. Accuracy figures of 94% for Part of Speech Tagging and 97% for Chunking using a modestly sized corpus (20K words) vindicate our claim that for morphologically rich languages linguistic insight can obviate the need for large amount of annotated corpora.

1 Introduction

Shallow parsing which involves Part-of-Speech (POS) tagging and Chunking is a fundamental task of Natural Language Processing (NLP). It is natural to view each of these sub-tasks as a sequence labeling task of assigning POS/chunk labels to a given word sequence. For languages like English where annotated corpora are available in abundance these tasks can be performed with very high accuracy using data-driven machine learning techniques. Languages of the world show different levels of readiness with respect to such annotated resources and hence not all languages may provide a conducive platform for machine learning techniques.

In this scenario, morphologically rich languages from the Indian subcontinent present a very interesting case. While these languages do not enjoy the resource abundance of English, their linguistic richness can be used to offset this resource deficit. Specifically, in such languages, the suffixes carry a lot of information about the category of a word which can be harnessed for shallow parsing. This is especially true in the case of verbs where suffixes like णे {ne}, णारे {naare} [2] clearly indicate the category of the word. Further, the structure of verb groups in such languages is relatively rigid and can be used to reduce the ambiguity between main verbs and auxiliary verbs.

In the current work, we aim to reduce the data requirement of machine learning techniques by appropriate feature engineering based on the characteristics of the language. Specifically, we target Marathi- a morphologically rich language- and show that a powerful morphological analyzer backed by a high coverage lexicon and a simple but accurate Verb Group Identifier (VGI) can go a long way in improving the accuracy of a state of the art sequence classifier. Further, we show that harnessing such features is the only way by which one can hope to build a high-accuracy classifier for such languages, and that simply throwing in a large amount of annotated corpora does not serve the purpose. Hence it makes more sense to invest time and money in developing good morphological analyzers for such languages than investing in annotation. Accuracy figures of 94% for Part of

[1] Marathi is the official language of Maharashtra, a state in Western India. The language has close to 20 million speakers in the world.

[2] These are the suffixes which derive infinitive and gerund verb forms respectively.

Coling 2008: Poster Volume, pages 347–355,
Beijing, August 2010

Speech Tagging and 97% for Chunking using a modestly sized corpus (20K words) vindicate our claim that for morphologically rich languages linguistic knowledge plays a very important role in shallow parsing of these languages.

2 Related Work

Many POS taggers have been built for English employing machine learning techniques ranging from Decision Trees (Black et al., 1992) to Graphical Models (Brants, 2000; Brill, 1995; Ratnaparkhi, 1996; Lafferty et al., 2001). Even hybrid taggers such as CLAWS (Garside and Smith, 1997) which combine stochastic and rule based approaches have been developed. However, most of these techniques do not focus on harnessing the morphology; instead they rely on the abundance of data which is not a very suitable proposition for some of the resource deprived languages of the Indian sub-continent.

Morphological processing based taggers using a combination of hand-crafted rules and annotated corpora have been tried for Turkish (Oflazer and Kuruöz, 1994), Arabic (Tlili-Guiassa, 2006), Hungarian (Megyesi, 1999) and Modern Greek (Giorgos et al., 1999). The work on Hindi POS tagging (Singh et al., 2006) comes closest to our approach which showed that using a detailed linguistic analysis of morphosyntactic phenomena, followed by leveraging suffix information and accurate verb group identification can help to build a high-accuracy (93-94%) part of speech tagger for Hindi. However, to the best of our knowledge, there is no POS tagger and Chunker available for Marathi and ours is the first attempt at building one.

3 Motivating Examples

To explain the importance of suffix information for shallow parsing we present two motivating examples. First, consider the following Marathi sentence,

हा रस्ता दोन गावांना जोडणारा आहे.
*haa rasta don gavaannaa **jodaNaaraa_VM** aahe.*
*this road two villages **connecting** is*
*this is the road **connecting_VM** two villages.*

The word जोडणारा {jodaNaaraa} (connecting) in the above sentence is a verb and can be categorized as such by simply looking at the suffix णारा {Naaraa} as this suffix does not appear with any other POS category. When suffix information is used as a feature a statistical POS tagger is able to identify the correct POS tag of जोडणारा {jodaNaaraa} even when it does not appear in the training data. Hence, using suffix information ensures that a classifier is able to learn meaningful patterns even in the absence of large training data. Next, we consider two examples for chunking.

- **VGNN (Gerund Verb Chunk)**
  माणसाने उडण्याचा प्रयत्न केला.
  *maaNasaane **uDaNyaachaa_B-VGNN**[3]*
  prayatna kelaa.
  *man **fly** try do*
  *man tried **flying_B-VGNN**.*

- **VGINF (Infinitival Verb Chunk)**
  त्याने चालायला सुरुवात केली.
  *tyaane **chaalaayalaa_B-VGNF** suruvaata kelii.*
  *he **walk** start did*
  *he started **to walk_B-VGINF**.*

Here, we are dealing with the case of two specific verb chunks, *viz.*, VGNN (gerund verb chunk) and VGINF (infinitival verb chunk). A chunk having a gerund always gets annotated as VGNN and a chunk having an infinitival verb always gets annotated as VGINF. Thus, the correct identification of these verb chunks boils down to the correct identification of gerunds and infinitival verb forms in the sentence which in turn depend on the careful analysis of suffix information. For example, in Marathi, the attachment of the verbal suffix "ण्याचा" {Nyaachaa} to a verb root always results in a gerund. Similarly, the attachment of the verbal suffix "यला" {yalaa} to a verb root always results in an infinitival verb form. The use of such suffix information as features can thus lead to better generalization for handling unseen words and thereby reduce the need for additional training data. For instance, in the first sentence, even when the word "उडण्याचा" {uDaNyaachaa} does not appear in

[3]Note that for all our experiments we used BI scheme for chunking as opposed to the BIO scheme

the training data, a classifier which uses suffix information is able to label it correctly based on its experience of previous words having suffix "ण्याचा" {Nyaachaa} whereas a classifier which does not use suffix information fails to classify it correctly.

4 Morphological Structure of Marathi

Marathi nouns inflect for number and case. They may undergo derivation on the attachment of postpositions. In the oblique case, first a stem is obtained from the root by applying the rules of inflection. Then a postposition is attached to the stem. Postpositions (including case markers and the derivational suffixes) play a very important role in Marathi morphology due to the complex morphotactics.

Marathi adjectives can be classified into two categories: ones that do not inflect and others that inflect for gender, number and case where such an inflection agrees with the gender and number of the noun modified by them.

The verbs inflect for gender, number and person of the subject and the direct object in a sentence. They also inflect for tense and aspect of the action as well as mood of the speaker in an illocutionary act. They may even undergo derivation to derive the nouns, adjectives or postpositions. Verbal morphology in Marathi is based on *Aakhyaata* theory for inflection and *Krudanta* theory for derivation which are two types of verb suffixes (Damale, 1970).

Aakhyaata Theory: *Aakhyaata* refers to tense, aspect and mood. Aakhyaata form is realized through an aakhyaata suffix which is a closing suffix attached to verb root. For example, बसला {basalaa} (*sat*) comes from *basa* + *laa*. There are 8 types of aakhyaatas named after the phonemic shape of the aakhyaata suffix. Associated with every aakhyaata are various aakhyaata-arthas which indicate the features: tense, aspect and mood. An aakhyaata may or may not agree with gender.

Krudanta Theory: Krudanta suffixes are attached to the end of verbs to form non-infinitive verb forms. For example, धावायला (धाव + आयला) {dhaavaayalaa} (to run). There are 8 types of krudantas defined in Marathi.

5 Design of Marathi Shallow Parser

Figure 1 and 2 show the overall architectures of Marathi POS tagger and chunker. The proposed system contains 3 important components. First, a morphological analyzer which provides ambiguity schemes and suffix information for generating a rich set of features. Ambiguity Scheme refers to the list of possible POS categories a word can take. This can add valuable information to a sequence classifier by restricting the set of possible POS categories for a word. For example, the word जात {jaat} meaning caste or go(caste-noun, go- VM/VAUX) can appear as a noun or a main verb or an auxiliary verb. Hence it falls in the ambiguity scheme <NN-VM-VAUX>. This information is stored in a lexicon. These features are then fed to a CRF based engine which couples them with other elementary features (previous/next words and bigram tags) for training a sequence labeler. Finally, in the case of POS tagger, we use a Verb Group Identifier (VGI) which acts as an error correcting module for correcting the output of the CRF based sequence labeler. Each of these components is described in detail in the following sub-sections.

5.1 Morphological Analyzer

The formation of polymorphemic words leads to complexities which need to be handled during the analysis process. For example, consider the steps involved in the formation of the word देवासमोरच्याने {devasamorchyane} (the one in front of the God + ERGATIVE).

$$
\begin{aligned}
\text{devaasamora} &= (\text{deva} \rightarrow \text{devaa}) \\
&\quad + \text{samora} \\
\text{devaasamorachaa} &= (\text{devaasamora} \rightarrow \text{devaasamora}) \\
&\quad + \text{chaa} \\
\text{devaasamorachyaane} &= (\text{devaasamorachaa} \rightarrow \text{devaasamorachyaa}) \\
&\quad + \text{ne}
\end{aligned}
$$

In theory, the process can continue recursively for the attachment of any number of suffixes. However, in practice, we have observed that a word in Marathi contains at most 4 suffixes.

FSMs prove to be elegant and computationally efficient tools for analyzing polymorphemic

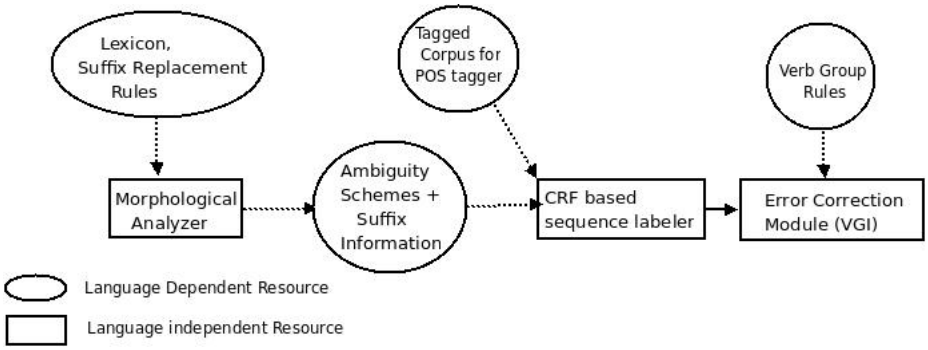


Figure 1: Architecture of POS Tagger

words. However, the recursive process of word formation in Marathi involves inflection at the time of attachment of every new suffix. The FSM needs to be modified to handle this. However, during the *i-th* recursion only *(i-1)-th* morpheme changes its form which can be handled by suitably modifying the FSM. The formation of word देवासमोरच्याने {devaasamorachyaane} can be viewed as:

devaasamora	=	(deva → devaa)
		+ samora
devaasamorachaa	=	(deva → devaa)
		+ (samora → samora)
		+ chaa
devaasamorachyaane	=	(deva → devaa)
		+ (samora → samora)
		+ (chaa → chyaa)
		+ ne

In general,
Polymorphemic word = $(inflected\_morpheme_1)$ + $(inflected\_morpheme_2)$ + ...

Now, we can create an FSM which is aware of these inflected forms of morphemes in addition to the actual morphemes to handle the above recursive process of word formation. These inflected forms are generated using the paradigm-based[4] system written in Java and then fed to the FSM implemented using SFST[5].

[4]A paradigm identifies the uninflected form of words which share similar inflectional patterns.

[5]http://www.ims.uni-stuttgart.de/projekte/gramotron/SOFTWARE/SFST.html

Our lexicon contains 16448 nouns categorized into 76 paradigms, 8516 adjectives classified as inflecting and non-inflecting adjectives, 1160 verbs classified into 22 classes. It contains 142 postpositions, 80 aakhyaata and 8 krudanta suffixes.

5.2 CRF

Conditional Random Fields (Lafferty et al., 2001) are undirected graphical models used for labeling sequential data. Under this model, the conditional probability distribution of a tag given the observed word sequence is given by,

$$P(Y|X;\lambda) = \frac{1}{Z(X)} \cdot e^{\sum_{t=1}^{T}\sum_{k=1}^{K}\lambda_k f_k(Y_{t-1},Y_t,X,t)} \quad (1)$$

where,

$$X = source\ word$$
$$Y = target\ word$$
$$T = length\ of\ sentence$$
$$K = number\ of\ features$$
$$\lambda_k = feature\ weight$$
$$Z(X) = normalization\ constant$$

We used CRF++[6], an open source implementation of CRF, for training and further decoding the tag sequence. We used the following features for training the sequence labeler (here, w_i is the *i-th* word, t_i is the *i-th* pos tag and c_i is the *i-th* chunk tag).

[6]http://crfpp.sourceforge.net/

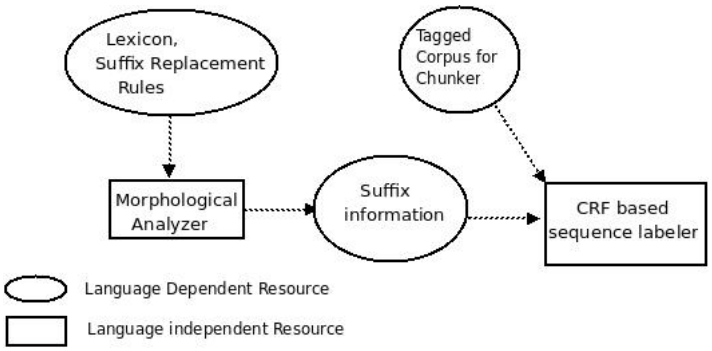


Figure 2: Architecture of Chunker

Features used for POS tagger training
Consider position of interest = i

- $t_i\ t_{i-1}$ and w_j such that $i-3 < j < i+3$
- $t_i\ t_{i-1}$ and suffix information of w_i
- $t_i\ t_{i-1}$ and ambiguity scheme of w_i

Here, the first features are ***weak features*** which depend only on the previous/next words and bigram tags. The next two are ***rich morphological features*** which make use of the output of the morphological analyzer.

Features used for Chunker training
Consider position of interest = i

- $c_i\ c_{i-1}$ and t_j, w_j such that $i-3 < j < i+3$
- $c_i\ c_{i-1}$ and suffix information of w_i

where c_i, $c_{i-1} \in \{B, I\}$. Here again, the first set of features are ***weak features*** and the second set of features are ***rich morphological features***.

5.3 Verb Group Identification (VGI)

In Marathi, certain auxiliaries like असते {asate} (be), आहे {aahe} (is) *etc.*. can also act as main verbs in certain contexts. This ambiguity between VM (main verbs) and VAUX (auxiliary verbs) can lead to a large number of errors in POS tagging if not handled correctly. However, the relatively rigid structure of Marathi VG coupled with distinct suffix-affinity of auxiliary verbs allows us to capture this ambiguity well using the following simple regular expression:

MainVerbRoot (KrudantaSuffix AuxVerbRoot)* AakhyaataSuffix

The above regular expression imposes some restriction on the occurrence of certain auxiliary verbs after specific *krudanta* suffixes. This restriction is captured with the help of a rule file containing *krudanta suffix-auxiliary verb* pairs. A sample entry from this file is

ऊन , काढ *[oon, kaaDh]*

which suggests that the auxiliary verb काढ {kaaDh} can appear after the suffix ऊन {oon}. We created a rule file containing around 350 such valid *krudanta suffix-auxiliary* verb pairs.

An important point which needs to be highlighted here is that a simple left to right scan ignoring suffix information and marking the first verb constituent as main verb and every other constituent as auxiliary verb does not work for Marathi. For example, consider the following verb sequence,

त्याला उचलून आणावे लागले.
tyaalaa uchaluun aaNaave laagale
He carry bring need
It was needed to carry and bring him.

Here, a simple left to right scan of the verb sequence ignoring the suffix information would imply that उचलून is a VM whereas आणावे and लागले are VAUX. However, this is not the case and can be identified correctly by considering the suffix affinity of auxiliary verbs. Specifically, in this case, the verb root आण cannot take the role of an auxiliary verb when it appears after the *krudanta* suffix ऊन. This suggests that the verb

आणावे does not belong to the same verb group as उचलून and hence is not a VAUX. This shows suffix and regular expression help in disambiguating VM-VAUX which is a challenge in all POS taggers.

6 Experimental Setup

We used documents from the TOURISM and NEWS domain for all our experiments [7]. These documents were hand annotated by two Marathi lexicographers. The total size of the corpus was kept large (106273 POS tagged words and 63033 chunks) to study the impact of the size of training data versus the amount of linguistic information used. The statistics about each POS tag and chunk tag are summarized in Table 1 and Table 2.

POS Tag	Frequency in Corpus	POS Tag	Frequency in Corpus
NN	51047	RP	359
NST	578	CC	3735
PRP	8770	QW	630
DEM	3241	QF	1928
VM	17716	QC	2787
VAUX	6295	QO	277
JJ	7311	INTF	158
RB	1060	INJ	22
UT	97	RDP	39
PSP	69	NEG	154

Table 1: POS Tags in Training Data

Chunk Tag	Frequency in Corpus	Chunk Tag	Frequency in Corpus
NP	40254	JJP	2680
VGF	7425	VGNF	3553
VGNN	1105	VGINF	58
RBP	782	BLK	2337
CCP	4796	NEGP	43

Table 2: Chunk Tags in Training Data

7 Results

We report results in four different settings:
Weak Features (WF): Here we use the basic CRF classifier with elementary word features (*i.e.*, words appearing in a context window of 3) and bigram tag features and POS tags in case of chunker.
Weak Morphological Features (Weak-MF): In addition to the elementary features we use substrings of length 1 to 7 appearing at the end of the word as feature. The idea here is that such substrings taken from the end of the word can provide a good approximation of the actual suffix of the word. Such substrings thus provide a statistical approximation of the suffixes in the absence of a full fledged morphological analyzer. This should not be confused with weak features which mean tags and word.
Rich Morphological Features (Rich-MF): In addition to the elementary features we use the ambiguity schemes and suffix information provided by the morphological analyzer.
Reach Morphological Features + Verb Group Identification (Rich-MF+VGI): This setting is applicable only for POS tagging where we apply an error correcting VGI module to correct the output of the feature rich CRF tagger.

In each case we first divided the data into four folds (75% for training and 25% for testing). Next, we varied the training data in increments of 10K and calculated the accuracy of each of the above models. The x-axis represents the size of the training data and the y-axis represents the precision of the tagger/chunker. Figure 3 plots the average precision of the POS tagger across all categories using WF, Weak-MF, Rich-MF and Rich-MF_VGI for varying sizes of the training data. Figure 6 plots the average precision of the chunker across all categories using WF, Weak-MF and Rich-MF. Next, to show that the impact of morphological analysis is felt more for verbs than other POS categories we plot the accuracies of verb pos tags (Figure 4) and verb chunk tags (Figure 7) using WF, Weak-MF, Rich-MF and Rich-MF_VGI for varying sizes of the training data.

8 Discussions

We made the following interesting observations from the above graphs and tables.
1. Importance of linguistic knowledge: Figure 3 shows that using a large amount of annotated corpus (91k), the best accuracy one can hope

[7] The data can be found at www.cfilt.iitb.ac.in/

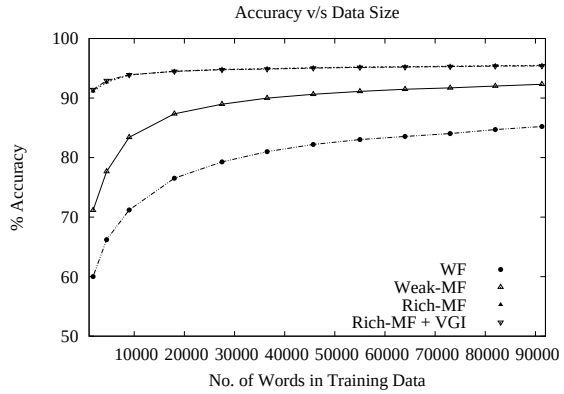


Figure 3: Average Accuracy of all POS Tags
(Note: The graphs for Rich-MF and Rich-MF+VGI coincide)

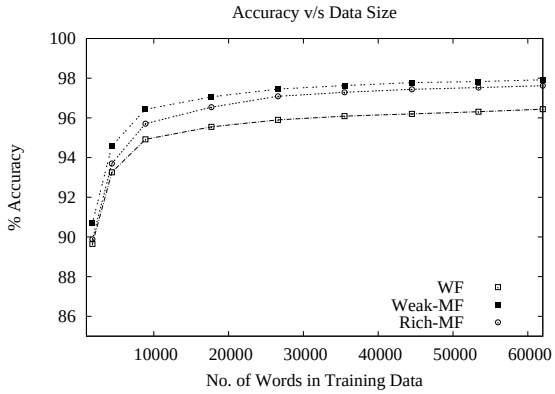


Figure 6: Average Accuracy of all Chunk Tags

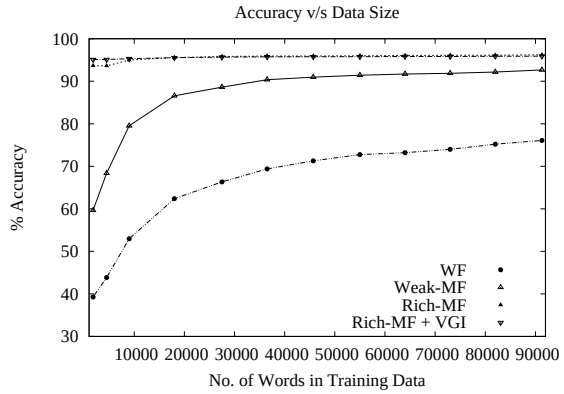


Figure 4: Average Accuracy of Verb POS Tags
(Note: The graphs for Rich-MF and Rich-MF+VGI almost coincide)

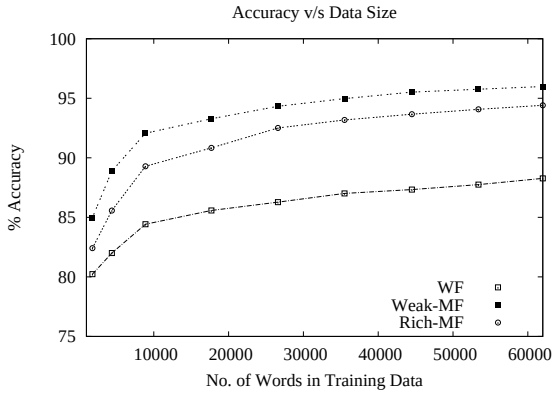


Figure 7: Average Accuracy of Verb Chunks

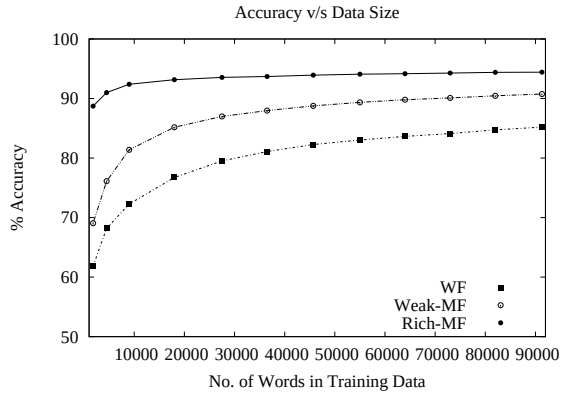


Figure 5: Average Accuracy of Non Verb POS Tags

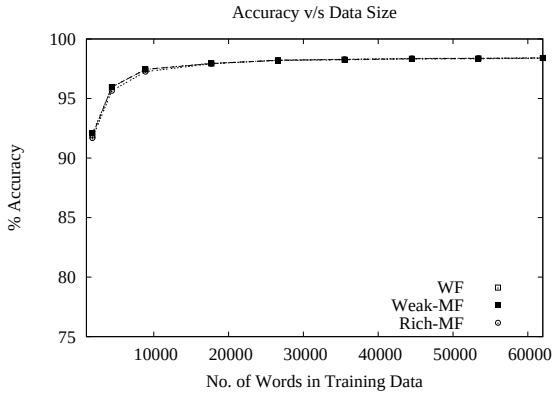


Figure 8: Average Accuracy of Non Verb Chunks
(Note: All the graphs coincide.)

for is around 85% if morphological information is not harnessed *i.e.*, if only weak features are used. Adding more data will definitely not be of much use as the curve is already close to saturation. On the other hand, if morphological information is completely harnessed using a rich morphological analyzer then an accuracy as high as 94% can be obtained by using data as small as 20k words. Figure 6 tells a similar story. In the absence of morphological features a large amount of annotated corpus (62k words) is needed to reach an accuracy of 96%, whereas if suffix information is used then the same accuracy can be reached using a much smaller training corpus (20k words). This clearly shows that while dealing with morphologically rich languages, time and effort should be invested in building powerful morphological analyzer.

2. Weak morphological features vs rich morphological analyzer: Figure 3 shows that in the case of POS tagging using just weak morphological features gives much better results than the baseline (i.e. using only weak features). However, it does not do as well as the rich features especially when the training size is small, thereby suggesting that an approximation of the morphological suffixes may not work for a language having rich and diverse morphology. On the other hand, in the case of chunking, the weak morphological features do marginally better than the rich morphological features suggesting that for a relatively easier task (chunking as compared to POS tagging) even a simple approximation of the actual suffixes may deliver the goods.

3. Specific case of verbs: Figure 4 shows that in case of POS tagging using suffixes as features results in a significant increase in accuracy of verbs. Specifically accuracy increases from 62% to 95% using a very small amount of annotated corpus (20K words). Comparing this with figure 5 we see that while using morphological information definitely helps other POS categories, the impact is not as high as that felt for verbs. Figures 7 and 8 for chunking show a similar pattern *i.e.*, the accuracy of verb chunks is affected more by morphology as compared to other chunk tags. These figures support our claim that "verbs is where all the action lies" and they indeed need special treatment in terms of morphological analysis.

	VM	VAUX
VM	17078	347
VAUX	257	6025

Table 3: Confusion matrix for VM-VAUX using Rich-MF

4. Effect of VGI: Figures 3 and 4 show that the VGI module does not lead to any improvement in the overall accuracy. A detailed analysis showed that this is mainly because there was not much VM-VAUX ambiguity left after applying CRF model containing rich morphological features. To further illustrate our point we present the confusion matrix (see Table 3) for verb tags for a POS tagger using Rich-MF. Table 3 shows that there were only 347 VM tags which got wrongly tagged as VAUX and 257 VAUX tags which got wrongly tagged as VM. Thus the rich morphological features were able to take care of most VM-VAUX ambiguities in the data. However we feel that if the data contains several VM-VAUX ambiguities such as the one illustrated in the example in Section 5.3 then the VGI module would come in play and help to boost the performance by resolving such ambiguities.

9 Conclusion

We presented here our work on shallow parsing of a morphologically rich language- Marathi. Our results show that while dealing with such languages one cannot ignore the importance of harnessing morphological features. This is especially true for verbs where improvements upto 50% in accuracy can be obtained by adroit handling of suffixes and accurate verb group identification. An important conclusion that can be drawn from our work is that while dealing with morphologically rich languages it makes sense to invest time and money in developing powerful morphological analyzers than placing all the bets on annotating data.

References

Black, Ezra, Fred Jelinek, John Lafferty, Robert Mercer, and Salim Roukos. 1992. Decision tree models applied to the labeling of text with parts-of-

speech. In *HLT '91: Proceedings of the workshop on Speech and Natural Language*, pages 117–121, Morristown, NJ, USA. Association for Computational Linguistics.

Brants, Thorsten. 2000. TnT - A Statistical Part-of-Speech Tagger. In *6th Applied Natural Language Processing (ANLP '00), April 29 - May 4*, pages 224–231. Association for Computational Lingusitics.

Brill, Eric. 1995. Transformation-Based Error-Driven Learning and Natural Language Processing: A Case Study in Part-of-Speech Tagging. *Computational Linguistics*, 21(4):543–565.

Damale, M K. 1970. *Shastriya Marathi Vyaakarana*. Pune Deshmukh and Company.

Garside, Roger and Nicholas Smith. 1997. A Hybrid Grammatical Tagger: CLAWS. In Garside, Roger, Geoffrey Leech, and Tony McEnery, editors, *Corpus Annotation*, pages 102–121. Longman, London.

Giorgos, Orphanos, Kalles Dimitris, Papagelis Thanasis, and Christodoulakis Dimitris. 1999. Decision Trees and NLP: A case study in POS Tagging.

Lafferty, John, Andrew McCallum, and F. Pereira. 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proc. 18th International Conf. on Machine Learning*, pages 282–289. Morgan Kaufmann, San Francisco, CA.

Megyesi, Beta. 1999. Improving Brill's POS Tagger For An Agglutinative Language, 02.

Oflazer, Kemal and Ilker Kuruöz. 1994. Tagging and Morphological Disambiguation of Turkish Text. In *ANLP*, pages 144–149.

Ratnaparkhi, Adwait. 1996. A Maximum Entropy Model for Part-of-Speech Tagging. In Brill, Eric and Kenneth Church, editors, *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 133–142. Association for Computational Linguistics, Somerset, New Jersey.

Singh, Smriti, Kuhoo Gupta, Manish Shrivastava, and Pushpak Bhattacharyya. 2006. Morphological Richness Offsets Resource Demand - Experiences in Constructing a POS Tagger for Hindi. In *Proceedings of ACL-2006*.

Tlili-Guiassa, Yamina. 2006. Hybrid Method for Tagging Arabic Text. *Journal of Computer Science 2*, 3:245–248.

A Semantic Network Approach to Measuring Relatedness

Brian Harrington
Oxford University Computing Laboratory
brian.harrington@comlab.ox.ac.uk

Abstract

Humans are very good at judging the strength of relationships between two terms, a task which, if it can be automated, would be useful in a range of applications. Systems attempting to solve this problem automatically have traditionally either used relative positioning in lexical resources such as WordNet, or distributional relationships in large corpora. This paper proposes a new approach, whereby relationships are derived from natural language text by using existing NLP tools, then integrated into a large scale semantic network. Spreading activation is then used on this network in order to judge the strengths of all relationships connecting the terms. In comparisons with human measurements, this approach was able to obtain results on par with the best purpose built systems, using only a relatively small corpus extracted from the web. This is particularly impressive, as the network creation system is a general tool for information collection and integration, and is not specifically designed for tasks of this type.

1 Introduction

The ability to determine semantic relatedness between terms is useful for a variety of NLP applications, including word sense disambiguation, information extraction and retrieval, and text summarisation (Budanitsky and Hirst, 2006). However, there is an important distinction to be made between semantic *relatedness* and semantic *similarity*. As (Resnik, 1999) notes, "Semantic similarity represents a special case of semantic relatedness: for example, cars and gasoline would seem to be more closely related than, say, cars and bicycles, but the latter pair are certainly more similar". (Budanitsky and Hirst, 2006) further note that "Computational applications typically require relatedness rather than just similarity; for example, money and river are cues to the in-context meaning of bank that are just as good as trust company".

Systems for automatically determining the degree of semantic relatedness between two terms have traditionally either used a measurement based on the distance between the terms within WordNet (Banerjee and Pedersen, 2003; Hughes and Ramage, 2007), or used co-occurrence statistics from a large corpus (Mohammad and Hirst, 2006; Padó and Lapata, 2007). Recent systems have, however, shown improved results using extremely large corpora (Agirre et al., 2009), and existing large-scale resources such as Wikipedia (Strube and Ponzetto, 2006).

In this paper, we propose a new approach to determining semantic relatedness, in which a semantic network is automatically created from a relatively small corpus using existing NLP tools and a network creation system called ASKNet (Harrington and Clark, 2007), and then spreading activation is used to determine the strength of the connections within that network. This process is more analogous to the way the task is performed by humans. Information is collected from fragments and assimilated into a large semantic knowledge structure which is not purposely built for a single task, but is constructed as a general resource containing a wide variety of information. Relationships represented within this

Coling 2008: Poster Volume, pages 356–364,
Beijing, August 2010

structure can then be used to determine the total strength of the relations between any two terms.

2 Existing Approaches

2.1 Resource Based Methods

A popular method for automatically judging semantic distance between terms is through WordNet (Fellbaum, 1998), using the lengths of paths between words in the taxonomy as a measure of distance. While WordNet-based approaches have obtained promising results for measuring semantic similarity (Jiang and Conrath, 1997; Banerjee and Pedersen, 2003), the results for the more general notion of semantic relatedness have been less promising (Hughes and Ramage, 2007).

One disadvantage of using WordNet for evaluating semantic relatedness is its hierarchical taxonomic structure. This results in terms such as *car* and *bicycle* being close in the network, but terms such as *car* and *gasoline* being far apart. Another difficulty arises from the non-scalability of WordNet. While the quality of the network is high, the manual nature of its construction means that arbitrary word pairs may not occur in the network. Hence in this paper we pursue an approach in which the resource for measuring semantic relatedness is created automatically, based on naturally occurring text.

A similar project, not using WordNet is WikiRelate (Strube and Ponzetto, 2006), which uses the existing link structure of Wikipedia as its base network, and uses similar path based measurements to those found in WordNet approaches to compute semantic relatedness. This project has seen improved results over most WordNet base approaches, largely due to the nature of Wikipedia, where articles tend to link to other articles which are related, rather than just ones which are similar.

2.2 Distributional Methods

An alternative method for judging semantic distance is using word co-occurrence statistics derived from a very large corpus (McDonald and Brew, 2004; Padó and Lapata, 2007) or from the web using search engine results (Turney, 2001).

In a recent paper, Agirre et al. (2009) parsed 4 billion documents (1.6 Terawords) crawled from the web, and then used a search function to extract syntactic relations and context windows surrounding key words. These were then used as features for vector space, in a similar manner to work done in (Padó and Lapata, 2007), using the British National Corpus (BNC). This system has produced excellent results, indicating that the quality of the results for these types of approaches is related to the size and coverage of their corpus. This does however present problems moving forward, as 1.6 Terawords is obviously an extremely large corpus, and it is likely that there would be a diminishing return on investment for increasingly large corpora. In the same paper, another method was shown which used the pagerank algorihm, run over a network formed from WordNet and the WordNet gloss tags in order to produce equally impressive results.

3 A Semantic Network Approach

The resource we use is a semantic network, automatically created by the large scale network creation program, ASKNet. The relations between nodes in the network are based on the relations returned by a parser and semantic analyser, which are typically the arguments of predicates found in the text. Hence terms in the network are related by the chain of syntactic/semantic relations which connect the terms in documents, making the network ideal for measuring the general notion of semantic relatedness.

Distinct occurrences of terms and entities are combined into a single node using a novel form of spreading activation (Collins and Loftus, 1975). This combining of distinct mentions produces a cohesive connected network, allowing terms and entities to be related across sentences and even larger units such as documents. Once the network is built, spreading activation is used to determine semantic

relatedness between terms. For example, to determine how related *car* and *gasoline* are, activation is given to one of the nodes, say *car*, and the network is "fired" to allow the activation to spread to the rest of the network. The amount of activation received by *gasoline* is then a measure of the strength of the semantic relation between the two terms.

We use three datasets derived from human judgements of semantic relatedness to test our technique. Since the datasets contain general terms which may not appear in an existing corpus, we create our own corpus by harvesting text from the web via Google. This approach has the advantage of requiring little human intervention and being extensible to new datasets. Our results using the semantic network derived from the web-based corpus are comparable to the best performing existing methods tested on the same datasets.

4 Creating the Semantic Networks

ASKNet creates the semantic networks using existing NLP tools to extract syntactic and semantic information from text. This information is then combined using a modified version of the update algorithm used by Harrington and Clark (2007) to create an integrated large-scale network. By mapping together concepts and objects that relate to the same real-world entities, the system is able to transform the output of various NLP tools into a single network, producing semantic resources which are more than the sum of their parts. Combining information from multiple sources results in a representation which would not have been possible to obtain from analysing the original sources separately.

The NLP tools used by ASKNet are the C&C parser (Clark and Curran, 2007) and the semantic analysis program Boxer (Bos et al., 2004), which operates on the CCG derivations output by the parser to produce a first-order representation. The named entity recognizer of Curran and Clark (2003) is also used to recognize the standard set of MUC entities, including `person`, `location` and `organisation`.

As an example of the usefulness of information integration, consider the *monk-asylum* example, taken from the RG dataset (described in Section 5.1). It is possible that even a large corpus could contain sentences linking *monk* with *church*, and linking *church* with *asylum*, but no direct links between *monk* and *asylum*. However, with an integrated semantic network, activation can travel across multiple links, and through multiple paths, and will show a relationship, albeit probably not a very strong one, between *monk* and *asylum*, which corresponds nicely with our intuition.

Figure 1, which gives an example network built from DUC documents describing the Elian Gonzalez custody battle, gives an indication of the kind of network that ASKNet builds. This figure does not give the full network, which is too large to show in a single figure, but shows the "core" of the network, where the core is determined using the technique described in (Harrington and Clark, 2009). The black boxes represent named entities mentioned in the text, which may have been mentioned a number of times across documents, and possibly using different names (e.g. Fidel Castro vs. President Castro). The diamonds are named directed edges, which represent relationships between entities.

A manual evaluation using human judges has been performed to measure the accuracy of ASKNet networks. On a collection of DUC documents, the "cores" of the resulting networks were judged to be 80% accurate on average, where accuracy was measured for the merged entity nodes in the networks and the relations between those entities (Harrington and Clark, 2009). The motivation for fully automatic creation is that very large networks, containing millions of edges, can be created in a matter of hours.

Automatically creating networks does result in lower precision than manual creation, but this is offset by the scalability and speed of creation. The experiments described in this paper are a good test of the automatic creation methodology.

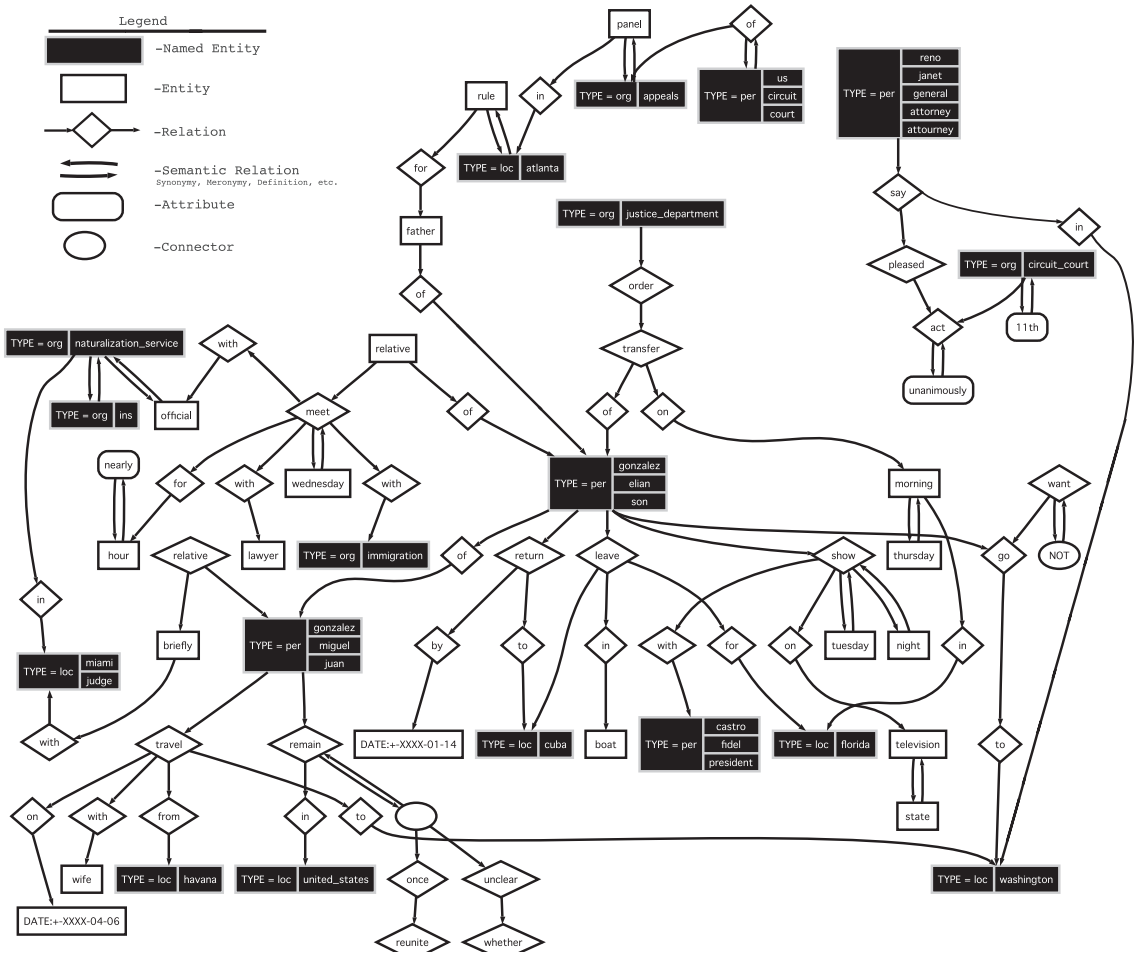


Figure 1: Example network core derived from DUC documents describing a news story from 2000.

The update algorithm

In order to merge information from multiple sources into a single cohesive resource, ASKNet uses a spreading activation based update algorithm (Harrington and Clark, 2007). As the system encounters new information, it attempts to map named entities in the new sentences it encounters to those already in its network. Each new sentence is turned into an *update fragment*; a fragment of the network representing the information contained in the sentence. Initial mapping scores, based on lexical similarity and named entity type, are made between the update fragment's named entities and those in the main network. The job of the update algorithm is to improve upon those initial scorings using the semantic information contained within the network.

The update algorithm iterates over each named entity node in the update fragment. This *base* node is provided with activation, which is allowed to spread throughout the fragment. All named entities which receive activation in this process, then transfer their activation to their *target* named entity nodes (nodes in the main network with which they have a current mapping score greater than zero). The amount transferred is based on the strength of the mapping score. The activation is then allowed to circulate through the main network until it reaches a stable state. At this point, the base node's mappings are updated based on which of its target nodes received activation. The more activation a target node receives, the more its mapping score with the

base node will increase.

The intuition behind the update algorithm is that we can use relatedness of nodes in the update fragment to determine appropriate mappings in the main network. So if our base node has the label "Crosby", and is related to named entity nodes referring to "Canada", "Vancouver" and "2010", those nodes will pass their activation onto their main network targets, and hopefully onto the node representing the ice hockey player Sidney Crosby. We would then increase the mapping score between our base node and this target, while at the same time decreasing the mapping score between out base node and the singer Bing Crosby, who (hopefully) would have received little or no activation. The update algorithm is also self-reinforcing, as in the successive stages, the improved scores will focus the activation further. In our example, in successive iterations, more of the activation coming to the "Crosby" node will be sent to the appropriate target node, and therefore there will be less spurious activation in the network to create noise.

For the purposes of these experiments, we extended the update algorithm to map together general object nodes, rather than focusing solely on named entities. This was necessary due to the nature of the task. Simply merging named entities would not be sufficient, as many of the words in datasets would not likely be associated strongly with any particular named entities. Extending the algorithm in this way resulted in a much higher frequency of mapping, and a much more connected final network. Because of this, we found that several of the parameters had to be changed from those used in Harrington and Clark (2009). Our initial activation input was set to double that used in Harrington and Clark's experiments (100 instead of 50), to compensate for the activation lost over the higher number of links. We also found that the number of iterations required to reach a stable state had increased to more than 4 times the previous number. This was to be expected due to the increased number of links passing activation. We also had to remove the named entity type calculation from the initial mapping score, thus leaving the initial scoring to be simply the ratio of labels in the two nodes which overlapped. These changes were all done after manual observation of test networks built from searches not relating to any dataset, and were not changed once the experiments had begun.

4.1 Measuring semantic relatedness

Once a large-scale network has been constructed from a corpus of documents, spreading activation can be used to efficiently obtain a distance score between any two nodes in the network, which will represent the semantic relatedness of the pair. Each node in the network has a current amount of activation and a threshold (similar to classical ideas from the neural network literature). If a node's activation exceeds its threshold, it will fire, sending activation to all of its neighbours, which may cause them to fire, and so on. The amount of activation sent between nodes decreases with distance, so that the effect of the original firing is localized. The localized nature of the algorithm is important because it means that semantic relatedness scores can be calculated efficiently even for pairs of nodes in very large networks.

To obtain a score between nodes `x` and `y`, first a set amount of activation is placed in node `x`; then the network is fired until it stabilises, and the total amount of activation received by node `y` is stored as `act(x,y)`. This process is repeated starting with node `y` to obtain `act(y,x)`. The sum of these two values, which we call `dist(x,y)`, is used as the measure of semantic relatedness between `x` and `y`.[1]

`dist(x,y)` is a measure of the total strength of connection between nodes `x` and `y`, relative to the other nodes in their region. This takes into account not just direct paths, but also indirect paths, if the links along those paths are of sufficient strength. Since the

[1]The average could be used also but this has no effect on the ranking statistics used in the later experiments.

networks potentially contain a wide variety of relations between terms, the calculation of `dist(x,y)` has access to a wide variety of information linking the two terms. If we consider the *(car, gasoline)* example mentioned earlier, the intuition behind our approach is that these two terms are likely to be closely related in a semantic network built from text, either fairly directly because they appear in the same sentence or document, or indirectly because they are related to the same entities.

5 Experiments

The purpose of the experiments was to develop an entirely automated approach for replicating human judgements of semantic relatedness of words. We used three existing datasets of human judgements: the Hodgson, Rubenstein & Goodenough (RG) and Wordsimilarity-353 (WS-353) datasets. For each dataset we created a corpus using results returned by Google when queried for each word independently (Described in Section 5.2). We then built a semantic network from that corpus and used the spreading activation technique described in the previous section to measure semantic relatedness between the word pairs in the dataset.

The parser and semantic analysis tool used to create the networks were developed on newspaper data (a CCG version of the Penn Treebank (Steedman and Hockenmaier, 2007; Clark and Curran, 2007)), but our impression from informally inspecting the parser output was that the accuracy on the web data was reasonable. The experimental results show that the resulting networks were of high enough quality to closely replicate human judgements.

5.1 The datasets

Many studies have shown a marked priming effect for semantically related words. In his single-word lexical priming study, (Hodgson, 1991) showed that the presentation of a *prime word* such as *election* directly facilitates processing of a target word such as *vote*. Hodgson showed an increase in both response speed and accuracy when the prime and target are semantically related. 143 word pairs were tested across 6 different lexical relations: antonymy (e.g., *enemy*, *friend*); conceptual association (e.g., *bed*, *sleep*); category coordination (e.g., *train*, *truck*); phrasal association (e.g., *private*, *property*); superordination/subordination (e.g., *travel*, *drive*); and synonymy (e.g., *value*, *worth*). It was shown that equivalent priming effects (i.e., reduced processing time) were present across all relation types, thus indicating that priming was a result of the terms' semantic relatedness, not merely their similarity or other simpler relation type.

The Hodgson dataset consists of the 143 word pairs divided by lexical category. There were no scores given as all pairs were shown to have relatively similar priming effects. No examples of unrelated pairs are given in the dataset. We therefore used the unrelated pairs created by McDonald and Brew (2004).

The task in this experiment was to obtain scores for all pairs, and to do an ANOVA test to determine if there is a significant difference between the scores for related and unrelated pairs.

The WS-353 dataset (Finkelstein et al., 2002) contains human rankings of the semantic distance between pairs of terms. Although the name may imply that the scores are based on similarity, human judges were asked to score 353 pairs of words for their *relatedness* on a scale of 1 to 10, and so the dataset is ideal for our purposes. For example, the pair *(money, bank)* is in the dataset and receives a high relatedness score of 8.50, even though the terms are not lexically similar.

The dataset contains regular nouns and named entities, as well as at least one term which does not appear in WordNet (*Maradona*). In this experiment, we calculated scores for all word pairs, and then used rank correlation to compare the similarity of our generated scores to those obtained from human judgements.

The RG dataset (Rubenstein and Goodenough, 1965) is very similar to the WS-353,

though with only 65 word pairs, except that the human judges were asked to judge the pairs based on synonymy, rather than overall relatedness. Thus, for example, the pair *(monk,asylum)*, receives a significantly lower score than the pair *(crane,implement)*.

5.2 Data collection, preparation and processing

In order to create a corpus from which to build the semantic networks, we first extracted each individual word from the pairings, resulting in a list of 440 words for the WS-353 collection, 48 words for the RG (some words were used in multiple pairings), and 282 words for the Hodgson collection. For each of the words in this list, we then performed a query using Google, and downloaded the first 5 page results for that query. The choice of 5 as the number of documents to download for each word was based on a combination of informal intuition about the precision and recall of search engines, as well as the practical issue of obtaining a corpus that could be processed in reasonable space and time.

Each of the downloaded web pages was then cleaned by a set of Perl scripts which removed all HTML markup. Statistics for the resulting corpora are given in Table 1.

Three rules were added to the retrieval process to deal with problems encountered in formatting of web-pages:

1. Pages from which no text could be retrieved were ignored and replaced with the next result.
2. HTML lists preceded by a colon were recombined into sentences.
3. For Wikipedia disambiguation pages (pages which consist of a list of links to articles relating to the various possible senses of a word), all of the listed links were followed and the resulting pages added to the corpus.

Each of these heuristics was performed automatically and without human intervention.

The largest of the networks, created for the WS-353 dataset, took slightly over 24 hours to complete, including time for parsing and semantic analysis.

corpus	sentences	words
Hodgson	814,779	3,745,870
RG	150,165	573,148
WS-353	1,042,128	5,027,947

Table 1: Summary statistics for the corpora generated for the experiments.

6 Results

6.1 Hodgson priming dataset

After processing the Hodgson corpus to build a semantic network with approximately 500,000 nodes and 1,300,000 edges, the appropriate node pairs were fired to obtain the distance measure as previously described. Those measurements were then recorded as measurements of semantic relatedness between two terms. If a term was used as a label in two or more nodes, all nodes were tried, and the highest scoring pairs were used.

As the Hodgson dataset did not provide examples of unrelated pairs against which we could compare, unrelated pairs were generated as described in (McDonald and Brew, 2004). This is not an ideal method, as several pairs that were identified as unrelated did have some relatively obvious relationship (e.g. *tree – house*, *poker – heart*). However we chose to retain the methodology for consistency with previous literature as it was also used in (Padó and Lapata, 2007).

Scores were obtained from the network for the word pairs, and for each target an average score was calculated for all primes in its category. Example scores are given in Table 2.

Two-way analysis of variance (ANOVA) was carried out on the network scores with the the relatedness status of the pair being the independent variable. A reliable effect was observed for the network scores with the primes for related words being significantly larger than those for unrelated words. The results are given in Table 3.

The use of ANOVA shows that there is a

Word pair	Related	Network Score
empty - full	Yes	10.13
coffee - mug	Yes	5.86
horse - goat	Yes	0.96
dog - leash	Yes	4.70
friend - antonym	No	0.53
vote - conceptual	No	1.37
property - phrasal	No	2.47
drive - super/sub	No	1.86

Table 2: Example scores obtained from the network for related and unrelated word pairs from the Hodgson dataset

difference in the scores of the related and unrelated word pairs that cannot be accounted for by random variance. However, in order to compare the strength of the experimental effects between two experiments, additional statistics must be used. Eta-squared (η^2) is a measure of the strength of an experimental effect. A high η^2 indicates that the independent variable accounts for more of the variability, and thus indicates a stronger experimental effect. In our experiments, we found an η^2 of 0.411, which means that approximately 41% of the overall variance can be explained by the relatedness scores.

For comparison, we provide the ANOVA results for experiments by (McDonald and Brew, 2004) and (Padó and Lapata, 2007) on the same dataset. Both of these experiments obtained scores using vector based models populated with data from the BNC.

We also include the results obtained from performing the same ANOVA tests on Pointwise Mutual Information scores collected over our corpus. These results were intended to provide a baseline when using the web-based corpus. To calculate the PMI scores for this experiment, we computed scores for the number of times the two words appeared in the same paragraph or document, and the total number of occurrences of words in the corpus. The PMI scores were calculated by simply dividing the number of times the words co-occurred within a paragraph, by the product of the number of occurrences of each word within a document.

	F	MSE	p	η^2
McDonald & Brew	71.73	0.004	< 0.001	
Padó & Lapata	182.46	0.93	< 0.01	0.332
PMI	42.53	3.79	< 0.001	0.263
Network	50.71	3.28	< 0.0001	0.411

Table 3: ANOVA results of scores generated from the Hodgson dataset compared to those reported for existing systems. (F = F-test statistic, MSE = Mean squared error, p = P-value, η^2 = Effect size)

6.2 ws-353 and rg datasets

The methodology used to obtain scores for the WS-353 and RG collections was identical to that used for the Hodgson data, except that scores were only obtained for those pairs listed in the data set. Because both collections provided direct scores, there was no need to retrieve network scores for unrelated pairings.

	WS-353	RG
WikiRelate!	0.48	0.86
Hughes-Ramage	0.55	0.84
Agirre Et Al	0.66	0.89
PMI	0.41	0.80
Network	0.62	0.86

Table 4: Rank correlation scores for the semantic network and PMI-based approaches, calculated on the WS-353 and RG collections, shown against scores for existing systems.

For consistency with previous literature, the scores obtained by the semantic network were compared with those from the collections using Spearman's rank correlation. The correlation results are given in Table 4. For comparison, we have included the results of the same correlation on scores from three top scoring systems using the approaches described above. We also include the scores obtained by using a simple PMI calculation as in the previous experiment.

The scores obtained by our system were not an improvement on those obtained by existing systems. However, our scores were on par with the best performing systems, which were purpose built for this application, and at least in the case of the system by Agirre et al. used a corpus several orders of magnitude larger.

7 Conclusion

In this paper we have shown that a semantic network approach to determining semantic relatedness of terms can achieve performance on par with the best purpose built systems. This is interesting for two reasons. Firstly, the approach we have taken in this paper is much more analogous to the way humans perform similar tasks. Secondly, the system used was not purpose built for this application. It is instead a general tool for information collection and integration, and this result indicates that it will likely be useful for a wide variety of language processing applications.

References

Agirre, Eneko, Enrique Alfonseca, Keith Hall, Jana Kravalova, Marius Pasca, and Aitor Soroa. 2009. A study on similarity and relatedness using distributional and wordnet-based approaches. In *Proceedings of NAACL-HLT*.

Banerjee, Satanjeev and T. Pedersen. 2003. Extended gloss overlaps as a measure of semantic relatedness. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence*, Acapulco, Mexico.

Bos, Johan, Stephen Clark, Mark Steedman, James R. Curran, and Julia Hockenmaier. 2004. Wide-coverage semantic representations from a CCG parser. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING-04)*, pages 1240–1246, Geneva, Switzerland.

Budanitsky, Alexander and Graeme Hirst. 2006. Evaluating wordnet-based measures of semantic distance. *Computational Linguistics*, 32:13 – 47, March.

Clark, Stephen and James R. Curran. 2007. Wide-coverage efficient statistical parsing with CCG and log-linear models. *Computational Linguistics*, 33(4):493–552.

Collins, Allan M. and Elizabeth F. Loftus. 1975. A spreading-activation theory of semantic processing. *Psychological Review*, 82(6):407–428.

Curran, James R. and Stephen Clark. 2003. Language independent NER using a maximum entropy tagger. In *Proceedings of the Seventh Conference on Natural Language Learning*, pages 164–167, Edmonton, Canada.

Fellbaum, Christiane, editor. 1998. *WordNet : An Electronic Lexical Database*. MIT Press, Cambridge, Mass, USA.

Finkelstein, Lev, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. 2002. Placing search in context: The concept revisited. In *ACM Transactions on Information Systems*, volume 20(1), pages 116–131.

Harrington, Brian and Stephen Clark. 2007. Asknet: Automated semantic knowledge network. In *Proceedings of the 22nd National Conference on Artificial Intelligence (AAAI'07)*, pages 889–894, Vancouver, Canada.

Harrington, Brian and Stephen Clark. 2009. Asknet: Creating and evaluating large scale integrated semantic networks. *International Journal of Semantic Computing*, 2(3):343–364.

Hodgson, James. 1991. Information constraints on pre-lexical priming. *Language and Cognitive Processes*, 6:169 – 205.

Hughes, Thad and Daniel Ramage. 2007. Lexical semantic relatedness with random graph walks. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 581–589, Prague, Czech Republic.

Jiang, J. J. and D. W. Conrath. 1997. Semantic similarity based on corpus statistics and lexical taxonomy. In *International Conference on Research on Computational Linguistics*, Taipei, Taiwan, September.

McDonald, Scott and Chris Brew. 2004. A distributional model of semantic context effects in lexical processing. In *Proceedings of the 42th Annual Meeting of the Association for Computational Linguistics*, pages 17 – 24, Barcelona, Spain.

Mohammad, Saif and Graeme Hirst. 2006. Distributional measures of concept-distance: A task-oriented evaluation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2006)*, Sydney, Australia.

Padó, Sebastian and Mirella Lapata. 2007. Dependency-based construction of semantic space models. *Computational Linguistics*, 33(2):161–199.

Resnik, Philip. 1999. Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. *Journal of Artificial Intelligence Research*, 11:95–130.

Rubenstein, H. and J.B. Goodenough. 1965. Contextual correlates of synonymy. *Computational Linguistics*, 8:627 – 633.

Steedman, Mark and Julia Hockenmaier. 2007. Ccgbank: A corpus of ccg derivations and dependency structures extracted from the penn treebank. *Computational Linguistics*, 33:355–396.

Strube, Michael and Simone Paolo Ponzetto. 2006. Wikirelate! computing semantic relatedness using wikipedia. In *Proceedings of the 21st national conference on Artificial intelligence*, pages 1419–1424. AAAI Press.

Turney, Peter D. 2001. Lecture notes in computer science 1: Mining the web for synonyms: PMI-IR versus LSA on TOEFL.

Conundrums in Unsupervised Keyphrase Extraction: Making Sense of the State-of-the-Art

Kazi Saidul Hasan and **Vincent Ng**
Human Language Technology Research Institute
University of Texas at Dallas
{saidul,vince}@hlt.utdallas.edu

Abstract

State-of-the-art approaches for unsupervised keyphrase extraction are typically evaluated on a single dataset with a single parameter setting. Consequently, it is unclear how effective these approaches are on a new dataset from a different domain, and how sensitive they are to changes in parameter settings. To gain a better understanding of state-of-the-art unsupervised keyphrase extraction algorithms, we conduct a systematic evaluation and analysis of these algorithms on a variety of standard evaluation datasets.

1 Introduction

The keyphrases for a given document refer to a group of phrases that *represent* the document. Although we often come across texts from different domains such as scientific papers, news articles and blogs, which are labeled with keyphrases by the authors, a large portion of the Web content remains untagged. While keyphrases are excellent means for providing a concise summary of a document, recent research results have suggested that the task of automatically identifying keyphrases from a document is by no means trivial. Researchers have explored both supervised and unsupervised techniques to address the problem of automatic keyphrase extraction. Supervised methods typically recast this problem as a binary classification task, where a model is trained on annotated data to determine whether a given phrase is a keyphrase or not (e.g., Frank et al. (1999), Turney (2000; 2003), Hulth (2003), Medelyan et al. (2009)). A disadvantage of supervised approaches is that they require a lot of training data and yet show bias towards the domain on which they are trained, undermining their ability to generalize well to new domains. Unsupervised approaches could be a viable alternative in this regard.

The unsupervised approaches for keyphrase extraction proposed so far have involved a number of techniques, including language modeling (e.g., Tomokiyo and Hurst (2003)), graph-based ranking (e.g., Zha (2002), Mihalcea and Tarau (2004), Wan et al. (2007), Wan and Xiao (2008), Liu et al. (2009a)), and clustering (e.g., Matsuo and Ishizuka (2004), Liu et al. (2009b)). While these methods have been shown to work well on a particular domain of text such as short paper abstracts and news articles, their effectiveness and portability across different domains have remained an unexplored issue. Worse still, each of them is based on a set of assumptions, which may only hold for the dataset on which they are evaluated.

Consequently, *we have little understanding of how effective the state-of the-art systems would be on a completely new dataset from a different domain*. A few questions arise naturally. How would these systems perform on a different dataset with their original configuration? What could be the underlying reasons in case they perform poorly? Is there any system that can generalize fairly well across various domains?

We seek to gain a better understanding of the state of the art in unsupervised keyphrase extraction by examining the aforementioned questions. More specifically, we compare five unsupervised keyphrase extraction algorithms on four corpora with varying domains and statistical characteristics. These algorithms represent the ma-

Coling 2008: Poster Volume, pages 365–373,
Beijing, August 2010

jor directions in this research area, including Tf-Idf and four recently proposed systems, namely, TextRank (Mihalcea and Tarau, 2004), SingleRank (Wan and Xiao, 2008), ExpandRank (Wan and Xiao, 2008), and a clustering-based approach (Liu et al., 2009b). Since none of these systems (except TextRank) are publicly available, we reimplement all of them and make them freely available for research purposes.[1] To our knowledge, this is the *first* attempt to compare the performance of state-of-the-art unsupervised keyphrase extraction systems on multiple datasets.

2 Corpora

Our four evaluation corpora belong to different domains with varying document properties. Table 1 provides an overview of each corpus.

The **DUC-2001** dataset (Over, 2001), which is a collection of 308 news articles, is annotated by Wan and Xiao (2008). We report results on all 308 articles in our evaluation.

The **Inspec** dataset is a collection of 2,000 abstracts from journal papers including the paper title. Each document has two sets of keyphrases assigned by the indexers: the *controlled keyphrases*, which are keyphrases that appear in the *Inspec* thesaurus; and the *uncontrolled keyphrases*, which do not necessarily appear in the thesaurus. This is a relatively popular dataset for automatic keyphrase extraction, as it was first used by Hulth (2003) and later by Mihalcea and Tarau (2004) and Liu et al. (2009b). In our evaluation, we use the set of 500 abstracts designated by these previous approaches as the test set and its set of uncontrolled keyphrases. Note that the average document length for this dataset is the smallest among all our datasets.

The **NUS Keyphrase Corpus** (Nguyen and Kan, 2007) includes 211 scientific conference papers with lengths between 4 to 12 pages. Each paper has one or more sets of keyphrases assigned by its authors and other annotators. We use all the 211 papers in our evaluation. Since the number of annotators can be different for different documents and the annotators are not specified along with the annotations, we decide to take the union of all the gold standard keyphrases from all the sets to construct one single set of annotation for each paper. As Table 1 shows, each NUS paper, both in terms of the average number of tokens (8291) and candidate phrases (2027) per paper, is more than five times larger than any document from any other corpus. Hence, the number of candidate keyphrases that can be extracted is potentially large, making this corpus the most challenging of the four.

Finally, the **ICSI meeting corpus** (Janin et al., 2003), which is annotated by Liu et al. (2009a), includes 161 meeting transcriptions. Following Liu et al., we remove topic segments marked as 'chitchat' and 'digit' from the dataset and use all the remaining segments for evaluation. Each transcript contains three sets of keyphrases produced by the same three human annotators. Since it is possible to associate each set of keyphrases with its annotator, we evaluate each system on this dataset three times, once for each annotator, and report the average score. Unlike the other three datasets, the gold standard keys for the ICSI corpus are mostly unigrams.

3 Unsupervised Keyphrase Extractors

A generic unsupervised keyphrase extraction system typically operates in three steps (Section 3.1), which will help understand the unsupervised systems explained in Section 3.2.

3.1 Generic Keyphrase Extractor

Step 1: Candidate lexical unit selection The first step is to filter out unnecessary word tokens from the input document and generate a list of potential keywords using heuristics. Commonly used heuristics include (1) using a stop word list to remove non-keywords (e.g., Liu et al. (2009b)) and (2) allowing words with certain part-of-speech tags (e.g., nouns, adjectives, verbs) to be considered candidate keywords (Mihalcea and Tarau (2004), Liu et al. (2009a), Wan and Xiao (2008)). In all of our experiments, we follow Wan and Xiao (2008) and select as candidates words with the following Penn Treebank tags: NN, NNS, NNP, NNPS, and JJ, which are obtained using the Stanford POS tagger (Toutanova and Manning, 2000).

[1] See `http://www.hlt.utdallas.edu/~saidul/code.html` for details.

	Corpora			
	DUC-2001	*Inspec*	NUS	ICSI
Type	News articles	Paper abstracts	Full papers	Meeting transcripts
# Documents	308	500	211	161
# Tokens/Document	876	134	8291	1611
# Candidate words/Document	312	57	3271	453
# Candidate phrases/Document	207	34	2027	296
# Tokens/Candidate phrase	1.5	1.7	1.6	1.5
# Gold keyphrases	2484	4913	2327	582
# Gold keyphrases/Document	8.1	9.8	11.0	3.6
U/B/T/O distribution (%)	17/61/18/4	13/53/25/9	27/50/16/7	68/29/2/1
# Tokens/Gold keyphrase	2.1	2.3	2.1	1.3

Table 1: Corpus statistics for the four datasets used in this paper. A candidate word/phrase, typically a sequence of one or more adjectives and nouns, is extracted from the document initially and considered a potential keyphrase. The U/B/T/O distribution indicates how the gold standard keys are distributed among unigrams, bigrams, trigrams, and other higher order n-grams.

Step 2: Lexical unit ranking Once the candidate list is generated, the next task is to rank these lexical units. To accomplish this, it is necessary to build a representation of the input text for the ranking algorithm. Depending on the underlying approach, each candidate word is represented by its syntactic and/or semantic relationship with other candidate words. The relationship can be defined using co-occurrence statistics, external resources (e.g., neighborhood documents, Wikipedia), or other syntactic clues.

Step 3: Keyphrase formation In the final step, the ranked list of candidate words is used to form keyphrases. A candidate phrase, typically a sequence of nouns and adjectives, is selected as a keyphrase if (1) it includes one or more of the top-ranked candidate words (Mihalcea and Tarau (2004), Liu et al. (2009b)), or (2) the sum of the ranking scores of its constituent words makes it a top scoring phrase (Wan and Xiao, 2008).

3.2 The Five Keyphrase Extractors

As mentioned above, we re-implement five unsupervised approaches for keyphrase extraction. Below we provide a brief overview of each system.

3.2.1 Tf-Idf

Tf-Idf assigns a score to each term t in a document d based on t's frequency in d (term frequency) and how many other documents include t (inverse document frequency) and is defined as:

$$\text{tfidf}_t = \text{tf}_t \times log(D/D_t) \quad (1)$$

where D is the total number of documents and D_t is the number of documents containing t.

Given a document, we first compute the Tf-Idf score of each candidate word (see Step 1 of the generic algorithm). Then, we extract all the longest n-grams consisting of candidate words and score each n-gram by summing the Tf-Idf scores of its constituent unigrams. Finally, we output the top N n-grams as keyphrases.

3.2.2 TextRank

In the TextRank algorithm (Mihalcea and Tarau, 2004), a text is represented by a graph. Each vertex corresponds to a word type. A weight, w_{ij}, is assigned to the edge connecting the two vertices, v_i and v_j, and its value is the number of times the corresponding word types co-occur within a window of W words in the associated text. The goal is to (1) compute the score of each vertex, which reflects its *importance*, and then (2) use the word types that correspond to the highest-scored vertices to form keyphrases for the text. The score for v_i, $S(v_i)$, is initialized with a default value and is computed in an iterative manner until convergence using this recursive formula:

$$S(v_i) = (1-d) + d \times \sum_{v_j \epsilon Adj(v_i)} \frac{w_{ji}}{\sum_{v_k \epsilon Adj(v_j)} w_{jk}} S(v_j) \quad (2)$$

where $Adj(v_i)$ denotes v_i's neighbors and d is the damping factor set to 0.85 (Brin and Page, 1998). Intuitively, a vertex will receive a high score if it has many high-scored neighbors. As noted before, after convergence, the T% top-scored vertices are

selected as keywords. Adjacent keywords are then collapsed and output as a keyphrase.

According to Mihalcea and Tarau (2004), TextRank's best score on the *Inspec* dataset is achieved when only nouns and adjectives are used to create a uniformly weighted graph for the text under consideration, where an edge connects two word types only if they co-occur within a window of two words. Hence, our implementation of TextRank follows this configuration.

3.2.3 SingleRank

SingleRank (Wan and Xiao, 2008) is essentially a TextRank approach with three major differences. First, while each edge in a TextRank graph (in Mihalcea and Tarau's implementation) has the same weight, each edge in a SingleRank graph has a weight equal to the number of times the two corresponding word types co-occur. Second, while in TextRank only the word types that correspond to the top-ranked vertices can be used to form keyphrases, in SingleRank, we do not filter out any low-scored vertices. Rather, we (1) score each candidate keyphrase, which can be any longest-matching sequence of nouns and adjectives in the text under consideration, by summing the scores of its constituent word types obtained from the SingleRank graph, and (2) output the N highest-scored candidates as the keyphrases for the text. Finally, SingleRank employs a window size of 10 rather than 2.

3.2.4 ExpandRank

ExpandRank (Wan and Xiao, 2008) is a TextRank extension that exploits neighborhood knowledge for keyphrase extraction. For a given document d, the approach first finds its k nearest neighboring documents from the accompanying document collection using a similarity measure (e.g., cosine similarity). Then, the graph for d is built using the co-occurrence statistics of the candidate words collected from the document itself and its k nearest neighbors.

Specifically, each document is represented by a term vector where each vector dimension corresponds to a word type present in the document and its value is the Tf-Idf score of that word type for that document. For a given document d_0, its k nearest neighbors are identified, and together they form a larger document set of k+1 documents, $D = \{d_0, d_1, d_2, ..., d_k\}$. Given this document set, a graph is constructed, where each vertex corresponds to a candidate word type in D, and each edge connects two vertices v_i and v_j if the corresponding word types co-occur within a window of W words in the document set. The weight of an edge, $w(v_i, v_j)$, is computed as follows:

$$w(v_i, v_j) = \sum_{d_k \epsilon D} sim(d_0, d_k) \times freq_{d_k}(v_i, v_j) \quad (3)$$

where $sim(d_0, d_k)$ is the cosine similarity between d_0 and d_k, and $freq_{d_k}(v_i, v_j)$ is the co-occurrence frequency of v_i and v_j in document d_k. Once the graph is constructed, the rest of the procedure is identical to SingleRank.

3.2.5 Clustering-based Approach

Liu et al. (2009b) propose to cluster candidate words based on their semantic relationship to ensure that the extracted keyphrases *cover* the entire document. The objective is to have each cluster represent a unique aspect of the document and take a representative word from each cluster so that the document is covered from all aspects.

More specifically, their algorithm (henceforth referred to as KeyCluster) first filters out the stop words from a given document and treats the remaining unigrams as candidate words. Second, for each candidate, its relatedness with another candidate is computed by (1) counting how many times they co-occur within a window of size W in the document and (2) using Wikipedia-based statistics. Third, candidate words are clustered based on their relatedness with other candidates. Three clustering algorithms are used of which spectral clustering yields the best score. Once the clusters are formed, one representative word, called an exemplar term, is picked from each cluster. Finally, KeyCluster extracts from the document all the longest n-grams starting with zero or more adjectives and ending with one or more nouns, and if such an n-gram includes one or more exemplar words, it is selected as a keyphrase. As a post-processing step, a frequent word list generated from Wikipedia is used to filter out the frequent unigrams that are selected as keyphrases.

4 Evaluation

4.1 Experimental Setup

TextRank and SingleRank setup Following Mihalcea and Tarau (2004) and Wan and Xiao (2008), we set the co-occurrence window size for TextRank and SingleRank to 2 and 10, respectively, as these parameter values have yielded the best results for their evaluation datasets.

ExpandRank setup Following Wan and Xiao (2008), we find the 5 nearest neighbors for each document from the remaining documents in the same corpus. The other parameters are set in the same way as in SingleRank.

KeyCluster setup As argued by Liu et al. (2009b), Wikipedia-based relatedness is computationally expensive to compute. As a result, we follow them by computing the *co-occurrence-based* relatedness instead, using a window of size 10. Then, we cluster the candidate words using spectral clustering, and use the frequent word list that they generously provided us to post-process the resulting keyphrases by filtering out those that are frequent unigrams.

4.2 Results and Discussion

In an attempt to gain a better insight into the five unsupervised systems, we report their performance in terms of precision-recall curves for each of the four datasets (see Figure 1). This contrasts with essentially all previous work, where the performance of a keyphrase extraction system is reported in terms of an F-score obtained via a particular parameter setting on a particular dataset. We generate the curves for each system as follows. For Tf-Idf, SingleRank, and ExpandRank, we vary the number of keyphrases, N, predicted by each system. For TextRank, instead of varying the number of predicted keyphrases, we vary T, the percentage of top-scored vertices (i.e., unigrams) that are selected as keywords at the end of the ranking step. The reason is that TextRank only imposes a ranking on the unigrams but not on the keyphrases generated from the high-ranked unigrams. For KeyCluster, we vary the number of clusters produced by spectral clustering rather than the number of predicted keyphrases, again because KeyCluster does not impose a ranking on the resulting keyphrases. In addition, to give an estimate of how each system performs in terms of F-score, we also plot curves corresponding to different F-scores in these graphs.

Tf-Idf Consistent with our intuition, the precision of Tf-Idf drops as recall increases. Although it is the simplest of the five approaches, Tf-Idf is the best performing system on all but the *Inspec* dataset, where TextRank and KeyCluster beat Tf-Idf on just a few cases. It clearly outperforms all other systems for NUS and ICSI.

TextRank The TextRank curves show a different progression than Tf-Idf: precision does not drop as much when recall increases. For instance, in case of DUC and ICSI, precision is not sensitive to changes in recall. Perhaps somewhat surprisingly, its precision increases with recall for *Inspec*, allowing it to even reach a point (towards the end of the curve) where it beats Tf-Idf. While additional experiments are needed to determine the reason for this somewhat counter-intuitive result, we speculate that this may be attributed to the fact that the TextRank curves are generated by progressively increasing T (i.e., the percentage of top-ranked vertices/unigrams that are used to generate keyphrases) rather than the number of predicted keyphrases, as mentioned before. Increasing T does not necessarily imply an increase in the number of predicted keyphrases, however. To see the reason, consider an example in which we want TextRank to extract the keyphrase "advanced machine learning" for a given document. Assume that TextRank ranks the unigrams "advanced", "learning", and "machine" first, second, and third, respectively in its ranking step. When $T = \frac{2}{n}$, where n denotes the total number of candidate unigrams, only the two highest-ranked unigrams (i.e., "advanced" and "learning") can be used to form keyphrases. This implies that "advanced" and "learning" will each be predicted as a keyphrase, but "advanced machine learning" will not. However, when $T = \frac{3}{n}$, all three unigrams can be used to form a keyphrase, and since TextRank collapses unigrams adjacent to each other in the text to form a keyphrase, it will correctly predict "advanced machine learning" as a keyphrase. Note that as we increase T from $\frac{2}{n}$ to $\frac{3}{n}$, recall increases, and so does precision, since

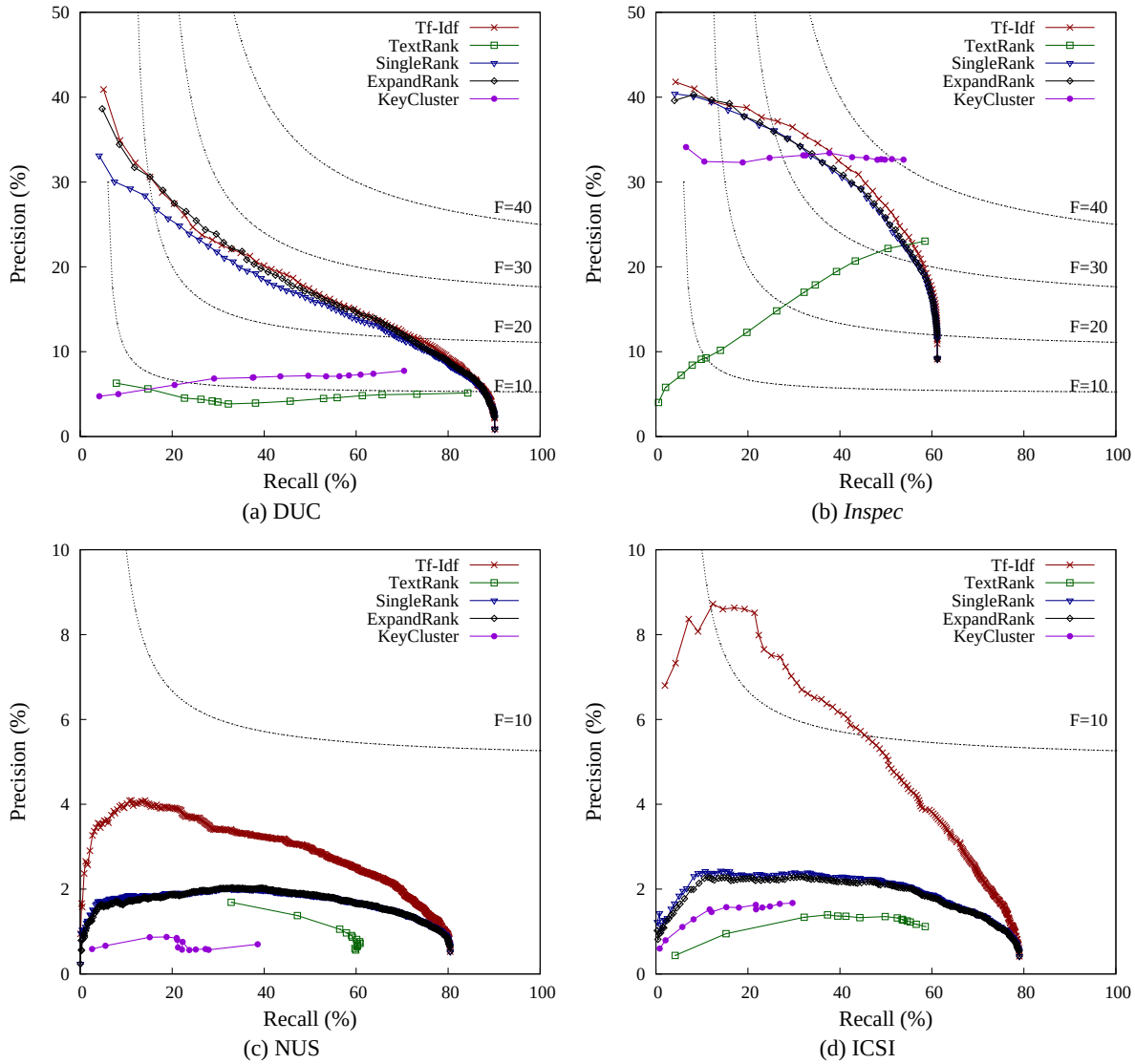


Figure 1: Precision-recall curves for all four datasets

"advanced" and "learning" are now combined to form one keyphrase (and hence the number of predicted keyphrases decreases). In other words, it is possible to see a simultaneous rise in precision and recall in a TextRank curve. A natural question is: why does is happen only for *Inspec* but not the other datasets? The reason could be attributed to the fact that *Inspec* is composed of abstracts: since the number of keyphrases that can be generated from these short documents is relatively small, precision may not drop as severely as with the other datasets even when all of the unigrams are used to form keyphrases.

On average, TextRank performs much worse compared to Tf-Idf. The curves also prove TextRank's sensitivity to T on *Inspec*, but not on the other datasets. This certainly gives more insight into TextRank since it was evaluated on *Inspec* only for T=33% by Mihalcea and Tarau (2004).

SingleRank SingleRank, which is supposed to be a simple variant of TextRank, surprisingly exhibits very different performance. First, it shows a more intuitive nature: precision drops as recall increases. Second, SingleRank outperforms TextRank by big margins on all the datasets. Later, we will examine which of the differences between them is responsible for the differing performance.

	DUC		*Inspec*		NUS		ICSI	
	Parameter	F	Parameter	F	Parameter	F	Parameter	F
Tf-Idf	N = 14	27.0	N = 14	36.3	N = 60	6.6	N = 9	12.1
TextRank	T = 100%	9.7	T = 100%	33.0	T = 5%	3.2	T = 25%	2.7
SingleRank	N = 16	25.6	N = 15	35.3	N = 190	3.8	N = 50	4.4
ExpandRank	N = 13	26.9	N = 15	35.3	N = 177	3.8	N = 51	4.3
KeyCluster	$m = 0.9n$	14.0	$m = 0.9n$	40.6	$m = 0.25n$	1.7	$m = 0.9n$	3.2

Table 2: Best parameter settings. N is the number of predicted keyphrases, T is the percentage of vertices selected as keywords in TextRank, m is the number of clusters in KeyCluster, expressed in terms of n, the fraction of candidate words.

ExpandRank Consistent with Wan and Xiao (2008), ExpandRank beats SingleRank on DUC when a small number of phrases are predicted, but their difference diminishes as more phrases are predicted. On the other hand, their performance is indistinguishable from each other on the other three datasets. A natural question is: why does ExpandRank improve over SingleRank only for DUC but not for the other datasets? To answer this question, we look at the DUC articles and find that in many cases, the 5-nearest neighbors of a document are on the same topic involving the same entities as the document itself, presumably because many of these news articles are simply updated versions of an evolving event. Consequently, the graph built from the neighboring documents is helpful for predicting the keyphrases of the given document. Such topic-wise similarity among the nearest documents does not exist in the other datasets, however.

KeyCluster As in TextRank, KeyCluster does not always yield a drop in precision as recall improves. This, again, may be attributed to the fact that the KeyCluster curves are generated by varying the number of clusters rather than the number of predicted keyphrases, as well as the way keyphrases are formed from the exemplars. Another reason is that the frequent Wikipedia unigrams are excluded during post-processing, making KeyCluster more resistant to precision drops. Overall, KeyCluster performs slightly better than TextRank on DUC and ICSI, yields the worst performance on NUS, and scores the best on *Inspec* when the number of clusters is high. These results seem to suggest that KeyCluster works better if more clusters are used.

Best parameter settings Table 2 shows for each system the parameter values yielding the best F-score on each dataset. Two points deserve mention. First, in comparison to SingleRank and ExpandRank, Tf-Idf outputs fewer keyphrases to achieve its best F-score on most datasets. Second, the systems output more keyphrases on NUS than on other datasets to achieve their best F-scores (e.g., 60 for Tf-Idf, 190 for SingleRank, and 177 for ExpandRank). This can be attributed in part to the fact that the F-scores on NUS are low for all the systems and exhibit only slight changes as we output more phrases.

Our re-implementations Do our duplicated systems yield scores that match the original scores? Table 3 sheds light on this question.

First, consider KeyCluster, where our score lags behind the original score by approximately 5%. An examination of Liu et al.'s (2009b) results reveals a subtle caveat in keyphrase extraction evaluations. In *Inspec*, not all gold-standard keyphrases appear in their associated document, and as a result, none of the five systems we consider in this paper can achieve a recall of 100. While Mihalcea and Tarau (2004) and our reimplementations use *all* of these gold-standard keyphrases in our evaluation, Hulth (2003) and Liu et al. address this issue by using as gold-standard keyphrases only those that appear in the corresponding document when computing recall.[2] This explains why our KeyCluster score (38.9) is lower than the original score (43.6). If we follow Liu et al.'s way of computing recall, our reimplementation score goes up to 42.4, which lags behind their score by only 1.2.

Next, consider TextRank, where our score lags behind Mihalcea and Tarau's original score by more than 25 points. We verified our implementation against a publicly available implementation

[2] As a result, Liu et al. and Mihalcea and Tarau's scores are not directly comparable, but Liu et al. did not point this out while comparing scores in their paper.

	Dataset	F-score	
		Original	Ours
Tf-Idf	DUC	25.4	25.7
TextRank	*Inspec*	36.2	10.0
SingleRank	DUC	27.2	24.9
ExpandRank	DUC	31.7	26.4
KeyCluster	*Inspec*	43.6	38.9

Table 3: Original vs. re-implementation scores

of TextRank[3], and are confident that our implementation is correct. It is also worth mentioning that using our re-implementation of SingleRank, we are able to match the best scores reported by Mihalcea and Tarau (2004) on *Inspec*.

We score 2 and 5 points less than Wan and Xiao's (2008) implementations of SingleRank and ExpandRank, respectively. We speculate that document pre-processing (e.g., stemming) has contributed to the discrepancy, but additional experiments are needed to determine the reason.

SingleRank vs. TextRank Figure 1 shows that SingleRank behaves very differently from TextRank. As mentioned in Section 3.2.3, the two algorithms differ in three major aspects. To determine which aspect is chiefly responsible for the large difference in their performance, we conduct three "ablation" experiments. Each experiment modifies exactly one of these aspects in SingleRank so that it behaves like TextRank, effectively ensuring that the two algorithms differ only in the remaining two aspects. More specifically, in the three experiments, we (1) change SingleRank's window size to 2, (2) build an unweighted graph for SingleRank, and (3) incorporate TextRank's way of forming keyphrases into SingleRank, respectively. Figure 2 shows the resultant curves along with the SingleRank and TextRank curves on *Inspec* taken from Figure 1b. As we can see, the way of forming phrases, rather than the window size or the weight assignment method, has the largest impact on the scores. In fact, after incorporating TextRank's way of forming phrases, SingleRank exhibits a remarkable drop in performance, yielding a curve that resembles the TextRank curve. Also note that SingleRank achieves better recall values than TextRank. To see the reason, recall that TextRank requires that every word of a gold keyphrase must appear among the top-ranked unigrams. This is a fairly strict criterion, especially in comparison to SingleRank, which does not require all unigrams of a gold keyphrase to be present in the top-ranked list. We observe similar trends for the other datasets.

[3] `http://github.com/sharethis/textrank`

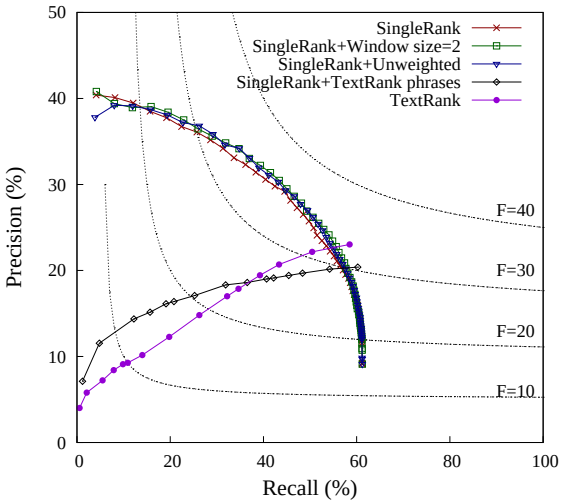


Figure 2: Ablation results for SingleRank on *Inspec*

5 Conclusions

We have conducted a systematic evaluation of five state-of-the-art unsupervised keyphrase extraction algorithms on datasets from four different domains. Several conclusions can be drawn from our experimental results. First, to fully understand the strengths and weaknesses of a keyphrase extractor, it is essential to evaluate it on multiple datasets. In particular, evaluating it on a single dataset has proven inadequate, as good performance can sometimes be achieved due to certain statistical characteristics of a dataset. Second, as demonstrated by our experiments with TextRank and SingleRank, post-processing steps such as the way of forming keyphrases can have a large impact on the performance of a keyphrase extractor. Hence, it may be worthwhile to investigate alternative methods for extracting candidate keyphrases (e.g., Kumar and Srinathan (2008), You et al. (2009)). Finally, despite the large amount of recent work on unsupervised keyphrase extractor, our results indicated that Tf-Idf remains a strong baseline, offering very robust performance across different datasets.

Acknowledgments

We thank the three anonymous reviewers for their comments. Many thanks to Anette Hulth and Yang Liu for providing us with the *Inspec* and ICSI datasets; Rada Mihalcea, Paco Nathan, and Xiaojun Wan for helping us understand their algorithms/implementations; and Peng Li for providing us with the frequent word list that he and his co-authors used in their paper. This work was supported in part by NSF Grant IIS-0812261.

References

Brin, Sergey and Lawrence Page. 1998. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks*, 30(1–7):107–117.

Frank, Eibe, Gordon W. Paynter, Ian H. Witten, Carl Gutwin, and Craig G. Nevill-Manning. 1999. Domain-specific keyphrase extraction. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence*, pages 668–673.

Hulth, Anette. 2003. Improved automatic keyword extraction given more linguistic knowledge. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, pages 216–223.

Janin, Adam, Don Baron, Jane Edwards, Dan Ellis, David Gelbart, Nelson Morgan, Barbara Peskin, Thilo Pfau, Elizabeth Shriberg, Andreas Stolcke, and Chuck Wooters. 2003. The ICSI meeting corpus. In *Proceedings of 2003 IEEE Conference on Acoustics, Speech, and Signal Processing*, pages 364–367.

Kumar, Niraj and Kannan Srinathan. 2008. Automatic keyphrase extraction from scientific documents using n-gram filtration technique. In *Proceedings of the Eighth ACM Symposium on Document Engineering*, pages 199–208.

Liu, Feifan, Deana Pennell, Fei Liu, and Yang Liu. 2009a. Unsupervised approaches for automatic keyword extraction using meeting transcripts. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 620–628.

Liu, Zhiyuan, Peng Li, Yabin Zheng, and Maosong Sun. 2009b. Clustering to find exemplar terms for keyphrase extraction. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 257–266.

Matsuo, Yutaka and Mitsuru Ishizuka. 2004. Keyword extraction from a single document using word co-occurrence statistical information. *International Journal on Artificial Intelligence Tools*, 13(1):157–169.

Medelyan, Olena, Eibe Frank, and Ian H. Witten. 2009. Human-competitive tagging using automatic keyphrase extraction. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1318–1327.

Mihalcea, Rada and Paul Tarau. 2004. TextRank: Bringing order into texts. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411.

Nguyen, Thuy Dung and Min-Yen Kan. 2007. Keyphrase extraction in scientific publications. In *Proceedings of the International Conference on Asian Digital Libraries*, pages 317–326.

Over, Paul. 2001. Introduction to DUC-2001: An intrinsic evaluation of generic news text summarization systems. In *Proceedings of the 2001 Document Understanding Conference*.

Tomokiyo, Takashi and Matthew Hurst. 2003. A language model approach to keyphrase extraction. In *Proceedings of the ACL Workshop on Multiword Expressions*.

Toutanova, Kristina and Christopher D. Manning. 2000. Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In *Proceedings of the 2000 Joint SIGDAT Conference on Empirical Methods in Natural Language processing and Very Large Corpora*, pages 63–70.

Turney, Peter. 2000. Learning algorithms for keyphrase extraction. *Information Retrieval*, 2:303–336.

Turney, Peter. 2003. Coherent keyphrase extraction via web mining. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, pages 434–439.

Wan, Xiaojun and Jianguo Xiao. 2008. Single document keyphrase extraction using neighborhood knowledge. In *Proceedings of the 23rd AAAI Conference on Artificial Intelligence*, pages 855–860.

Wan, Xiaojun, Jianwu Yang, and Jianguo Xiao. 2007. Towards an iterative reinforcement approach for simultaneous document summarization and keyword extraction. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 552–559.

You, Wei, Dominique Fontaine, and Jean-Paul Barthès. 2009. Automatic keyphrase extraction with a refined candidate set. In *Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology*, pages 576–579.

Zha, Hongyuan. 2002. Generic summarization and keyphrase extraction using mutual reinforcement principle and sentence clustering. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 113–120.

Integrating N-best SMT Outputs into a TM System

Yifan He **Yanjun Ma** **Andy Way** **Josef van Genabith**
Centre for Next Generation Localisation
School of Computing
Dublin City University
{yhe,yma,away,josef}@computing.dcu.ie

Abstract

In this paper, we propose a novel framework to enrich Translation Memory (TM) systems with Statistical Machine Translation (SMT) outputs using ranking. In order to offer the human translators multiple choices, instead of only using the top SMT output and top TM hit, we merge the N-best output from the SMT system and the k-best hits with highest fuzzy match scores from the TM system. The merged list is then ranked according to the prospective post-editing effort and provided to the translators to aid their work. Experiments show that our ranked output achieve 0.8747 precision at top 1 and 0.8134 precision at top 5. Our framework facilitates a tight integration between SMT and TM, where full advantage is taken of TM while high quality SMT output is availed of to improve the productivity of human translators.

1 Introduction

Translation Memories (TM) are databases that store translated segments. They are often used to assist translators and post-editors in a Computer Assisted Translation (CAT) environment by returning the most similar translated segments. Professional post-editors and translators have long been relying on TMs to avoid duplication of work in translation.

With the rapid development in statistical machine translation (SMT), MT systems are beginning to generate acceptable translations, especially in domains where abundant parallel corpora exist. It is thus natural to ask if these translations can be utilized in some way to enhance TMs.

However advances in MT are being adopted only slowly and sometimes somewhat reluctantly in professional localization and post-editing environments because of 1) the usefulness of the TM, 2) the investment and effort the company has put into TMs, and 3) the lack of robust SMT confidence estimation measures which are as reliable as fuzzy match scores (cf. Section 4.1.2) used in TMs. Currently the localization industry relies on TM fuzzy match scores to obtain both a good approximation of post-editing effort and an estimation of the overall translation cost.

In a forthcoming paper, we propose a translation recommendation model to better integrate MT outputs into a TM system. Using a binary classifier, we only recommend an MT output to the TM-user when the classifier is highly confident that it is better than the TM output. In this framework, post-editors continue to work with the TM while benefiting from (better) SMT outputs; the assets in TMs are not wasted and TM fuzzy match scores can still be used to estimate (the upper bound of) post-editing labor.

In the previous work, the binary predictor works on the 1-best output of the MT and TM systems, presenting either the one or the other to the post-editor. In this paper, we develop the idea further by moving from binary prediction to ranking. We use a ranking model to merge the k-best lists of the two systems, and produce a ranked merged

Coling 2008: Poster Volume, pages 374–382,
Beijing, August 2010

list for post-editing. As the list is an enriched version of the TM's k-best list, the TM related assets are better preserved and the cost estimation is still valid as an upper bound.

More specifically, we recast SMT-TM integration as a ranking problem, where we apply the Ranking SVM technique to produce a ranked list of translations combining the k-best lists of both the MT and the TM systems. We use features independent of the MT and TM systems for ranking, so that outputs from MT and TM can have the same set of features. Ideally the translations should be ranked by their associated post-editing efforts, but given the very limited amounts of human annotated data, we use an automatic MT evaluation metric, TER (Snover et al., 2006), which is specifically designed to simulate post-editing effort to train and test our ranking model.

The rest of the paper is organized as follows: we first briefly introduce related research in Section 2, and review Ranking SVMs in Section 3. The formulation of the problem and experiments with the ranking models are presented in Sections 4 and 5. We analyze the post-editing effort approximated by the TER metric in Section 6. Section 7 concludes and points out avenues for future research.

2 Related Work

There has been some work to help TM users to apply MT outputs more smoothly. One strand is to improve the MT confidence measures to better predict post-editing effort in order to obtain a quality estimation that has the potential to replace the fuzzy match score in the TM. To the best of our knowledge, the first paper in this area is (Specia et al., 2009a), which uses regression on both the automatic scores and scores assigned by post-editors. The method is improved in (Specia et al., 2009b), which applies Inductive Confidence Machines and a larger set of features to model post-editors' judgment of the translation quality between 'good' and 'bad', or among three levels of post-editing effort.

Another strand is to integrate high confidence MT outputs into the TM, so that the 'good' TM entries will remain untouched. In our forthcoming paper, we recommend SMT outputs to a TM user when a binary classifier predicts that SMT outputs are more suitable for post-editing for a particular sentence.

The research presented here continues the line of research in the second strand. The difference is that we do not limit ourselves to the 1-best output but try to produce a k-best output in a ranking model. The ranking scheme also enables us to show all TM hits to the user, and thus further protects the TM assets.

There has also been work to improve SMT using the knowledge from the TM. In (Simard and Isabelle, 2009), the SMT system can produce a better translation when there is an exact or close match in the corresponding TM. They use regression Support Vector Machines to model the quality of the TM segments. This is also related to our work in spirit, but our work is in the opposite direction, i.e. using SMT to enrich TM.

Moreover, our ranking model is related to reranking (Shen et al., 2004) in SMT as well. However, our method does not focus on producing better 1-best translation output for an SMT system, but on improving the overall quality of the k-best list that TM systems present to post-editors. Some features in our work are also different in nature to those used in MT reranking. For instance we cannot use N-best posterior scores as they do not make sense for the TM outputs.

3 The Support Vector Machines

3.1 The SVM Classifier

Classical SVMs (Cortes and Vapnik, 1995) are binary classifiers that classify an input instance based on decision rules which minimize the regularized error function in (Eq. 1):

$$\begin{array}{ll} \min\limits_{w,b,\xi} & \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum\limits_{i=1}^{l}\xi_i \\ \text{subject to:} & y_i(\mathbf{w}^T\mathbf{x}_i + b) \geqslant 1 - \xi_i \\ & \xi_i \geqslant 0 \end{array} \quad (1)$$

where $(\mathbf{x}_i, y_i) \in R^n \times \{1, -1\}$ are l training instances. $\mathbf{w}$ is the weight vector, ξ is the relaxation variable and $C > 0$ is the penalty parameter.

3.2 Ranking SVM for SMT-TM Integration

The SVM classification algorithm is extended to the ranking case in (Joachims, 2002). For a cer-

tain group of instances, the Ranking SVM aims at producing a ranking r that has the maximum Kendall's τ coefficient with the the gold standard ranking r^*.

Kendall's τ measures the relevance of two rankings: $\tau(r_a, r_b) = \frac{P-Q}{P+Q}$, where P and Q are the amount of concordant and discordant pairs in r_a and r_b. In practice, this is done by building constraints to minimize the discordant pairs Q. Following the basic idea, we show how Ranking SVM can be applied to MT-TM integration as follows.

Assume that for each source sentence s, we have a set of outputs from MT, $\mathbf{M}$ and a set of outputs from TM, $\mathbf{T}$. If we have a ranking $r(s)$ over translation outputs $\mathbf{M} \bigcup \mathbf{T}$ where for each translation output $d \in \mathbf{M} \bigcup \mathbf{T}$, $(d_i, d_j) \in r(s)$ iff $d_i <_{r(s)} d_j$, we can rewrite the ranking constraints as optimization constraints in an SVM, as in Eq. (2).

$$\begin{array}{l} \min\limits_{w,b,\xi} \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum\xi \\ \text{subject to:} \\ \forall(d_i,d_j) \in r(s_1) : \mathbf{w}(\Phi(s_1,d_i) - \Phi(s_1,d_j)) \geqslant 1 - \xi_{i,j,1} \\ ... \\ \forall(d_i,d_j) \in r(s_n) : \mathbf{w}(\Phi(s_n,d_i) - \Phi(s_n,d_j)) \geqslant 1 - \xi_{i,j,n} \\ \xi_{i,j,k} \geqslant 0 \end{array} \quad (2)$$

where $\Phi(s_n, d_i)$ is a feature vector of translation output d_i given source sentence s_n. The Ranking SVM minimizes the discordant number of rankings with the gold standard according to Kendall's τ.

When the instances are not linearly separable, we use a mapping function ϕ to map the features $\mathbf{x}_i$ ($\Phi(s_n, d_i)$ in the case of ranking) to high dimensional space, and solve the SVM with a kernel function K in where $K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^T\phi(\mathbf{x}_j)$.

We perform our experiments with the Radial Basis Function (RBF) kernel, as in Eq. (3).

$$K(\mathbf{x}_i, \mathbf{x}_j) = exp(-\gamma||\mathbf{x}_i - \mathbf{x}_j||^2), \gamma > 0 \quad (3)$$

4 The Ranking-based Integration Model

In this section we present the Ranking-based SMT-TM integration model in detail. We first introduce the k-best lists in MT (called N-best list) and TM systems (called m-best list in this section) and then move on to the problem formulation and the feature set.

4.1 K-Best Lists in SMT and TM

4.1.1 The SMT N-best List

The N-best list of the SMT system is generated during decoding according to the internal feature scores. The features include language and translation model probabilities, reordering model scores and a word penalty.

4.1.2 The TM M-Best List and the Fuzzy Match Score

The m-best list of the TM system is generated in descending fuzzy match score. The fuzzy match score (Sikes, 2007) uses the similarity of the source sentences to predict a level to which a translation is reusable or editable.

The calculation of fuzzy match scores is one of the core technologies in TM systems and varies among different vendors. We compute fuzzy match cost as the minimum Edit Distance (Levenshtein, 1966) between the source and TM entry, normalized by the length of the source as in Eq. (4), as most of the current implementations are based on edit distance while allowing some additional flexible matching.

$$FuzzyMatch(t) = \min_e \frac{EditDistance(s,e)}{Len(s)} \quad (4)$$

where s is the source side of the TM hit t, and e is the source side of an entry in the TM.

4.2 Problem Formulation

Ranking lists is a well-researched problem in the information retrieval community, and Ranking SVMs (Joachims, 2002), which optimizes on the ranking correlation τ have already been applied successfully in machine translation evaluation (Ye et al., 2007). We apply the same method here to rerank a merged list of MT and TM outputs.

Formally given an MT-produced N-best list $\mathbf{M} = \{m_1, m_2, ..., m_n\}$, a TM-produced m-best list $\mathbf{T} = \{t_1, t_2, ..., t_m\}$ for a input sentence s, we define the gold standard using the TER metric (Snover et al., 2006): for each $d \in \mathbf{M} \bigcup \mathbf{T}$, $(d_i, d_j) \in r(s)$ iff $TER(d_i) < TER(d_j)$. We train and test a Ranking SVM using cross validation on a data set created according to this criterion. Ideally the gold standard would be created by human annotators. We choose to use TER

as large-scale annotation is not yet available for this task. Furthermore, TER has a high correlation with the HTER score (Snover et al., 2006), which is the TER score using the post-edited MT output as a reference, and is used as an estimation of post-editing effort.

4.3 The Feature Set

When building features for the Ranking SVM, we are limited to features that are independent of the MT and TM system. We experiment with system-independent fluency and fidelity features below, which capture translation fluency and adequacy, respectively.

4.3.1 Fluency Features

Source-side Language Model Scores. We compute the LM probability and perplexity of the input source sentence on a language model trained on the source-side training data of the SMT system, which is also the TM database. The inputs that have lower perplexity on this language model are more similar to the data set on which the SMT system is built.

Target-side Language Model Scores. We compute the LM probability and perplexity as a measure of the fluency of the translation.

4.3.2 Fidelity Features

The Pseudo-Source Fuzzy Match Score. We translate the output back to obtain a pseudo source sentence. We compute the fuzzy match score between the original source sentence and this pseudo-source. If the MT/TM performs well enough, these two sentences should be the same or very similar. Therefore the fuzzy match score here gives an estimation of the confidence level of the output.

The IBM Model 1 Score. We compute the IBM Model 1 score in both directions to measure the correspondence between the source and target, as it serves as a rough estimation of how good a translation it is on the word level.

5 Experiments

5.1 Experimental Settings

5.1.1 Data

Our raw data set is an English–French translation memory with technical translation from a multi-national IT security company, consisting of 51K sentence pairs. We randomly select 43K to train an SMT system and translate the English side of the remaining 8K sentence pairs, which is used to run cross validation. Note that the 8K sentence pairs are from the same TM, so that we are able to create a gold standard by ranking the TER scores of the MT and TM outputs.

Duplicated sentences are removed from the data set, as those will lead to an exact match in the TM system and will not be translated by translators. The average sentence length of the training set is 13.5 words and the size of the training set is comparable to the (larger) translation memories used in the industry.

5.1.2 SMT and TM systems

We use a standard log-linear PB-SMT model (Och and Ney, 2002): GIZA++ implementation of IBM word alignment model 4, the phrase-extraction heuristics described in (Koehn et al., 2003), minimum-error-rate training (Och, 2003), a 5-gram language model with Kneser-Ney smoothing trained with SRILM (Stolcke, 2002) on the English side of the training data, and Moses (Koehn et al., 2007) to decode. We train a system in the opposite direction using the same data to produce the pseudo-source sentences.

We merge distinct 5-best lists from MT and TM systems to produce a new ranking. To create the distinct list for the SMT system, we search over a 100-best list and keep the top-5 distinct outputs. Our data set consists of mainly short sentences, leading to many duplications in the N-best output of the SMT decoder. In such cases, top-5 distinct outputs are good representations of the SMT's output.

5.2 Training, Tuning and Testing the Ranking SVM

We run training and prediction of the Ranking SVM in 4-fold cross validation. We use the

SVM*light*[1] toolkit to perform training and testing.

When using the Ranking SVM with the RBF kernel, we have two free parameters to tune on: the cost parameter C in Eq. (1) and the radius parameter γ in Eq. (3). We optimize C and γ using a brute-force grid search before running cross-validation and maximize precision at top-5, with an inner 3-fold cross validation on the (outer) Fold-1 training set. We search within the range $[2^{-6}, 2^{9}]$, the step size is 2 on the exponent.

5.3 The Gold Standard

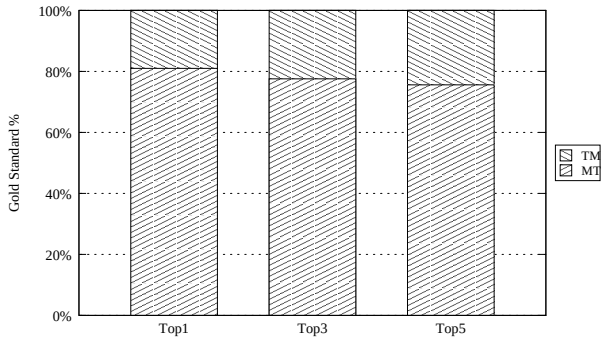


Figure 1: MT and TM's percentage in gold standard

Figure 1 shows the composition of translations in the gold standard. Each source sentence is associated with a list of translations from two sources, i.e. MT output and TM matches. This list of translations is ranked from best to worst according TER scores. The figure shows that over 80% of the translations are from the MT system if we only consider the top-1 translation. As the number of top translations we consider increases, more TM matches can be seen. On the one hand, this does show a large gap in quality between MT output and TM matches; on the other hand, however, it also reveals that we will have to ensure two objectives in ranking: the first is to rank the 80% MT translations higher and the second is to keep the 20% 'good' TM hits in the Top-5. We design our evaluation metrics accordingly.

5.4 Evaluation Metrics

The aim of this research is to provide post-editors with translations that in many cases are easier to edit than the original TM output. As we formulate this as a ranking problem, it is natural to measure the quality of the ranking output by the number of better translations that are ranked high. Sometimes the top TM output is the easiest to edit; in such a case we need to ensure that this translation has a high rank, otherwise the system performance will degrade.

Based on this observation, we introduce the idea of *relevant* translations, and our evaluation metrics: PREC@k and HIT@k.

Relevant Translations. We borrow the idea of *relevence* from the IR community to define the idea of translations worth ranking high. For a source sentence s which has a top TM hit t, we define an MT/TM output m as relevant, if $TER(m) \leq TER(t)$. According to the definition, relevant translations should need no more post-edits than the original top hit from the TM system. Clearly the top TM hit is always relevant.

PREC@k. We calculate the precision (PREC@k) of the ranking for evaluation. Assuming that there are n relevant translations in the top k list for a source sentence s, we have PREC@k= n/k for s. We test PREC@k, for $k = 1...10$, in order to evaluate the overall quality of the ranking.

HIT@k. We also estimate the probability of having one of the relevant translations in the top k, denoted as HIT@k. For a source sentence s, HIT@k equals to 1 if there is at least one relevant translation in top k, and 0 otherwise. This measures the quality of the best translation in top k, which is the translation the post-editor will find and work on if she reads till the kth place in the list. HIT@k equals to 1.0 at the end of the list.

We report the mean PREC@k and HIT@k for all s with the 0.95 confidence interval.

5.5 Experimental Results

In Table 1 we report PREC@k and HIT@k for $k = 1..10$. The ranking receives 0.8747 PREC@1, which means that most of the top ranked translations have at least the same quality as the top TM output. We notice that precision remains above 0.8 till $k = 5$, leading us to conclude that most of the *relevant* translations are ranked in the top-5 positions in the list.

[1] `http://svmlight.joachims.org/`

Table 1: PREC@k and HIT@k of Ranking

	PREC %	HIT %
k=1	87.47±1.60	87.47±1.60
k=2	85.42±1.07	93.36±0.53
k=3	84.13±0.94	95.74±0.61
k=4	82.79±0.57	97.08±0.26
k=5	81.34±0.51	98.04±0.23
k=6	79.26±0.59	99.41±0.25
k=7	74.99±0.53	99.66±0.29
k=8	70.87±0.59	99.84±0.10
k=9	67.23±0.48	99.94±0.08
k=10	64.00±0.46	100.0±0.00

Using the HIT@k scores we can further confirm this argument. The HIT@k score grows steadily from 0.8747 to 0.9941 for $k = 1...6$, so most often there will be at least one *relevant* translation in top-6 for the post-editor to work with. After that room for improvement becomes very small.

In sum, both of the PREC@k scores and the HIT@k scores show that the ranking model effectively integrates the two translation sources (MT and TM) into one merged k-best list, and ranks the *relevant* translations higher.

Table 2: PREC@k - MT and TM Systems

	MT %	TM %
k=1	85.87±1.32	100.0±0.00
k=2	82.52±1.60	73.58±1.04
k=3	80.05±1.11	62.45±1.14
k=4	77.92±0.95	56.11±1.11
k=5	76.22±0.87	51.78±0.78

To measure whether the ranking model is effective compared to pure MT or TM outputs, we report the PREC@k of those outputs in Table 2. The k-best output used in this table is ranked by the MT or TM system, without being ranked by our model. We see the ranked outputs consistently outperform the MT outputs for all $k = 1...5$ w.r.t. precision at a significant level, indicating that our system preserves some high quality hits from the TM.

The TM outputs alone are generally of much lower quality than the MT and Ranked outputs, as is shown by the precision scores for $k = 2...5$. But TM translations obtain 1.0 PREC@1 according to the definition of the PREC calculation. Note that it does not mean that those outputs will need less post-editing (cf. Section 6.1), but rather indicates that each one of these outputs meet the lowest acceptable criterion to be *relevant*.

6 Analysis of Post-Editing Effort

A natural question follows the PREC and HIT numbers: after reading the ranked k-best list, will the post-editors edit less than they would have to if they did not have access to the list? This question would be best answered by human post-editors in a large-scale experimental setting. As we have not yet conducted a manual post-editing experiment, we try to measure the post-editing effort implied by our model with the edit statistics captured by the TER metric, sorted into four types: *Insertion*, *Substitution*, *Deletion* and *Shift*. We report the average number of edits incurred along with the 0.95 confidence interval.

6.1 Top-1 Edit Statistics

We report the results on the 1-best output of TM, MT and our ranking system in Table 3.

In the single best results, it is easy to see that the 1-best output from the MT system requires the least post-editing effort. This is not surprising given the distribution of the gold standard in Section 5.3, where most MT outputs are of better quality than the TM hits.

Moreover, since TM translations are generally of much lower quality as is indicated by the numbers in Table 3 (e.g. 2x as many substitutions and 3x as many deletions compared to MT), unjustly including very few of them in the ranking output will increase loss in the edit statistics. This explains why the ranking model has better ranking precision in Tables 1 and 2, but seems to incur more edit efforts. However, in practice post-editors can neglect an obvious 'bad' translation very quickly.

6.2 Top-k Edit Statistics

We report edit statistics of the Top-3 and Top-5 outputs in Tables 4 and 5, respectively. For each system we report two sets of statistics: the Best-statistics calculated on the best output (according

Table 3: Edit Statistics on Ranked MT and TM Outputs - Single Best

	Insertion	Substitution	Deletion	Shift
TM-Top1	0.7554 ± 0.0376	4.2461 ± 0.0960	2.9173 ± 0.1027	1.1275 ± 0.0509
MT-Top1	0.9959 ± 0.0385	2.2793 ± 0.0628	0.8940 ± 0.0353	1.2821 ± 0.0575
Rank-Top1	1.0674 ± 0.0414	2.6990 ± 0.0699	1.1246 ± 0.0412	1.2800 ± 0.0570

to TER score) in the list, and the Mean- statistics calculated on the whole Top-k list.

The Mean- numbers allow us to have a general overview of the ranking quality, but it is strongly influenced by the poor TM hits that can easily be neglected in practice. To control the impact of those TM hits, we rely on the Best- numbers to estimate the edits performed on the translations that are more likely to be used by post-editors.

In Table 4, the ranking output's edit statistics is closer to the MT output than the Top-1 case in Table 3. Table 5 continues this tendency, in which the Best-in-Top5 Ranking output requires marginally less *Substitution* and *Deletion* operations and significantly less *Insertion* and *Shift* operations (starred) than its MT counterpart. This shows that when more of the list is explored, the advantage of the ranking model – utilizing multiple translation sources – begins to compensate for the possible large number of edits required by poor TM hits and finally leads to reduced post-editing effort.

There are several explanations to why the relative performance of the ranking model improves when k increases, as compared to other models. The most obvious explanation is that a single poor translation is less likely to hurt edit statistics on a k-best list with large k, if most of the translations in the k-best list are of good quality. We see from Tables 1 and 2 that the ranking output is of better quality than the MT and TM outputs w.r.t. precision. For a larger k, the small number of incorrectly ranked translations are less likely to be chosen as the Best- translation and hold back the Best- numbers.

A further reason is related to our ranking model which optimizes on Kendall's τ score. Accordingly the output might not be optimal when we evaluate the Top-1 output, but will behave better when we evaluate on the list. This is also in accordance with our aim, which is to enrich the TM with MT outputs and help the post-editor, instead of choosing the translation for the post-editor.

6.3 Comparing the MT, TM and Ranking Outputs

One of the interesting findings from Tables 3 and 4 is that according to the TER edit statistics, the MT outputs generally need a smaller number of edits than the TM and Ranking outputs. This certainly confirms the necessity to integrate MT into today's TM systems.

However, this fact should not lead to the conclusion that TMs should be replaced by MT completely. First of all, all of our experiments exclude exact TM matches, as those translations will simply be reused and not translated. While this is a realistic setting in the translation industry, it removes all sentences for which the TM works best from our evaluations.

Furthermore, Table 5 shows that the Best-in-Top5 Ranking output performs better than the MT outputs, hence there are TM outputs that lead to smaller number of edits. As k increases, the ranking model is able to better utilize these outputs.

Finally, in this task we concentrate on ranking useful translations higher, but we are not interested in how useless translations are ranked. Ranking SVM optimizes on the ranking of the whole list, which is slightly different from what we actually require. One option is to use other optimization techniques that can make use of this property to get better Top-k edit statistics for a smaller k. Another option is obviously to perform regression directly on the number of edits instead of modeling on the ranking. We plan to explore these ideas in future work.

7 Conclusions and Future Work

In this paper we present a novel ranking-based model to integrate SMT into a TM system, in order to facilitate the work of post-editors. In such

Table 4: Edit Statistics on Ranked MT and TM Outputs - Top 3

	Insertion	Substitution	Deletion	Shift
TM-Best-in-Top3	0.4241 ± 0.0250	3.7395 ± 0.0887	2.9561 ± 0.0966	0.9738 ± 0.0505
TM-Mean-Top3	0.6718 ± 0.0200	5.1428 ± 0.0559	3.6192 ± 0.0649	1.3233 ± 0.0310
MT-Best–in-Top3	0.7696 ± 0.0351	1.9210 ± 0.0610	0.7706 ± 0.0332	1.0842 ± 0.0545
MT-Mean-Top3	1.1296 ± 0.0229	2.4405 ± 0.0368	0.9341 ± 0.0209	1.3797 ± 0.0344
Rank-Best-in-Top3	0.8170 ± 0.0355	2.0744 ± 0.0608	0.8410 ± 0.0338	1.0399 ± 0.0529
Rank-Mean-Top3	1.0942 ± 0.0234	2.7437 ± 0.0392	1.0786 ± 0.0231	1.3309 ± 0.0334

Table 5: Edit Statistics on Ranked MT and TM Outputs

	Insertion	Substitution	Deletion	Shift
TM-Best-in-Top5	0.4239 ± 0.0250	3.7319 ± 0.0885	2.9552 ± 0.0967	0.9673 ± 0.0504
TM-Mean-Top5	0.6143 ± 0.0147	5.5092 ± 0.0473	3.9451 ± 0.0521	1.3737 ± 0.0240
MT-Best-in-Top5	0.7690 ± 0.0351	1.9163 ± 0.0610	0.7685 ± 0.0332	1.0811 ± 0.0544
MT-Mean-Top5	1.1912 ± 0.0182	2.5326 ± 0.0291	0.9487 ± 0.0165	1.4305 ± 0.0272
Rank-Best-in-Top5	0.7246 ± 0.0338*	1.8887 ± 0.0598	0.7562 ± 0.0327	0.9705 ± 0.0515*
Rank-Mean-Top5	1.1173 ± 0.0181	2.8777 ± 0.0312	1.1585 ± 0.0200	1.3675 ± 0.0260

a model, the user of the TM will be presented with an augmented k-best list, consisting of translations from both the TM and the MT systems, and ranked according to ascending prospective post-editing effort.

From the post-editors' point of view, the TM remains intact. And unlike in the binary translation recommendation, where only one translation recommendation is provided, the ranking model offers k-best post-editing candidates, enabling the user to use more resources when translating. As we do not actually throw away any translation produced from the TM, the assets represented by the TM are preserved and the related estimation of the upper bound cost is still valid.

We extract system independent features from the MT and TM outputs and use Ranking SVMs to train the ranking model, which outperforms both the TM's and MT's k-best list w.r.t. precision at k, for all ks.

We also analyze the edit statistics of the integrated k-best output using the TER edit statistics. Our ranking model results in slightly increased number of edits compared to the MT output (apparently held back by a small number of poor TM outputs that are ranked high) for a smaller k, but requires less edits than both the MT and the TM output for a larger k.

This work can be extended in a number of ways. Most importantly, We plan to conduct a user study to validate the effectiveness of the method and to gather HTER scores to train a better ranking model. Furthermore, we will try to experiment with learning models that can further reduce the number of edit operations on the top ranked translations. We also plan to improve the adaptability of this method and apply it beyond a specific domain and language pair.

Acknowledgements

This research is supported by the Science Foundation Ireland (Grant 07/CE/I1142) as part of the Centre for Next Generation Localisation (www.cngl.ie) at Dublin City University. We thank Symantec for providing the TM database and the anonymous reviewers for their insightful comments.

References

Cortes, Corinna and Vladimir Vapnik. 1995. Support-vector networks. *Machine learning*, 20(3):273–297.

Joachims, Thorsten. 2002. Optimizing search engines using clickthrough data. In *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 133–142, New York, NY, USA.

Koehn, Philipp., Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology (NAACL/HLT-2003)*, pages 48 – 54, Edmonton, Alberta, Canada.

Koehn, Philipp, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions (ACL-2007)*, pages 177–180, Prague, Czech Republic.

Levenshtein, Vladimir Iosifovich. 1966. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10(8):707–710.

Och, Franz Josef and Hermann Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics (ACL-2002)*, pages 295–302, Philadelphia, PA, USA.

Och, Franz Josef. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics (ACL-2003)*, pages 160–167, Morristown, NJ, USA.

Shen, Libin, Anoop Sarkar, and Franz Josef Och. 2004. Discriminative reranking for machine translation. In *HLT-NAACL 2004: Main Proceedings*, pages 177–184, Boston, Massachusetts, USA. Association for Computational Linguistics.

Sikes, Richard. 2007. Fuzzy matching in theory and practice. *Multilingual*, 18(6):39 – 43.

Simard, Michel and Pierre Isabelle. 2009. Phrase-based machine translation in a computer-assisted translation environment. In *Proceedings of the Twelfth Machine Translation Summit (MT Summit XII)*, pages 120 – 127, Ottawa, Ontario, Canada.

Snover, Matthew, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of Association for Machine Translation in the Americas (AMTA-2006)*, pages 223–231, Cambridge, MA, USA.

Specia, Lucia, Nicola Cancedda, Marc Dymetman, Marco Turchi, and Nello Cristianini. 2009a. Estimating the sentence-level quality of machine translation systems. In *Proceedings of the 13th Annual Conference of the European Association for Machine Translation (EAMT-2009)*, pages 28 – 35, Barcelona, Spain.

Specia, Lucia, Craig Saunders, Marco Turchi, Zhuoran Wang, and John Shawe-Taylor. 2009b. Improving the confidence of machine translation quality estimates. In *Proceedings of the Twelfth Machine Translation Summit (MT Summit XII)*, pages 136 – 143, Ottawa, Ontario, Canada.

Stolcke, Andreas. 2002. SRILM-an extensible language modeling toolkit. In *Proceedings of the Seventh International Conference on Spoken Language Processing*, volume 2, pages 901–904, Denver, CO, USA.

Ye, Yang, Ming Zhou, and Chin-Yew Lin. 2007. Sentence level machine translation evaluation as a ranking. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 240–247, Prague, Czech Republic.

Learning Phrase Boundaries for Hierarchical Phrase-based Translation

Zhongjun HE Yao MENG Hao YU
Fujitsu R&D Center CO., LTD.
{hezhongjun, mengyao, yu}@cn.fujitsu.com

Abstract

Hierarchical phrase-based models provide a powerful mechanism to capture non-local phrase reorderings for statistical machine translation (SMT). However, many phrase reorderings are arbitrary because the models are weak on determining phrase boundaries for pattern-matching. This paper presents a novel approach to learn phrase boundaries directly from word-aligned corpus without using any syntactical information. We use phrase boundaries, which indicate the beginning/ending of phrase reordering, as soft constraints for decoding. Experimental results and analysis show that the approach yields significant improvements over the baseline on large-scale Chinese-to-English translation.

1 Introduction

The hierarchial phrase-based (HPB) model (Chiang, 2005) outperformed previous phrase-based models (Koehn et al., 2003; Och and Ney, 2004) by utilizing hierarchical phrases consisting of both words and variables. Thus the HPB model has generalization ability: a translation rule learned from a phrase pair can be used for other phrase pairs with the same pattern, e.g. reordering information of a short span can be applied for a large span during decoding. Therefore, the model captures both short and long distance phrase reorderings.

However, one shortcoming of the HPB model is that it is difficult to determine phrase boundaries for pattern-matching. Therefore, during decoding, a rule may be applied for all possible source phrases with the same pattern. However, incorrect pattern-matching will cause wrong translation.

Consider the following rule that is used to translate the Chinese sentence in Figure 1 into English:

$$X \rightarrow \langle X_L \text{ de } X_R, X_R \text{ in } X_L \rangle \qquad (1)$$

The rule translates the Chinese word "de" into English word "in", and swaps the left sub-phrase covered by X_L and the right sub-phrase covered by X_R on the target side. However, X_L may pattern-match 5 spans on the left side of "de" and X_R may pattern-match 3 spans on the right side. Therefore, the rule produces 15 different derivations. However, 14 of them are incorrect.

The correct derivation S_c is shown in Figure 2, while one of the wrong derivations S_i is shown in Figure 3. We observe that the basic difference between S_c and S_i is the phrase boundary matched by "X_R". In S_c, X_R matches the span $[7, 9]$ and moves it as a whole unit. While in S_i, X_R matches the span $[7, 8]$ and left the last word $[9, 9]$ be translated separately. Similarly, other incorrect derivations are caused by inadequate pattern-matching of X_L and/or X_R.

Previous research showed that phrases should be constrained to some extent for improving translation quality. Most of the existing approaches utilized syntactic information to constrain phrases to respect syntactic boundaries. Chiang (2005) introduced a constituent feature to reward phrases that match a syntactic tree but did not yield significant improvement. Marton and Resnik (2008) revised this method by distinguishing different constituent syntactic types, and defined features for each type to count whether a phrase matches or crosses the syntactic boundary. This led to a substantial improvements. Gimpel and Smith (2008) presented rich contextual features on the source side including constituent syntactical features for phrase-based translation. Cherry (2008) utilized a dependency tree as a soft constraint to detect syntactic cohesion violations for a phrase-based

Coling 2008: Poster Volume, pages 383–390,
Beijing, August 2010

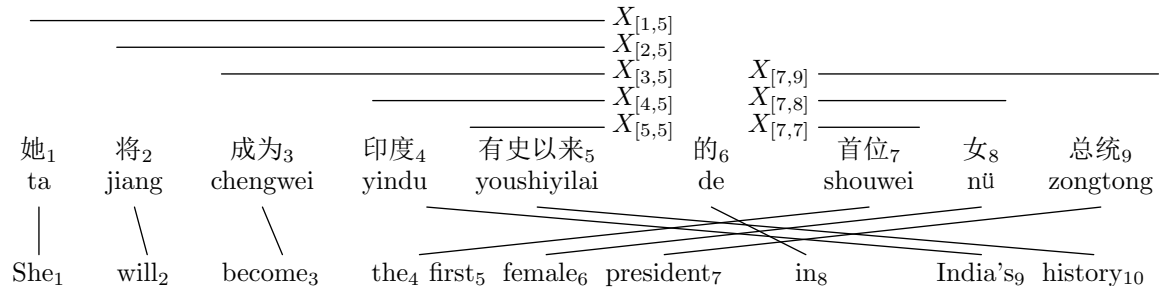


Figure 1: An example of Chinese-English translation. The rule $X \rightarrow \langle X_L \text{ de } X_R,\ X_R \text{ in } X_L \rangle$ pattern-matches 5 and 3 spans on the left and right of the Chinese word "de", respectively.

$$
\begin{aligned}
S_c \ &\Rightarrow\ \langle \text{她 将 成为 } X,\ \text{She will become } X \rangle \\
&\Rightarrow\ \langle \text{她 将 成为 } X_{[4,5]} \text{ 的 } X_{[7,9]},\ \text{She will become } X_{[7,9]} \text{ in } X_{[4,5]} \rangle \\
&\Rightarrow\ \langle \text{她 将 成为 } \| \text{ 印度 有史以来 } \| \text{ 的 } \| \text{ 首位 女 总统,} \\
&\qquad \text{She will become the first female president in India's history} \rangle
\end{aligned}
$$

Figure 2: The correct derivation with adequate pattern-matching of X_R.

$$
\begin{aligned}
S_i \ &\Rightarrow\ \langle \text{她 将 成为 } X \text{ 总统,}\ \text{She will become } X \text{ president} \rangle \\
&\Rightarrow\ \langle \text{她 将 成为 } X_{[4,5]} \text{ 的 } X_{[7,8]} \text{ 总统,}\ \text{She will become } X_{[7,8]} \text{ in } X_{[4,5]} \text{ president} \rangle \\
&\Rightarrow\ \langle \text{她 将 成为 } \| \text{ 印度 有史以来 } \| \text{ 的 } \| \text{ 首位 女 } \| \text{ 总统,} \\
&\qquad \text{She will become the first female in India's history president} \rangle
\end{aligned}
$$

Figure 3: A wrong derivation with inadequate pattern-matching of X_R.

system. Xiong et al. (2009) presented a syntax-driven bracketing model to predict whether two phrases are translated together or not, using syntactic features learned from training corpus. Although these approaches differ from each other, the main basic idea is the utilization of syntactic information.

In this paper, we present a novel approach to learn phrase boundaries for hierarchical phrase-based translation. A phrase boundary indicates the beginning or ending of a phrase reordering. Motivated by Ng and Low (2004) that built a classifier to predict word boundaries for word segmentation, we build a classifier to predict phrase boundaries. We classify each source word into one of the 4 boundary tags: "b" indicates the beginning of a phrase, "m" indicates a word appears in the middle of a phrase, "e" indicates the end of a phrase, "s" indicates a single-word phrase.

We use phrase boundaries as soft constraints for decoding. To do this, we incorporate our classifier as a feature into the HPB model and propose an efficient decoding algorithm.

Compared to the previous work, out approach has the following advantages:

- Our approach maintains the strength of the phrase-based models since it does not require any syntactical information. Therefore, phrases do not need to respect syntactic boundaries.
- The training instances are directly learned from a word-aligned bilingual corpus, rather than from manually annotated corpus.

- The decoder outputs phrase segmentation information as a byproduct, in addition to translation result.

We evaluate our approach on large-scale Chinese-to-English translation. Experimental results and analysis show that using phrase boundaries as soft constraints achieves significant improvements over the baseline system.

2 Previous Work

2.1 Learning Word Boundaries

In some languages, such as Chinese, words are not demarcated. Therefore, it is a preliminary task to determine word boundaries for a sentence, which is the so-called word segmentation.

Ng and Low (2004) regarded word segmentation as a classification problem. They labelled each Chinese character with one of 4 possible boundary tags: "b", "m", "e" respectively indicates the begin, the middle and the end of a word, and "s" indicates a single-character word. Their segmenter was built within a maximum entropy framework and trained on manually segmented sentences.

Learning phrase boundaries is analogous to word boundaries. The basic difference is that the unit for learning word boundaries is character while the unit for learning phrase boundaries is word. In this paper, we adopt the boundary tags presented by Ng and Low (2004) and build a classifier to predict phrase boundaries within maximum entropy framework. We train it directly on a word-aligned bilingual corpus, without any manually annotation and syntactical information.

2.2 The Hierarchical Phrase-based Model

We built a hierarchical phrase-based MT system (Chiang, 2007) based on weighted SCFG. The translation knowledge is represented by rewriting rules:

$$\mathrm{X} \rightarrow \langle \alpha, \gamma, \sim \rangle \tag{2}$$

where X is a non-terminal, α and γ are source and target strings, respectively. Both of them contain words and possibly co-indexed non-terminals. $\sim$ describes a one-to-one correspondence between non-terminals in α and γ.

Chiang (2007) used the standard log-linear framework (Och and Ney, 2002) to combine various features:

$$Pr(e|f) \propto \sum_i \lambda_i h_i(\alpha, \gamma) \tag{3}$$

where $h_i(\alpha, \gamma)$ is a feature function and λ_i is the weight of h_i. Analogous to the previous phrase-based model, Chiang defined the following features: translation probabilities $p(\gamma|\alpha)$ and $p(\alpha|\gamma)$, lexical weights $p_w(\gamma|\alpha)$ and $p_w(\alpha|\gamma)$, word penalty, rule penalty, and a target n-gram language model.

In this paper, we integrate a phrase boundary classifier as an additional feature into the log-linear model to provide soft constraint for pattern-matching during decoding. The feature weights are optimized by MERT algorithm (Och, 2003).

3 Learning Phrase Boundaries

We build a phrase boundary classifier (PBC) within a maximum entropy framework. The PBC predicts a boundary tag for each source word, considering contextual features:

$$P_{tag}(t|f_j, F_1^J) = \frac{exp(\sum_i \lambda_i h_i(t, f_j, F_1^J))}{\sum_t exp(\sum_i \lambda_i h_i(t, f_j, F_1^J)} \tag{4}$$

where, $t \in \{b, m, e, s\}$, f_j is the jth word in source sentence F_1^J, h_i is a feature function and λ_i is the weight of h_i.

To build PBC, we first present a method to recognize phrase boundaries and extract training examples from word-aligned bilingual corpus, then we define contextual feature functions.

3.1 Phrase Boundary

During decoding, intuitively, words within a phrase should be translated or moved together. Therefore, a phrase boundary should indicate reordering information. We assign one of the boundary tags (b, m, e, s) to each word in source sentences. Thus the word with tag *b*, *e* or *s* is a phrase boundary. One question is that how to assign boundary tag to a word? In this paper, we recognize the largest source span which has the monotone translation. Then we assign boundary

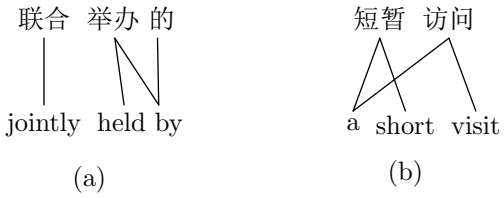


Figure 4: Illustration for monotone span (a) and PM span (b).

tags to each word in the source span, according to their position.

To do this, we first introduce some notations. Given a bilingual sentence (F_1^J, E_1^I) together with word alignment matrix A, we use $L(A_j)$ and $H(A_j)$ to represent the lowest and highest target word position which links to the source word f_j, respectively. Since the word alignment for f_j maybe "one-to-many", all the corresponding target words will appear in the span $[L(A_j), H(A_j)]$.

we define a source span $[j_1, j_2]$ $(1 \leq j_1 \leq j_2 \leq J)$ a *monotone* span, iff:

1. $\forall (j,i) \in A, j_1 \leq j \leq j_2 \leftrightarrow L(A_{j_1}) \leq i \leq H(A_{j_2})$
2. $\forall k_1, k_2 \in [j_1, j_2], k_1 \leq k_2 \rightarrow H(A_{k_1}) \leq L(A_{k_2})$

The first condition indicates that $(F_{j_1}^{j_2}, E_{L(A_{j_1})}^{H(A_{j_2})})$ is a phrase pair as described previously in phrase-based SMT models. While the second condition indicates that the lower target bound linked to a source word cannot be lower than any target word position linked to the previous source word. Therefore, a monotone span does not contain crossed links or internal reorderings.

Considering that word alignments could be very noisy and complex in real-world data, we define *pseudo-monotone* (PM) span by loosening the second condition:

$$\forall k_1, k_2 \in [j_1, j_2], k_1 \leq k_2 \rightarrow L(A_{k_1}) \leq L(A_{k_2}) \quad (5)$$

This condition allows crossed links to some extent by loosening the bound of A_{k_1} from upper to lower. Figure 4 (a) shows an example of monotone span, in which the translation is monotone. While Figure 4 (b) is not a monotone span because there is a cross link between the upper bound of "短暂" and the lower bound of "访问" on the target side. However, it is a PM span according to the definition. Note that in some cases, a source word may not be contained in any phrase pair, therefore we consider a single word span as a PM span, specificly.

An interesting feature of PM span is that if two PM spans are consecutive on both source side and their corresponding target side, the two PM spans can be combined as a larger PM span. Formally,

$$(F_{j_1}^{j}, E_{i_1}^{i}) \bigoplus (F_{j+1}^{j_2}, E_{i+1}^{i_2}) = (F_{j_1}^{j_2}, E_{i_1}^{i_2}) \quad (6)$$

where $[j_1, j]$ and $[j+1, j_2]$ are PM spans, $[i_1, i]$ and $[i+1, i_2]$ are the target spans corresponding to $[j_1, j]$ and $[j+1, j_2]$, respectively. For example, Figure 4 (a) shows a PM phrase pair that consists of two small PM pairs "联合, jointly" and "举办的, held by".

In this paper, we are interested in phrase reordering boundaries for a source sentence. We define *translation span* (TS) the largest possible PM span. A TS may consist of one or more PM spans. According to our definition, cross links may appear within PM spans but do not appear between PM spans within a TS. Therefore, TS is the largest possible span that will be translated as a unit and phrase reorderings may occur between TSs during decoding.

To obtain phrase boundary examples from word-aligned bilingual sentences, we first find all possible TSs and then assign boundary tags to each word. For a TS $[j_1, j_2]$ $(j_1 < j_2)$ that contain more than two words, we assign "b" to the first word f_{j_1} and "e" to the last word f_{j_2}, and "m" to the middle words f_j $(j_1 < j < j_2)$. For a single word span TS $[j, j]$, we assign "s" to the word f_j.

Figure 5 shows an example of labelling source words with boundary tags. The source sentence is segmented into 4 TSs. Using the phrase boundary information to guide decoding, the decoder will produce the correct derivation and translation as shown in Figure 2.

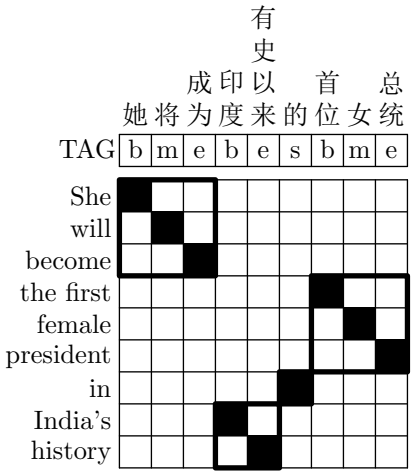


Figure 5: Illustration for labelling the source words with boundary tags. The solid boxes present word alignments. The bordered boxes are TSs.

3.2 Feature Definition

The features we used for the PS model are analogous to (Ng and Low, 2004). For a word W_0, we define the following contextual features with a window of "n":

- The word feature W_n, which denotes the left (right) n words of the current word W_0;
- The part-of-speech (POS) feature P_n, which denotes the POS tag of the word W_n.

For example, the tag of the word "成为 (become)" in Figure 5 is "e", indicating that it is the end of a phrase. If we set the context window $n = 2$, the features of the word "成为 (become)" are:

- W_{-2}=她 W_{-1}=将 W_0=成 为 W_1=印 度 W_2=有史以来
- P_{-2}=r P_{-1}=d P_0=v P_1=ns P_2=l

We collect TSs from bilingual sentences together with the contextual features and used a MaxEnt toolkit (Zhang, 2004) to train a PBC.

	她	将	成为
b	**0.78**	0.10	1.2e-5
m	6.4e-8	**0.75**	5.4e-5
e	2.1e-8	0.11	**0.87**
s	0.22	0.04	0.13

Table 1: The TPM for a source sentence. The highest probability of each word is in bold.

4 Phrase Boundary Constrained Decoding

Give a source sentence, we can assign boundary tags to each word by running the PBC. During decoding, a rule is prohibited to pattern-match across phrase boundaries. By doing this, the PBC is integrated as a hard constraint. However, this method will invalidate a large number of rules and the decoder suffers from a risk that there are not enough rules to cover the source sentence.

Alternatively, inspired by previous approaches, we integrate the phrase boundary classifier as a soft constraint by incorporating it as a feature into the HPB model:

$$h_{pbc}(F_1^J) = log(\prod_{j=1}^{J} P_{tag}(t|f_j, F_1^J)) \quad (7)$$

To perform translation, for each word f_j in a source sentence F_1^J, we first compute all tag probabilities $P_{tag}(t|f_j)$, where $t \in (b, m, e, s)$, $j \in [1, J]$, according to Equation 4. Therefore, we build a $4 \times J$ tag-word probability matrix (TPM). $TPM[i, j]$ indicates the probability of the word f_j labelled with the tag t_i. Table 1 shows the TPM for a source text "她 将 成为".

Then we select rule options from the rule table that can be used for translating the source text. Since each rule option $(\widetilde{f}, \widetilde{e}, a)$ [1] can be regarded as a bilingual sentence with word alignments, thus we find all TS in $\widetilde{f}$ and assign an *initial tag* (IT) for each source word. This procedure is analogous to label phrase boundary tags for a word-aligned bilingual sentence. For example, the following rules are used for translating the Chinese sentence in Table 1:

[1] We keep word alignments of a rule when it is extracted from bilingual sentence.

$$X \rightarrow \langle \text{她}^{b} X_1^{*},\ \text{She } X_1 \rangle \quad (8)$$

$$X_1 \rightarrow \langle \text{将}^{b}\ \text{成为}^{e},\ \text{will become} \rangle \quad (9)$$

Since both the source sides of these two rules are PM spans according to the word alignments, the IT sequences for rule (8) and (9) are "b *"[2] and "b e", respectively. According to Table 1, the initial h_{pbc} score for these two rules can be computed as follows:

$$h_{pbc}^{(7)} = log(P_{tag}(b|\text{她})) = log(TPM[1,1]) \quad (10)$$

$$\begin{aligned} h_{pbc}^{(8)} &= log(P_{tag}(b|\text{将})) + log(P_{tag}(e|\text{成为})) \\ &= log(TPM[1,2]) + log(TPM[3,3]) \end{aligned} \quad (11)$$

Note that to keep the tag sequence valid, e.g. "m" follows "b" rather than "s", the ITs maybe updated during decoding. The tag-updating should be consistent with the definition of TS as described in Section 3.1. Specifically, when the non-terminal symbol X is derived from its covered span $f(X)$, the boundary tags should be updated.

When a tag of word f_j is updated from t_{k_1} to t_{k_2}, the PBC score should also be updated according to TPM:

$$\Delta PBC = log(TPM[k_2, j]) - log(TPM[k_1, j]) \quad (12)$$

The following is a derivation of the source sentence in Table 1:

$$\begin{aligned} S &\Rightarrow \langle \text{她}^{b} X_1^{*},\ \text{She } X_1 \rangle \\ &\Rightarrow \langle \text{她}^{b}\text{将}^{b \rightarrow m}\ \text{成为}^{e},\ \text{She will become} \rangle \end{aligned}$$

When X_1 is derived, the tag of its left boundary word "将" is updated from "b" to "m". The reason is that after derivation, the combined span forms a larger PM span and the left boundary of $f(X_1)$ should be updated.

As a result, the h_{pbc} score is recomputed:

$$h_{pbc}(F_1^3) = h_{pbc}^{(7)} + h_{pbc}^{(8)} + \Delta PBC \quad (13)$$

where,

$$\Delta PBC = log(TPM[2,2]) - log(TPM[1,2]) \quad (14)$$

[2]We use "*" as a tag of the non-terminal symbol "X_1" since it has not been derived.

The decoding algorithm is efficient since the computing of the PBC score is a procedure of table-lookup.

5 Experiments

5.1 Experimental Setup

Our experiments were on Chinese-to-English translation. The training corpus (77M+81M) we used are from LDC [3]. The evaluation metric is BLEU (Papineni et al., 2002), as calculated by mteval-v11b.pl with case-insensitive matching of n-grams, where $n = 4$.

To obtain word alignments, we first ran GIZA++ (Och and Ney, 2002) in both translation directions and then refined it by "grow-diag-final" method (Koehn et al., 2003).

For the language model, we used the SRI Language Modeling Toolkit (Stolcke, 2002) to train two 4-gram models on xinhua portion of GigaWord corpus and the English side of the training corpus.

The NIST MT03 test set is used to tune the feature weights of the log-linear model by MERT (Och, 2003). We tested our system on the NIST MT06 and MT08 test sets.

5.2 Results

The results are shown in Table 2. We tested various settings of the context window. It is observed that the small values of n ($n = 1, 2$) drop the BLEU score, suggesting that perhaps there are not enough contextual information. With more contextual information is used, the BLEU scores are improved over all test sets. When $n = 3$, the most significant improvements are obtained on MT06G and MT08. The improvements over the baseline are statistically significant at $p < 0.01$ by using the significant test method described in (Koehn, 2004). While for MT06N, the optimized context window size is $n = 4$ but the improvement is not statistically significant. In most cases, with n larger than 3, we do not obtain further improvements because of the data sparseness for training

[3]LDC2002E18, LDC2002L27, LDC2002T01, LDC2003E07, LDC2003E14, LDC2004T07, LDC2005E83, LDC2005T06, LDC2005T10, LDC2005T34, LDC2006E24, LDC2006E26, LDC2006E34, LDC2006E86, LDC2006E92, LDC2006E93, LDC2004T08(HK_News, HK_Hansards).

System	MT06G	MT06N	MT08
baseline	14.66	34.42	26.29
+PBC (n=1)	13.78	33.20	24.58
+PBC (n=2)	14.34	34.21	25.87
+PBC (n=3)	**15.19***	34.63	**27.25***
+PBC (n=4)	14.76	**34.73**	26.70

Table 2: Results on the test sets with different context window (n) of the phrase boundary classifier. The largest BLEU score on each test set is in bold. MT06G: MT06 GALE set. MT06N: MT06 NIST set. *: significantly better than the baseline at $p < 0.01$.

the classifier.

6 Discussion

The experimental results show that the phrase boundary constrained method improves the BLEU score over the baseline system. Furthermore, we are interested in how the PBC affects the translation results? We compared the outputs generated by the baseline and "+PBC ($n = 3$)" system and found some interesting translations. For example, the translations of a source sentence of NIST08 are as follows [4]:

- Src: 美$_1^b$ 财长$_2^m$ 抵$_3^m$ 中国$_4^m$ 访问$_5^e$ ‖ 环保$_6^b$ 与$_7^m$ 汇率$_8^e$ ‖ 是$_9^b$ 关切$_{10}^m$ 重点$_{11}^e$
- Ref: US$_1$ Treasury-Secretary$_2$ Arrives-in$_3$ China$_4$ for-a-Visit-with$_5$ Environment$_6$ and$_7$ Exchange-Rate$_8$ as$_9$ Focus$_{10,11}$
- HPB: US$_1$ Treasury$_2$ in-environmental-protection$_6$ and$_7$ visit$_5$ China$_4$ is$_9$ key$_{11}$ to-the-concern-of$_{10}$ the-exchange-rate$_8$
- +PBC: US$_1$ Treasury$_2$ arrived-in$_3$ China$_4$ for-a-visit$_5$ environmental-protection$_6$ and$_7$ exchange-rate$_8$ is$_9$ concerned-about$_{10}$ the-key$_{11}$

In the example, both "环保" and "汇率" in the source sentence are the concern of the "visit". Therefore, the source span $[6, 8]$ indicates a cohesive phrase, which should be translated as a whole unit. However, the baseline translates the spans $[6, 7]$ and $[8, 8]$ separately. It moves $[6, 7]$ before "visit China" and $[8, 8]$ after "concern". This makes an mistake on phrase reordering. We observe that the "+PBC" system produces a better translation. After incorporating the PBC as a soft constraint, the system assigns a boundary tag to each source word and segments the source sentence into three TSs. According to our definition, TSs are encouraged as pseudo-monotone translation unit during decoding. As a result, the "+PBC" system discourages some arbitrary reordering rules and produces more fluent translation.

[4]The co-indexes of the words on the source and target sentence indicate word alignments.

7 Conclusion and Future Work

This paper presented a phrase boundary constrained method for hierarchical phrase-based translation. A phrase boundary indicates begin or end of a phrase reordering. We built a phrase boundary classifier within a maximum entropy framework and learned phrase boundary examples directly from word-aligned bilingual corpus. We proposed an efficient decoding method to integrate the PBC into the decoder as a soft constraint. Experiments and analysis show that the phrase boundary constrained method achieves significant improvements over the baseline system.

The most advantage of the PBC is that it handles both syntactic and non-syntactic phrases. In the future, We would like to try different methods to determine more informative phrase boundaries, e.g. Xiong et al. (2010) proposed a method to learn translation boundaries from a hierarchical tree that decomposed from word alignments using a shift-reduce algorithm. In addition, we will try more features as described in (Chiang et al., 2008; Chiang et al., 2009), e.g. the length of the phrases that covered by non-terminals.

References

Cherry, Colin. 2008. Cohesive phrase-based decoding for statistical machine translation. In *Proceedings of the 46rd Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, page 72 – 80.

Chiang, David, Yuval Marton, and Philip Resnik.

2008. Online large-margin training of syntactic and structural translation features. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, page 224 – 233.

Chiang, David, Wei Wang, and Kevin Knight. 2009. 11,001 new features for statistical machine translation. In *Proceedings of Human Language Technologies: the 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, page 218 – 226.

Chiang, David. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 263–270.

Chiang, David. 2007. Hierarchical phrase-based translation. *Computational Linguistics*, pages 33(2):201–228.

Gimpel, Kevin and Noah A. Smith. 2008. Rich source-side context for statistical machine translation. In *In Proceedings of the ACL-2008 Workshop on Statistical Machine Translation (WMT-2008)*, pages 9–17.

Koehn, Philipp, Franz J. Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of HLT-NAACL 2003*, pages 127–133.

Koehn, Philipp. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 388–395.

Marton, Yuval and Philip Resnik. 2008. Soft syntactic constraints for hierarchical phrased-based translation. In *Proceedings of the 46rd Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1003–1011.

Ng, Hweetou and Jinkiat Low. 2004. Chinese part-of-speech tagging: One-at-a-time or all-at-once? word-based or character-based? In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP 2004)*, pages 277–284.

Och, Franz Josef and Hermann Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 295–302.

Och, Franz Josef and Hermann Ney. 2004. The alignment template approach to statistical machine translation. 30:417–449.

Och, Franz Josef. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 160–167.

Papineni, K., S. Roukos, T. Ward, and W.-J. Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318.

Stolcke, Andreas. 2002. SRILM – An extensible language modeling toolkit. In *Proceedings of the International Conference on Spoken language Processing*, volume 2, pages 901–904.

Xiong, Deyi, Min Zhang, Aiti Aw, and Haizhou Li. 2009. A syntax-driven bracketing model for phrase-based translation. In *ACL-IJCNLP 2009*, page 315 – 323.

Xiong, Deyi, Min Zhang, and Haizhou Li. 2010. Learning translation boundaries for phrase-based decoding. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the ACL*, page 136 – 144.

Zhang, Le. 2004. Maximum entropy modeling toolkit for python and c++. available at http://homepages.inf.ed.ac.uk/s0450736/maxent_toolkit.html.

Learning Summary Content Units with Topic Modeling

Leonhard Hennig **Ernesto William De Luca** **Sahin Albayrak**
Distributed Artificial Intelligence Laboratory (DAI-Lab)
Technische Universität Berlin
{leonhard.hennig,ernesto.deluca,sahin.albayrak}@dai-labor.de

Abstract

In the field of multi-document summarization, the Pyramid method has become an important approach for evaluating machine-generated summaries. The method is based on the manual annotation of text spans with the same meaning in a set of human model summaries. In this paper, we present an unsupervised, probabilistic topic modeling approach for automatically identifying such semantically similar text spans. Our approach reveals some of the structure of model summaries and identifies topics that are good approximations of the Summary Content Units (SCU) used in the Pyramid method. Our results show that the topic model identifies topic-sentence associations that correspond to the contributors of SCUs, suggesting that the topic modeling approach can generate a viable set of candidate SCUs for facilitating the creation of Pyramids.

1 Introduction

In the field of multi-document summarization (MDS), the Pyramid method has become an important approach for evaluating machine-generated summaries (Nenkova and Passonneau, 2004; Passonneau et al., 2005; Nenkova et al., 2007). The method rewards automatic summaries for conveying content that has the same meaning as content represented in a set of human model summaries. This approach allows for variation in the way the content is expressed, which contrasts the Pyramid method with other evaluation methods such as ROUGE that measure word n-gram overlap (Lin and Hovy, 2003).

The Pyramid method groups content with the same meaning into Summary Content Units (SCU). Shared content needs to be identified manually by human inspection of summaries, adding yet another level of human effort (on top of creating model summaries) to the task of summary evaluation. However, Nenkova and Passonneau (2004) as well as Harnly et al. (2005) observe that semantically similar text spans written by different human summarizers are often expressed with a similar choice of words, albeit with differences e.g. in word variants, word order and paraphrasing (Section 2).

In this paper, we present an approach for automatically identifying semantically similar text spans in human model summaries on the basis of such re-occurring word patterns. We utilize a method known as probabilistic topic modeling (Steyvers and Griffiths, 2007). Topic models are claimed to derive semantic information from text in an unsupervised fashion, using only the observed word distributions (Section 3).

- We train a probabilistic topic model based on Latent Dirichlet Allocation (Blei et al., 2003) on the term-sentence matrix of human model summaries used in the Document Understanding Conference (DUC) 2007 Pyramid evaluation[1]. We analyze the resulting model to evaluate whether a topic model captures useful structures of these summaries (Section 4.1).

[1] http://duc.nist.gov

Coling 2008: Poster Volume, pages 391–399,
Beijing, August 2010

- Given the model, we compare the automatically identified topics with SCUs on the basis of their word distributions (Sections 4.2 and 4.3). We discover a clear correspondence between topics and SCUs, which suggests that many automatically identified topics are good approximations of manually annotated SCUs.

- We analyze the distribution of topics over summary sentences in Section 4.4, and compare the topic-sentence associations computed by our model with the SCU-sentence associations given by the Pyramid annotation. Our results suggest that the topic model finds many SCU-like topics, and associates a given topic with the same summary sentences in which a human annotator identifies the corresponding SCU.

Automatically identifying topics that approximate SCUs has clear practical applications: The topics can be used as a candidate set of SCUs for human annotators to speed up the process of SCU creation. Topics can also be identified in machine-generated summaries using standard statistical inference techniques (Asuncion et al., 2009).

2 Summary Content Units

In this section, we briefly introduce the Pyramid method and the properties of Summary Content Units that we intend to exploit in our approach.

A Pyramid is a model predicting the distribution of information content in summaries, as reflected in the summaries humans write (Passonneau et al., 2005; Nenkova et al., 2007). Similar information content is identified by inspection of similar sentences, and parts of these, in different human model summaries. Typically, the text spans which express the same semantic content are not longer than a clause. An SCU consists of a collection of text spans with the same meaning (contributors) and a defining label specified by the annotator.

Each SCU is weighted by the number of human model summaries it occurs in (i.e. the number of contributors). The Pyramid metric assumes that an SCU with a high number of contributors is more informative than an SCU with few contributors. An optimal summary, in terms of content selection, is obtained by maximizing the sum of SCU weights, given a maximum number of SCUs that can be included for a predefined summary length (Nenkova and Passonneau, 2004).

Two example SCUs are given in Table 1. SCU 18 has a weight of 3, since three model summaries contribute to it, SCU 21 has a weight of 2. SCU 18 aggregates contributors which share some key phrases such as "Air National Guard" and "search", but otherwise exhibit a quite heterogeneous word usage. Contributor 3 gives details on the aircraft type, and specifies a time when the first sea vessel was launched to search for the missing plane. Only contributor 1 gives information about the location of the search. In SCU 21, the first contributor contains additional information about communication with the Kennedy family, which is not expressed in the SCU label and therefore not part of the meaning of the SCU. Both contributors contain key terms such as "officials", "search" and "recovery", but vary in word order and verb usage. Passonneau et al. (2005) discuss this observation, and argue that SCUs emerge from the judgment of annotators, and are thus independent of what words are used, or how many.

However, an analysis of typical SCUs shows that contributors written by different human summarizers are often expressed with a similar choice of words or even phrases. Contributors vary in using different forms of the same words (inflectional or derivational variants), different word order, syntactic structure, and paraphrases (Harnly et al., 2005; Nenkova et al., 2007).

3 Probabilistic Topic Models

Our approach for discovering semantically similar text spans makes use of a statistical method known as topic modeling. Probabilistic topic models can derive semantic information from text automatically, on the basis of the observed word patterns (Hofmann, 1999; Blei et al., 2003; Steyvers and Griffiths, 2007). The main assumption of these models is that a latent set of variables – the topics – can be utilized to explain the observed patterns in the data. Documents are represented as mixtures of topics, and each topic is a distribution

SCU 18	The US Coast Guard with help from the Air National Guard then began a massive search-and-rescue mission, searching waters along the presumed flight path
Contributor 1:	The US Coast Guard with help from the Air National Guard then began a massive search-and-rescue mission, searching waters along the presumed flight path
Contributor 2:	A multi-agency search and rescue mission began at 3:28 a.m., with the Coast Guard and Air National Guard participating
Contributor 3:	The first search vessel was launched at about 4:30am. An Air National Guard C-130 and many Civil Air Patrol aircraft joined the search
SCU 21	Federal officials shifted the mission to search and recovery
Contributor 1:	Federal officials shifted the mission to search and recovery and communicated the Kennedy and Bessette families
Contributor 2:	federal officials ended the search for survivors and began a search-and-recovery mission

Table 1: Example SCUs from topic D0742 of DUC 2007.

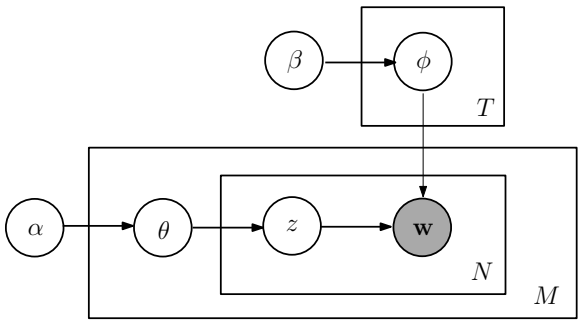


Figure 1: Graphical model representation of LDA for N words, T topics and a corpus of M documents.

over words. For example, a news article describing a meeting of the International Monetary Fund may in equal parts discuss economic and political issues. Topic models discover in a completely unsupervised fashion meaningful topics as well as intra- and inter-document statistical structure using no information except the distribution of the words themselves (Griffiths and Steyvers, 2004).

For our analysis, we use the Latent Dirichlet Allocation (LDA) model introduced by Blei et al. (2003). In this model, each document is generated by first choosing a distribution over topics $\theta^{(d)}$, parametrized by a conjugate Dirichlet prior α. Subsequently, each word of this document is generated by drawing a topic z_k from $\theta^{(d)}$, and then drawing a word w_i from topic z_k's distribution over words $\phi^{(k)}$. We follow Griffiths et al. (2004) and place a conjugate Dirichlet prior β over $\phi^{(k)}$ as well. Figure 1 shows the graphical model representation of LDA.

For T topics, the matrix $\mathbf{\Phi}$ specifies the probability $p(w|z)$ of words given topics, and $\mathbf{\Theta}$ specifies the probability $p(z|d)$ of topics given documents. $p(w|z)$ indicates which words are important in a topic, and $p(z|d)$ tells us which topics are dominant in a document. We employ Gibbs sampling (Griffiths and Steyvers, 2004) to estimate the posterior distribution over z (the assignment of word tokens to topics), given the observed words $\mathbf{w}$ of the document set. From this estimate we can approximate the distributions for the matrices $\mathbf{\Phi}$ and $\mathbf{\Theta}$.

4 Experiments

Can a topic model reveal some of the structure of human model summaries and learn topics that are approximations of manually annotated SCUs? To answer these questions, we train a topic model on the human model summaries of each of the 23 document clusters of the DUC 2007 dataset that were used in Pyramid evaluation[2]. There are 4 human model summaries available for each document cluster. On average, the summary sets contain 52.4 sentences, with a vocabulary of 260.5 terms, which occur a total of 549.7 times. The Pyramids of these summary sets consist of 68.8 SCUs on average. The number of SCUs per SCU

[2]http://www-nlpir.nist.gov/projects/duc/data.html

weight follows a Zipfian distribution, i.e. there are typically very few SCUs of weight 4, and very many SCUs of weight 1 (see also Passonneau et al. (2005)).

4.1 Topic model training

Since we are interested in modeling topics for sentences, we treat each sentence as a document[3]. We construct a matrix $\mathbf{A}$ of term-sentence co-occurrence observations for each set of human model summaries S. Each entry $\mathbf{A}_{ij}$ corresponds to the frequency of word i in sentence j, and j ranges over the union of the sentences contained in S. We preprocess terms using stemming and removing a standard list of stop words with the NLTK toolkit[4].

We run the Gibbs sampling algorithm on $\mathbf{A}$, setting the parameter T, the number of latent topics to learn, equal to the number of SCUs contained in the Pyramid of S. We use this particular value for T since we want to learn a topic model with a structure that reflects the SCUs and the distribution of SCUs of the corresponding Pyramid. For an unannotated set of summaries, determining an optimal value for T is a Bayesian model selection problem (Kass and Raftery, 1995).

The topic distribution for each sentence should be peaked toward a single or only very few topics. To ensure that the topic-specific word distributions $p(w|z)$ as well as the sentence-specific topic distributions $p(z|d)$ behave as intended, we set the Dirichlet priors $\alpha = 0.01$ and $\beta = 0.01$. This enforces a bias toward sparsity, resulting in distributions that are more peaked (Steyvers and Griffiths, 2007). A low value of β also favors more fine-grained topics (Griffiths and Steyvers, 2004). We run the Gibbs sampler for 2000 iterations, and collect a single sample from the resulting posterior distribution over topic assignments for words. From this sample we compute the conditional distributions $p(w|z)$ and $p(z|d)$.

During our experiments, we observed that the Gibbs Sampler did not always use all the topics available. Instead, some topics had a uniform distribution over words, i.e. no words were assigned to these topics during the sampling process. We assume this is due to the relatively low prior $\alpha = 0.01$ we use in our experiments. We explore the consequences of varying the LDA priors and T in Section 4.4.

This observation indicates that the topic model cannot learn as many distinct topics from a given set of summaries as there are SCUs in the Pyramid of these summaries. On average, 24.4% ($\sigma = 17.4$) of the sampled topics had a uniform word distribution, but the fraction of such topics varied. For some summary sets, it was very low (D0701, D0706 with 0%), whereas for others it was very high (D0704, D0728 with 52%). Both of the latter summary sets contain many SCUs with very similar labels and often only a single contributor, e.g. about 'Amnesty International':

- AI criticism frequently involves genocide
- AI criticism frequently involves intimidation
- AI criticism frequently involves police violence

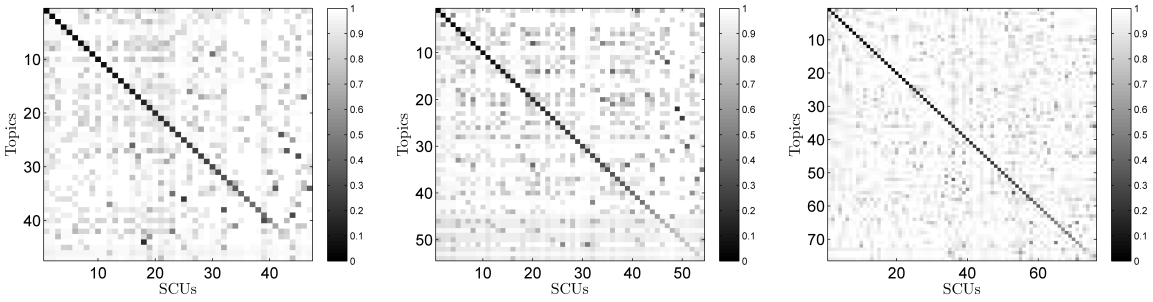
These SCUs are derived from summary sentences that contain enumerations: "AI criticism frequently involves political prisoners, torture, intimidation, police violence, the death penalty, no alternative service for conscientious objectors, and interference with the judiciary." A topic model is based on word-document co-occurrence data, and cannot distinguish between the different grammatical objects in this case. Instead, it treats these phrases as semantically similar since they occur in the same sentence.

4.2 SCU word distributions and SCU-sentence associations

In order to evaluate the quality of the LDA topics, we compare their word distributions to the word distributions of SCUs. This allows us to analyze if the LDA topics capture similar word patterns as SCUs. We approximate the distribution over words $p(w|s_l)$ for each SCU s_l as the relative frequency of word w_i in the bag-of-words constructed from the texts of s_l's label and contributors. We denote the resulting matrix of for a set of SCUs as $\widehat{\mathbf{\Phi}}$.

[3] We will use the words document and sentence interchangeably from here on.

[4] http://www.nltk.org



(a) DUC Topic D0706 (b) DUC Topic D0742 (c) DUC Topic D0743

Figure 2: Pairwise Jensen-Shannon divergence of word distributions of LDA topics and Summary Content Units (SCUs), for 3 DUC 2007 Pyramids. Topic-SCU matches are ordered by increasing divergence along the diagonal, using a simple greedy algorithm. The examples suggest that many of the automatically identified LDA topics correspond to manually annotated SCUs.

Topic 17	SCU 31	Topic 5	SCU 32	Topic 9	SCU 25	Topic 8	SCU 36
pilot	pilot	analysi	analysi	bodi	bodi	kennedi	kennedi
kennedi	condit	control	control	diver	diver	edward	edward
condit	conduc	corkscrew	corkscrew	entomb	entomb	recoveri	recoveri
conduc	dark	descent	descent	floor	floor	son	son
dark	disorient	fall	fall	found	found	wit	wit

Table 2: Top terms of best matching LDA topics and SCUs for summary set D0742

In addition, we can compare the topic-sentence associations computed by the model to the SCU-sentence associations given by the Pyramid annotation. If the probability of a given topic is high in those sentences which contribute to a particular SCU, this would suggest that the topic model can automatically learn topics which not only have a word distribution similar to a specific SCU, but also a similar distribution over contributing sentences.

SCU contributors are typically annotated as a set of contiguous sequences of words within a single sentence. In the DUC 2007 data, there are only a few cases where a contributor spans more than one sentence. The DUCView annotation tool[5] stores the start and end character position of the phrases marked as contributors of an SCU. We can utilize this information to define which sentences an SCU is associated with. We store the associations in a matrix $\widehat{\mathbf{\Theta}}$, where $\widehat{\mathbf{\Theta}}_{ij} = 1$ if SCU i is associated with sentence j. Sentences may contain multiple SCUs, and SCUs are associated with as many sentences as their number of contributors.

[5]http://www1.cs.columbia.edu/~becky/DUC2006/2006-pyramid-guidelines.html

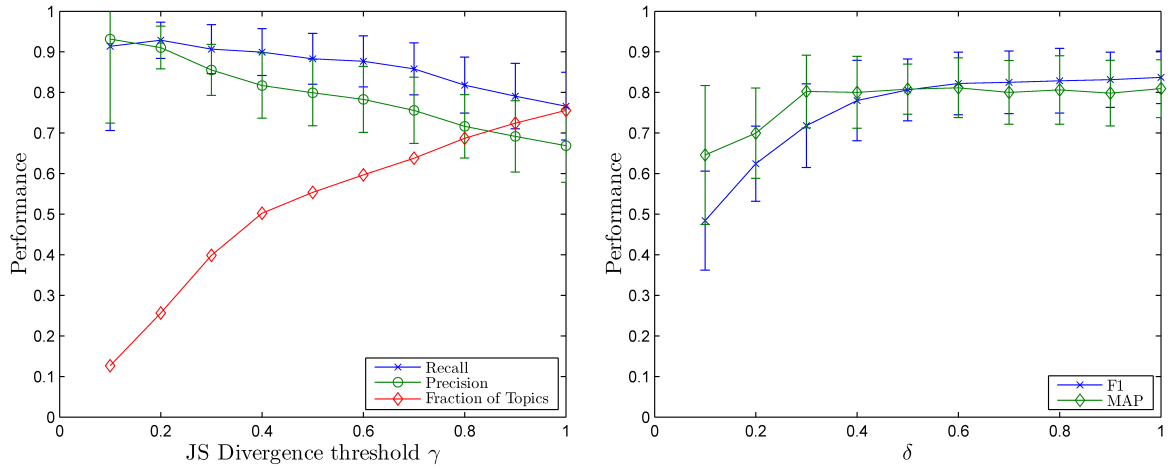
4.3 Matching SCUs and LDA topics

Before we can compare the topic-sentence associations computed by the LDA topic model with the SCU-sentence associations, we need to match SCUs to LDA topics. We consider a topic to be similar to an SCU if their word distributions are similar. We discard all LDA topics with a uniform word distribution (see Section 4.1) before the matching step.

We then compute the pair-wise Jensen-Shannon (JS) divergence between columns j of $\mathbf{\Phi}$ and k of $\widehat{\mathbf{\Phi}}$:

$$JS(\mathbf{\Phi}_j, \widehat{\mathbf{\Phi}}_k) = \left[\frac{1}{2} D_{KL}(\mathbf{\Phi}_j || M) + \frac{1}{2} D_{KL}(\widehat{\mathbf{\Phi}}_k || M)\right], \quad (1)$$

where $M = 1/2(\mathbf{\Phi}_j + \widehat{\mathbf{\Phi}}_k)$. SCUs from $\widehat{\mathbf{\Phi}}$ are matched to topics of $\mathbf{\Phi}$ on the basis of this dissimilarity using a simple greedy approach, i.e. by iteratively selecting the current most similar SCU-



(a) Precision, recall and fraction of Topic-SCU matches for different settings of γ

(b) F1 and MAP for different values of T as a fraction δ of the number of SCUs

Figure 3: (a) Precision, Recall and the fraction of LDA topics matched to SCUs for different settings of parameter γ, averaged over all summary sets with Pyramid annotations from DUC 2007. Error bars show the standard deviation. Only topic-SCU matches with $JS(\mathbf{\Phi}_j, \widehat{\mathbf{\Phi}}_k) \leq \gamma$ are considered when computing precision and recall. Both are very high, suggesting that the model identifies topics that are very similar to SCUs. (b) F_1 measure and Mean Average Precision (MAP) for different settings of the number of latent topics T as a fraction of the number of SCUs in the corresponding Pyramid ($\gamma = 0.5$).

topic pair. We reorder the rows of $\mathbf{\Theta}$ according to the computed matching.

Figure 2 shows some example SCU-topic matches for three different DUC 2007 summary sets. Each cell displays the JS divergence of the word distributions of an LDA topic (rows) compared to an SCU (columns). On the diagonal, the best matches of LDA topics and SCUs are ordered by increasing JS divergence. Multiple points with low JS divergence in a single column indicate that more than one LDA topic was very similar to this SCU. Overall, the graphs show a clear correspondence of LDA topics to the SCUs. The plots suggest that a large percentage of topics have similar distributions over words as the corresponding SCUs. Table 2 shows the most likely terms for some example topic-SCU matches. For each of these matches, the top terms are almost identical.

4.4 Evaluation

To compare the topic distributions $\mathbf{\Theta}$ with the SCU-sentence assignments $\widehat{\mathbf{\Theta}}$, we binarize $\mathbf{\Theta}$ to give $\mathbf{\Theta}'$ by setting all entries $\mathbf{\Theta}'_{ij} = 1$ if $\mathbf{\Theta}_{ij} > \epsilon$, and 0 otherwise. We set $\epsilon = 0.1$ in our experiments[6]. $\mathbf{\Theta}'_{ij}$ is therefore equal to 1 if a topic i has a high probability sentence j. We can now evaluate if a given topic occurs in the same sentences as the corresponding SCU (recall), and if it occurs in no other sentences (precision).

We compute precision and recall for each topic-SCU match with $JS(\mathbf{\Phi}_j, \widehat{\mathbf{\Phi}}_k) \leq \gamma$. Averaged over matches, these measures give us an indication of how well the LDA model approximates the set of SCUs. The parameter γ allows us to tune the performance of the model with respect to the quality and number of topic-SCU matches. Setting γ to a low value will consider only topic-SCU matches with a low JS divergence, which generally results in higher precision and recall. Increasing γ will include more topic-SCU matches, namely those with a larger JS divergence, which will therefore introduce some noise.

Figure 3(a) shows the precision and recall

[6]Since the LDA algorithm learns very peaked distributions, the actual value of this threshold does not have a large impact on the resulting binary matrix and subsequent evaluation results. We evaluated a range of settings for ϵ in $[0.001 - 0.5]$, all with similar performance. This observation is confirmed by the threshold-less Mean Average Precision results in Figure 3(b).

curves for different values of the parameter γ, averaged over all summary sets. The plots show that both the precision and recall of the discovered topic-sentence associations are quite high, suggesting that the model automatically identifies topics which are very similar to manually annotated SCUs. With increasing γ, precision and recall scores decrease: The word distributions of the topic-SCU pairs are increasingly dissimilar, and hence the sentences associated with a topic do not necessarily overlap anymore with the sentences of the paired SCU. The figure also shows the fraction of topic matches that are considered in the evaluation of precision and recall. There is a clear trade off between performance and the number of matches retrieved. However, many of the topic-SCU matches ($\approx 50\%$) have a JS divergence ≤ 0.4, suggesting that the word distributions of many LDA topics are very similar to SCU word distributions.

Since we observed that the Gibbs sampling does not always utilize the full set of topics, we repeat our experiments to evaluate how the performance of the model changes when varying the LDA priors and T. Figure 3(b) shows F_1 and Mean Average Precision (MAP)[7] results of the topic model for different values of the parameter δ, where $T = \delta * |SCU|$. For example, a value of 0.6 means that for each summary set, T was set to 60% of the number of SCUs in the corresponding Pyramid. We see that the MAP score increases quickly, and reaches a plateau for $\delta \geq 0.3$. The F_1 score increases more slowly, and levels out for $\delta \geq 0.6$. The model's performance is relatively robust with respect to δ. This observation can be helpful when training models for new summary sets without an existing Pyramid, and which therefore consider T as a parameter to be optimized.

When varying the LDA priors, we observe that for $0.01 \leq \alpha \leq 0.05$, F_1 and MAP scores are consistently high, whereas for other settings, performance decreases significantly. Similarly, $\beta \geq 0.05$ results in lower F_1 and MAP scores. The fraction of uniform topics decreases with higher α, e.g. for $\alpha = 0.1$ it is close to zero. In contrast, higher settings of β increase the fraction of uniform topics.[8]

Finally, Figure 4 shows separate precision and recall curves for SCUs of different weights, and for different settings of parameter γ. Results are again averaged over all summary sets. In 4(a), we see that the recall of topic-sentence associations is very similar for all SCUs, with SCUs of higher weight exhibiting a slightly better recall. However, as Figure 4(b) shows, the average precision of SCUs with lower weight is much higher. Intuitively, this is expectable as SCUs of higher weight tend to have a larger vocabulary due to the higher number of contributors. This results in a larger word overlap with non-relevant sentences. The fraction of topic-SCU matches retrieved for SCUs of different weight is similar for all types of SCUs (not shown here).

5 Related Work

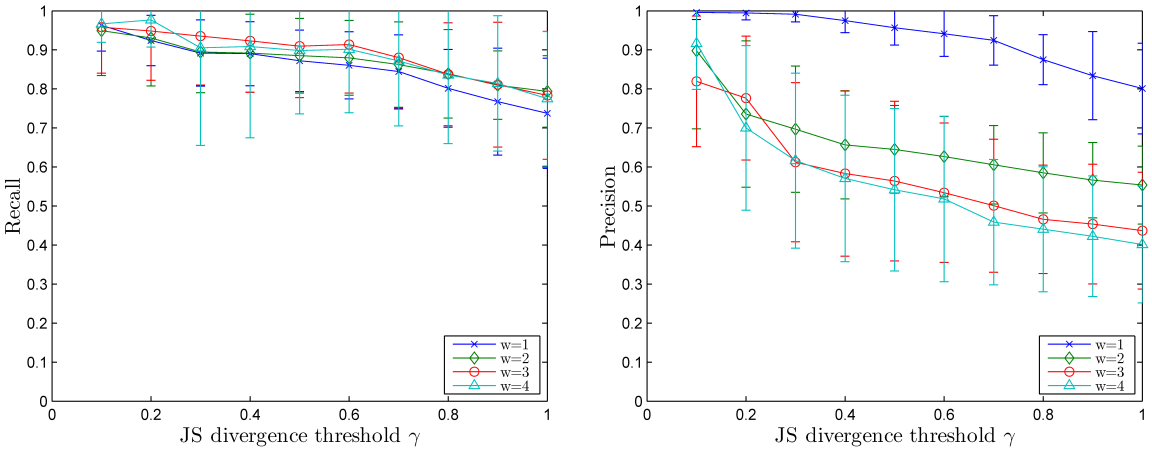
The Pyramid approach was introduced by Nenkova and Passonneau (2004) as a method for evaluating machine-generated summaries based on a set of human model summaries. The authors address a number of shortcomings of manual and automatic summary evaluation methods such as ROUGE (Lin and Hovy, 2003), and argue that the Pyramid method is reliable, diagnostic and predictive.

Passonneau and et al. (2005) give an account of the results of applying the Pyramid method during the DUC 2005 summarization evaluation, and discuss the annotation process. In subsequent work, Nenkova et al. (2007) describe in more detail the incorporation of human variation in the Pyramid method, the reliability of content annotation, and the correlation of Pyramid scores with other evaluation measures.

Harnly et al. (2005) present an approach for automatically scoring a machine summary given an existing Pyramid. Their method searches for an optimal set of candidate contributors created automatically from the machine summary and matches candidates to SCUs using a clustering approach.

[7] MAP is a rank-based measure, which avoids the need for introducing a threshold to binarize Θ (Baeza-Yates and Ribeiro-Neto, 1999). For each topic, we create a ranked list of sentences according to the transposed matrix $\mathbf{\Theta}^T$. This gives high ranks to sentences for which a particular topic has a high probability.

[8] Results are not shown due to space constraints.



(a) Recall of Topic-SCU matches for SCUs by weight

(b) Precision of Topic-SCU matches for SCUs by weight

Figure 4: (a) Recall of topic-SCU matches for SCUs of different weights, and settings of parameter γ, averaged over all summary sets. Recall is similar for SCUs of all weights. (b) Precision of the same topic-SCU matches. SCUs with a lower weight have a higher average precision. (Error bars show the standard deviation.)

The method assumes the existence of a Pyramid, whereas our approach aims to discover candidate SCUs from a set of human model summaries in an unsupervised fashion.

Recently, Louis and Nenkova (2009) presented an approach for fully automatic, model-free evaluation of machine-generated summaries. The method assumes that the distribution of words in the input and an informative summary should be similar. We think that it could be an interesting idea to combine the proposed method with our approach, in an attempt to exploit both the model-free evaluation and the shallow semantics of latent topics.

Probabilistic topic models have been successfully applied to a variety of tasks (Hofmann, 1999; Blei et al., 2003; Griffiths and Steyvers, 2004; Hall et al., 2008). In text summarization, most topic modeling approaches utilize a term-sentence co-occurrence matrix to discover topics in the set of input documents. Each sentence is typically assigned to a single topic, and a topic is a cluster of multiple sentences (Wang et al., 2009; Tang et al., 2009; Hennig, 2009).

6 Conclusions and future work

We presented a probabilistic topic modeling approach that reveals some of the structure of human model summaries. The topic model is trained on the term-sentence matrix of a set of human summaries, and discovers semantic topics in a completely unsupervised fashion. Many of the topics identified by our model for a given set of summaries show a similar distribution over words as the manually annotated Summary Content Units of the summaries' Pyramid.

We utilized the word distributions of SCUs and topics to match topics to similar SCUs, and showed that the topics identified by the model often occur in the same sentences as the contributors of the corresponding SCU. Precision and recall of these topic-sentence assignments are very high when compared to the SCU-sentence associations, indicating that many of the automatically acquired topics are good approximations of SCUs. Our results suggest that a topic model can be used to learn a candidate set of SCUs to facilitate the process of Pyramid creation.

We note that the topic model that we applied is one of the simplest latent variable models. A more complex model could integrate syntax to relax the bag-of-words assumption (Wallach, 2006), or combine the statistical model with more linguistically-grounded methods to handle linguistic features such as enumerations or negation.

References

Asuncion, Arthur, Max Welling, Padhraic Smyth, and Yee Whye Teh. 2009. On smoothing and inference for topic models. In *UAI '09: Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pages 27–34.

Baeza-Yates, Ricardo A. and Berthier Ribeiro-Neto. 1999. *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

Blei, David M., Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022.

Griffiths, T. L. and M. Steyvers. 2004. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101(Suppl. 1):5228–5235.

Hall, David, Daniel Jurafsky, and Christopher D. Manning. 2008. Studying the history of ideas using topic models. In *EMNLP '08: Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 363–371.

Harnly, A., A. Nenkova, R. Passonneau, and O. Rambow. 2005. Automation of summary evaluation by the Pyramid method. In *Proceedings of the Conference on Recent Advances in Natural Language Processing (RANLP)*.

Hennig, Leonhard. 2009. Topic-based multi-document summarization with probabilistic latent semantic analysis. In *International Conference on Recent Advances in Natural Language Processing (RANLP)*.

Hofmann, Thomas. 1999. Probabilistic latent semantic indexing. In *SIGIR '99: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57.

Kass, R. E. and A. E. Raftery. 1995. Bayes factors. *Journal of the American Statistical Association*, 90:773–795.

Lin, Chin-Yew and Eduard Hovy. 2003. Automatic evaluation of summaries using N-gram co-occurrence statistics. In *NAACL '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 71–78.

Louis, Annie and Ani Nenkova. 2009. Automatically evaluating content selection in summarization without human models. In *EMNLP '09: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 306–314.

Nenkova, Ani and Rebecca Passonneau. 2004. Evaluating Content Selection in Summarization: The Pyramid Method. In Susan Dumais, Daniel Marcu and Salim Roukos, editors, *HLT-NAACL 2004: Main Proceedings*, pages 145–152.

Nenkova, Ani, Rebecca Passonneau, and Kathleen McKeown. 2007. The Pyramid Method: Incorporating human content selection variation in summarization evaluation. *ACM Trans. Speech Lang. Process.*, 4(2):4.

Passonneau, R. J, A. Nenkova, K. McKeown, and S. Sigelman. 2005. Applying the Pyramid method in DUC 2005. In *Proceedings of the Document Understanding Conference (DUC'05)*.

Steyvers, Mark and Tom Griffiths. 2007. Probabilistic topic models. In Landauer, T., S. Dennis McNamara, and W. Kintsch, editors, *Latent Semantic Analysis: A Road to Meaning*. Laurence Erlbaum.

Tang, J., L. Yao, and D. Chen. 2009. Multi-topic based query-oriented summarization. In *Proceedings of the Siam International Conference on Data Mining*.

Wallach, Hanna M. 2006. Topic modeling: beyond bag-of-words. In *ICML '06: Proceedings of the 23rd international conference on Machine learning*, pages 977–984.

Wang, Dingding, Shenghuo Zhu, Tao Li, and Yihong Gong. 2009. Multi-document summarization using sentence-based topic models. In *ACL-IJCNLP '09: Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 297–300.

Learning to Model Domain-Specific Utterance Sequences for Extractive Summarization of Contact Center Dialogues

Ryuichiro Higashinaka†, Yasuhiro Minami‡, Hitoshi Nishikawa†,
Kohji Dohsaka‡, Toyomi Meguro‡, Satoshi Takahashi†, Genichiro Kikui†
† NTT Cyber Space Laboratories, NTT Corporation
‡ NTT Communication Science Laboratories, NTT Corporation
higashinaka.ryuichiro@lab.ntt.co.jp, minami@cslab.kecl.ntt.co.jp
nishikawa.hitoshi@lab.ntt.co.jp, {dohsaka,meguro}@cslab.kecl.ntt.co.jp
{takahashi.satoshi,kikui.genichiro}@lab.ntt.co.jp

Abstract

This paper proposes a novel extractive summarization method for contact center dialogues. We use a particular type of hidden Markov model (HMM) called Class Speaker HMM (CSHMM), which processes operator/caller utterance sequences of multiple domains simultaneously to model domain-specific utterance sequences and common (domain-wide) sequences at the same time. We applied the CSHMM to call summarization of transcripts in six different contact center domains and found that our method significantly outperforms competitive baselines based on the maximum coverage of important words using integer linear programming.

1 Introduction

In modern business, contact centers are becoming more and more important for improving customer satisfaction. Such contact centers typically have quality analysts who mine calls to gain insight into how to improve business productivity (Takeuchi et al., 2007; Subramaniam et al., 2009). To enable them to handle the massive number of calls, automatic summarization has been utilized and shown to successfully reduce costs (Byrd et al., 2008). However, one of the problems in current call summarization is that a domain ontology is required for understanding operator/caller utterances, which makes it difficult to port one summarization system from domain to domain.

This paper describes a novel automatic summarization method for contact center dialogues without the costly process of creating domain ontologies. More specifically, given contact center dialogues categorized into multiple domains, we create a particular type of hidden Markov model (HMM) called **Class Speaker HMM** (**CSHMM**) to model operator/caller utterance sequences. The CSHMM learns to distinguish sequences of individual domains and common sequences in all domains at the same time. This approach makes it possible to accurately distinguish utterances specific to a certain domain and thereby has the potential to generate accurate extractive summaries.

In Section 2, we review recent work on automatic summarization, including its application to contact center dialogues. In Section 3, we describe the CSHMM. In Section 4, we describe our automatic summarization method in detail. In Section 5, we describe the experiment we performed to verify our method and present the results. In Section 6, we summarize and mention future work.

2 Related Work

There is an abundance of research in automatic summarization. It has been successfully applied to single documents (Mani, 2001) as well as to multiple documents (Radev et al., 2004), and various summarization methods, such as the conventional LEAD method, machine-learning based sentence selection (Kupiec et al., 1995; Osborne, 2002), and integer linear programming (ILP) based sentence extraction (Gillick and Favre, 2009), have been proposed. Recent years have seen work on summarizing broadcast news speech (Hori and Furui, 2003), multi-party meetings (Murray et al., 2005), and contact center dialogues (Byrd et al., 2008). However, despite the large amount of previous work, little work has tackled the automatic summarization of multi-domain data.

Coling 2008: Poster Volume, pages 400–408,
Beijing, August 2010

In the past few decades, contact center dialogues have been an active research focus (Gorin et al., 1997; Chu-Carroll and Carpenter, 1999). Initially, the primary aim of such research was to transfer calls from answering agents to operators as quickly as possible in the case of problematic situations. However, real-time processing of calls requires a tremendous engineering effort, especially when customer satisfaction is at stake, which led to recent work on the offline processing of calls, such as call mining (Takeuchi et al., 2007) and call summarization (Byrd et al., 2008).

The work most related to ours is (Byrd et al., 2008), which maps operator/caller utterances to an ontology in the automotive domain by using support vector machines (SVMs) and creates a structured summary by heuristic rules that assign the mapped utterances to appropriate summary sections. Our work shares the same motivation as theirs in that we want to make it easier for quality analysts to analyze the massive number of calls. However, we tackle the problem differently in that we propose a new modeling of utterance sequences for extractive summarization that makes it unnecessary to create heuristics rules by hand and facilitates the porting of a summarization system.

HMMs have been successfully applied to automatic summarization (Barzilay and Lee, 2004). In their work, an HMM was used to model the transition of *content topics*. The Viterbi decoding (Rabiner, 1990) was performed to find content topics that should be incorporated into a summary. Their approach is similar to ours in that HMMs are utilized to model topic sequences, but they did not use data of multiple domains in creating their model. In addition, their method requires training data (original articles with their reference summaries) in order to find which content topics should be included in a summary, whereas our method requires only the raw sequences with their domain labels.

3 Class Speaker HMM

A Class Speaker HMM (CSHMM) is an extension of Speaker HMM (SHMM), which has been utilized to model two-party conversations (Meguro et al., 2009). In an SHMM, there are two states, and each state emits utterances of one of the two conversational participants. The states are connected ergodically and the emission/transition probabilities are learned from training data by using the EM-algorithm. Although Meguro et al., (2009) used SHMMs to analyze the flow of listening-oriented dialogue, we extend their idea to make it applicable to classification tasks, such as dialogue segmentation.

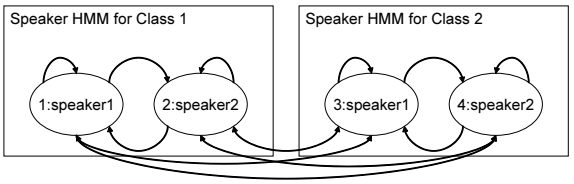


Figure 1: Topology of an ergodic CSHMM. Numbers before 'speaker1' and 'speaker2' denote state IDs.

A CSHMM is simply a concatenation of SHMMs, each of which is trained by using utterance sequences of a particular dialogue class. After such SHMMs are concatenated, the Viterbi algorithm is used to decode an input utterance sequence into class labels by estimating from which class each utterance has most likely to have been generated. Figure 1 illustrates the basic topology of a CSHMM where two SHMMs are concatenated ergodically. When the most likely state sequence for an input utterance sequence is <1,3,4,2>, we can convert these state IDs into their corresponding classes; that is, <1,2,2,1>, which becomes the result of utterance classification.

We have conceived three variations of CSHMM as we describe below. They differ in how we treat utterance sequences that appear commonly in all classes and how we train the transition probabilities between independently trained SHMMs.

3.1 Ergodic CSHMM

The most basic CSHMM is the ergodic CSHMM, which is a simple concatenation of SHMMs in an ergodic manner as shown in Fig. 1. For K classes, K SHMMs are combined with the initial and transition probabilities all set to *equal*. In this CSHMM, the assignment of class labels solely depends on the output distributions of each class.

3.2 Ergodic CSHMM with Common States

This type of CSHMM is the same as the ergodic CSHMM except that it additionally has a SHMM trained from all dialogues of all classes. There-

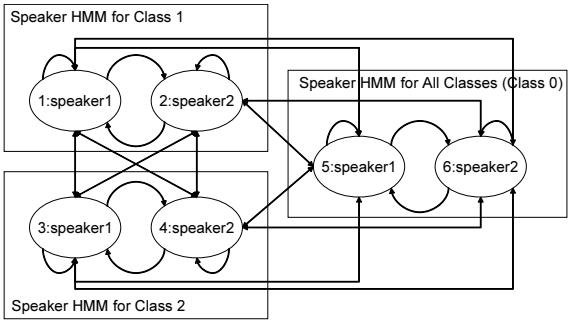


Figure 2: CSHMM with common states.

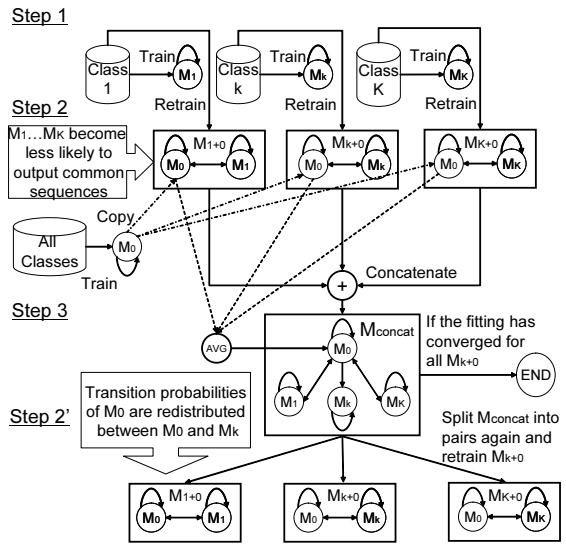


Figure 3: Three steps to create a CSHMM using concatenated training.

fore, for K classes, this CSHMM has $K + 1$ SHMMs. Figure 2 shows the model topology. This newly added SHMM works in a manner similar to the background model (Reynolds et al., 2000) representing sequences that are common to all classes. By having these *common states*, common utterance sequences can be classified as 'common', making it possible to avoid forcefully classifying common utterance sequences into one of the given classes.

Detecting common sequences is especially helpful when several classes overlap in nature. For example, most dialogues commonly start and end with greetings, and many calls at contact centers commonly contain exchanges in which the operator requests personal information about the caller for confirmation. Regarding the model topology in Fig. 2, if the most likely state sequence by the Viterbi decoding is <1,4,5,6,3,2>, we obtain a class label sequence <1,2,0,0,2,1> where the third and fourth utterances are classified as 'zero', meaning that they do not belong to any class.

3.3 CSHMM using Concatenated Training

The CSHMMs presented so far have two problems: one is that the order of utterances of different classes cannot be taken into account because of the equal transition probabilities. As a result, the very merit of HMMs, their ability to model time series data, is lost. The other is that the output distributions of common states may be overly broad because they are the averaged distributions over all classes; that is, the best path determined by the Viterbi decoding may not go through the common states at all.

Our solution to these problems is to apply concatenated training (Lee, 1989), which has been successfully used in speech recognition to model phoneme sequences in an unsupervised manner. The procedure for concatenated training is illustrated in Fig. 3 and has three steps.

step 1 Let M_k ($M_k \in M, 1 \leq k \leq K$) be the SHMM trained using dialogues D_k where $D_k = \{\forall d_j | c(d_j) = k\}$, and M_0 be the SHMM trained using all dialogues; i.e., D. Here, K means the total number of classes and $c(d_j)$ the class assigned to a dialogue d_j.

step 2 Connect each $M_k \in M$ with a copy of M_0 using equal initial and transition probabilities (we call this connected model M_{k+0}) and retrain M_{k+0} with $\forall d_j \in D_k$ where $c(d_j) = k$.

step 3 Merge all models M_{k+0} ($1 \leq k \leq K$) to produce one concatenated HMM (M_{concat}). Here, the output probabilities of the copies of M_0 are averaged over K when all models are merged to create a combined model. If the fitting of all M_{k+0} models has converged against the training data, exit this procedure; otherwise, go to step 2 by connecting a copy of M_0 and M_k for all k. Here, the transition probabilities from M_0 to $M_l (l \neq k)$ are summed and equally distributed between the copied M_0's self-loop and transitions to the states in M_k.

In concatenated training, the transition and output probabilities can be optimized between M_0 and

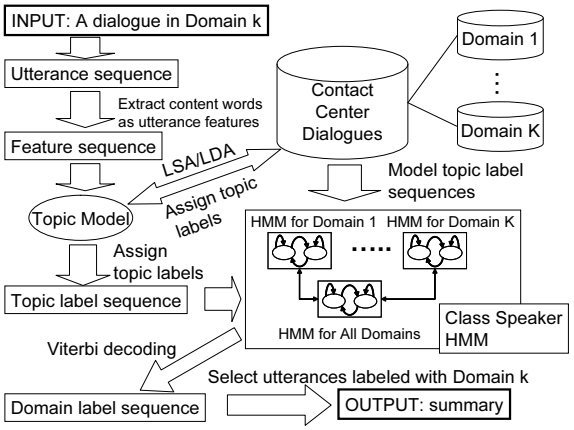


Figure 4: Overview of our summarization method.

M_k, meaning that the output probabilities of utterance sequences that are common and also found in M_k can be moved from M_k to M_0. This makes the distribution of M_k sharp (not broad/uniform), making it likely to output only the utterances representative of a class k. As regards M_0, its distribution of output probabilities can also be sharpened for utterances that occur commonly in all classes. This sharpening of distributions is likely to be helpful for class discrimination.

4 Summarization Method

We apply CSHMMs to extractive summarization of contact center dialogues because such dialogues are two-party, can be categorized into multiple classes by their call domains (e.g., inquiry types), and are likely contain many overlapping exchanges between an operator and a caller across domains, such as greetings, the confirmation of personal information, and other cliches in business (e.g., name exchanges, thanking/apologizing phrases, etc.), making them the ideal target for CSHMMs.

In our method, summarization is performed by decoding a sequence of utterances of a domain DM^k into domain labels and selecting those utterances that have domain labels DM^k. This makes it possible to extract utterances that are characteristic of DM^k in relation to other domains. Our assumption is that extracting characteristic sequences of a given domain provides a good summary for that domain because such sequences should contain important information necessitated by the domain.

Figure 4 outlines our extractive summarization process. The process consists of a training phase and a decoding phase as described below.

Training phase: Let D ($d_1 \ldots d_N$) be the entire set of contact center dialogues, DM^k ($DM^k \in DM, 1 \leq k \leq K$) the domain assigned to domain k, and $U_{d_i,1} \ldots U_{d_i,H}$ the utterances in d_i. Here, H is the number of utterances in d_i. From D, we create two models: a topic model (TM) and a CSHMM.

The topic model is used to assign a single topic to each utterance so as to facilitate the training of the CSHMM by reducing the dimensions of the feature space. The same approach has been taken in (Barzilay and Lee, 2004). The topic model can be created by such techniques as probabilistic latent semantic analysis (PLSA) (Šingliar and Hauskrecht, 2006) and latent Dirichlet allocation (LDA) (Tam and Schultz, 2005). PLSA models the latent topics of the documents and its Baysian extension is LDA, which also models the co-occurrence of topics using the Dirichlet prior. We first derive features $F_{d_1} \ldots F_{d_N}$ for the dialogues. Here, we assume a bag-of-words representation for the features; therefore, F_{d_i} is represented as $\{< w_1, c_1 > \ldots < w_V, c_V >\}$, where V means the total number of content words in the vocabulary and $< w_i, c_i >$ denotes that a content word w_i appears c_i times in a dialogue. Note that we derive the features for dialogues, not for utterances, because utterances in dialogue can be very short, often consisting of only one or two words and thus making it hard to calculate the word co-occurrence required for creating a topic model. From the features, we build a topic model that includes $\mathrm{P}(z|w)$, where w is a word and z is a topic. Using the topic model, we can assign a single topic label to every utterance in D by finding its likely topic; i.e., $\underset{z}{\mathrm{argmax}} \sum_{w \in \mathrm{words}(U_{d_i})} \mathrm{P}(z|w)$.

After labeling all utterances in D with topic labels, we train a CSHMM that learns characteristic topic label sequences in each domain as well as common topic label sequences across domains.

Decoding phase: Let d_j be the input dialogue, $DM(d_j)$ ($\in DM$) the table for obtaining the domain label of d_j, and $U_{d_j,1} \ldots U_{d_j,H_{d_j}}$ the utterances in d_j, where H_{d_j} is the number of the utterances. We use TM to map the utterances to topic

Domain	# Tasks	Sentences	Characters
FIN	15	8.93	289.93
ISP	15	7.20	259.53
LGU	20	9.85	328.55
MO	15	10.07	326.20
PC	15	9.40	354.07
TEL	18	8.44	322.22
ALL	98	9.01	314.46

Table 1: Scenario statistics: the number of tasks and averaged number of sentences/characters in a task scenario in the six domains.

labels $T_{d_j,1} \ldots T_{d_j,H_{d_j}}$ and convert them into domain label sequences $DM_{d_j,1} \ldots DM_{d_j,H_{d_j}}$ using the trained CSHMM by the Viterbi decoding. Then, we select $U_{d_j,h}$ $(1 \leq h \leq H_{d_j})$ whose corresponding domain label $DM_{d_j,h}$ equals $DM(d_j)$ and output the selected utterances in the order of appearance in the original dialogue as a summary.

5 Experiment

We performed an experiment to verify our summarization method. We first collected simulated contact center dialogues using human subjects. Then, we compared our method with baseline systems. Finally, we analyzed the created summaries to investigate what had been learned by our CSHMMs.

5.1 Dialogue Data

Since we do not have access to actual contact center data, we recruited human subjects to collect simulated contact center dialogues. A total of 90 participants (49 males and 41 females) took the roles of operator or a caller and talked over telephones in separate rooms. The callers were given realistic scenarios that included their motivation for a call as well as detailed instructions about what to ask. The operators, who had experience of working at contact centers, were given manuals containing the knowledge of the domain and explaining how to answer questions in specific scenarios.

The dialogues took place in six different domains: Finance (**FIN**), Internet Service Provider (**ISP**), Local Government Unit (**LGU**), Mail Order (**MO**), PC support (**PC**), and Telecommunication (**TEL**). In each domain, there were 15–20 tasks. Table 1 shows the statistics of the task scenarios used by the callers. We cannot describe the details of each domain for lack of space, but examples of the tasks for FIN are inquiries about insurance, notifications of the loss of credit cards, and applications for finance loans, and those for ISP are inquiries about fees for Internet access, requests to forward emails, and reissuance of passwords. Figure 5 shows one of the task scenarios in the MO domain.

MO task No. 3: It is becoming a good season for the Japanese Nabe (pan) cuisine. You own a Nabe restaurant and it is going well. When you were searching on the Internet, thinking of creating a new dish, you saw that drop-shipped Shimonoseki puffer fish was on sale. Since you thought the puffer fish cuisine would become hot in the coming season, you decided to order it as a trial. ... You ordered a puffer fish set on the Internet, but you have not received the confirmation email that you were supposed to receive. ... You decided to call the contact center to make an inquiry, ask them whether the order has been successful, and request them to send you the confirmation email.

Figure 5: Task scenario in the MO domain. The scenario was originally in Japanese and was translated by the authors.

We collected data on two separate occasions using identical scenarios but different participants, which gave us two sets of dialogue data. We used the former for training our summarization system and the latter for testing. We only use the transcriptions in this paper so as to avoid particular problems of speech. All dialogues were in Japanese. Tables 2 and 3 show the statistics of the training data and the test data, respectively. As can be seen from the tables, each dialogue is quite long, which attests to the complexity of the tasks.

5.2 Training our Summarization System

For training our system, we first created a topic model using LDA. We performed a morphological analysis using ChaSen[1] to extract content words from each dialogue and made its bag-of-words features. We defined content words as nouns, verbs, adjectives, unknown words, and interjections (e.g., "yes", "no", "thank you", and "sorry"). We included interjections because they occur very frequently in dialogues and often possess important content, such as agreement and refusal, in transactional communication. We use this definition of content words throughout the paper.

Then, using an LDA software package[2], we built a topic model. We tentatively set the number

[1] http://chasen-legacy.sourceforge.jp/

[2] http://chasen.org/~daiti-m/dist/lda/

		Utterances/Dial.			Characters/Utt.		
Domain	# dial.	OPE	CAL	Both	OPE	CAL	Both
FIN	59	75.73	72.69	148.42	17.44	7.54	12.59
ISP	64	55.09	53.17	108.27	20.11	8.03	14.18
LGU	76	58.28	50.55	108.83	12.83	8.55	10.84
MO	70	66.39	58.74	125.13	15.09	7.43	11.49
PC	56	89.34	77.80	167.14	15.48	6.53	11.31
TEL	66	75.58	63.97	139.55	12.74	8.24	10.67
ALL	391	69.21	61.96	131.17	15.40	7.69	11.76

Table 2: Training data statistics: Averaged number of utterances per dialogue and characters per utterance for each domain. OPE and CAL denote operator and caller, respectively. See Section 5.1 for the full domain names.

		Utterances/Dial.			Characters/Utt.		
Domain	# dial.	OPE	CAL	Both	OPE	CAL	Both
FIN	60	73.97	61.05	135.02	14.53	7.50	11.35
ISP	59	76.08	61.24	137.32	15.43	6.94	11.65
LGU	56	66.55	51.59	118.14	14.54	7.53	11.48
MO	47	75.53	64.87	140.40	10.53	6.79	8.80
PC	44	124.02	94.16	218.18	14.23	7.79	11.45
TEL	41	93.71	68.54	162.24	13.94	7.85	11.37
ALL	307	83.07	65.69	148.76	13.98	7.41	11.08

Table 3: Test data statistics.

of topics to 100. Using this topic model, we labeled all utterances in the training data using these 100 topic labels.

We trained seven different CSHMMs in all: one ergodic CSHMM (**ergodic0**), three variants of ergodic CSHMMs with common states (**ergodic1**, **ergodic2**, **ergodic3**), and three variants of CSHMMs with concatenated training (**concat1**, **concat2**, **concat3**). The difference within the variants is in the number of common states. The numbers 0–3 after 'ergodic' and 'concat' indicate the number of SHMMs containing common states. For example, ergodic3 has nine SHMMs (six SHMMs for the six domains plus three SHMMs containing common states). Since more states would enable more minute modeling of sequences, we made such variants in the hope that common sequences could be more accurately modeled. We also wanted to examine the possibility of creating sharp output distributions in common states without the concatenated training by such minute modeling. These seven CSHMMs make seven different summarization systems.

5.3 Baselines

Baseline-1: BL-TF We prepared two baseline systems for comparison. One is a simple summarizer based on the maximum coverage of high term frequency (TF) content words. We call this baseline BL-TF. This baseline summarizes a dialogue by maximizing the following objective function:

$$\max \sum_{z_i \in Z} \text{weight}(w_i) \cdot z_i$$

where 'weight' returns the importance of a content word w_i and z_i is a binary value indicating whether to include w_i in the summary. Here, 'weight' returns the count of w_i in a given dialogue. The maximization is done using ILP (we used an off-the-shelf solver `lp_solve`[3]) with the following three constraints:

$$x_i, z_i \in \{0, 1\}$$

$$\sum_{x_i \in X} l_i x_i \leq K$$

$$\sum_i m_{ij} x_i \geq z_j \quad (\forall z_j \in Z)$$

where x_i is a binary value that indicates whether to include the i-th utterance in the summary, l_i is the length of the i-th utterance, K is the maximum number of characters to include in a summary, and m_{ij} is a binary value that indicates whether w_i is included in the j-th utterance. The last constraint means that if a certain utterance is included in the summary, all words in that utterance have to be included in the summary.

Baseline-2: BL-DD Although BL-TF should be a very competitive baseline because it uses the state-of-the-art formulation as noted in (Gillick and Favre, 2009), having only this baseline is rather unfair because it does not make use of the training data, whereas our proposed method uses them. Therefore, we made another baseline that learns domain-specific dictionaries (DDs) from the training data and incorporates them into the weights of content words of the objective function of BL-TF. We call this baseline BL-DD. In this baseline, the weight of a content word w_i in a domain DM^k is

$$\text{weight}(w_{i,DM^k}) = \frac{\log(\text{P}(w_i|DM^k))}{\log(\text{P}(w_i|DM \backslash DM^k))}$$

[3] http://lpsolve.sourceforge.net/5.5/

	Metric	ergodic0	ergodic1	ergodic2	ergodic3	concat1	concat2	concat3
PROPOSED	F	0.177	0.177	0.177	0.177	0.187^{*e0e1}_{e2e3}	0.198^{*+e0e1}_{e2e3c1}	$\mathbf{0.199}^{*+e0e1}_{e2e3c1}$
	precision	0.145	0.145	0.145	0.145	0.161^{*}	0.191^{*+}	$\mathbf{0.195}^{*+}$
	recall	**0.294**	**0.294**	**0.294**	**0.294**	0.280^{*}	0.259^{*+}	0.259^{*+}
(Same-length) BL-TF	F	**0.171**	**0.171**	**0.171**	**0.171**	0.168	0.164	0.163
	precision	0.132	0.132	0.132	0.132	0.135	**0.140**	0.140
	recall	**0.294**	**0.294**	**0.294**	**0.294**	0.270	0.241	0.240
(Same-length) BL-DD	F	0.189	0.189	0.189	0.189	**0.189**	0.187	0.187
	precision	0.155	0.155	0.155	0.155	0.162	0.170	**0.172**
	recall	**0.287**	**0.287**	**0.287**	**0.287**	0.273	0.250	0.248
Compression Rate		0.42	0.42	0.42	0.42	0.37	0.30	0.30

Table 4: F-measure, precision, and recall averaged over all 307 dialogues (cf. Table 3) in the test set for the proposed methods and baselines BL-TF and BL-DD configured to output the same-length summaries as the proposed systems. The averaged compression rate for each proposed system is shown at the bottom. The columns (ergodic0–concat3) indicate our methods as well as the character lengths used by the baselines. Asterisks, '+', e0–e3, and c1–c3 indicate our systems' statistical significance by the Wilcoxon signed-rank test ($p<0.01$) over BL-TF, BL-DD, ergodic0–3, and concat1–3, respectively. Statistical tests for the precision and recall were only performed between the proposed systems and their same-length baseline counterparts. **Bold font** indicates the best score in each row.

where $\mathrm{P}(w_i|DM^k)$ denotes the occurrence probability of w_i in the dialogues of DM^k, and $\mathrm{P}(w_i|DM\backslash DM^k)$ the occurrence probability of w_i in all domains except for DM^k. This log likelihood ratio estimates how much a word is characteristic of a given domain. Incorporating such weights would make a very competitive baseline.

5.4 Evaluation Procedure

We made our seven proposed systems and two baselines (BL-TF and BL-DD) output extractive summaries for the test data. Since one of the shortcomings of our proposed method is its inability to set the compression rate, we made our systems output summaries first and made the baseline systems output their summaries within the character lengths of our systems' summaries.

We used scenario texts (See Fig. 5) as reference data; that is, a dialogue dealing with a certain task is evaluated using the scenario text for that task. As an evaluation criterion, we used the F-measure (F1) to evaluate the retrieval accuracy on the basis of the recall and precision of retrieved content words. We used the scenarios as references because they contain the basic content exchanged between an operator and a caller, the retrieval accuracy of which should be important for quality analysts.

We could have used ROUGE (Lin and Hovy, 2003), but we did not because ROUGE does not correlate well with human judgments in conversational data (Liu and Liu, 2008). Another benefit of using the F-measure is that summaries of varying lengths can be compared.

5.5 Results

Table 4 shows the evaluation results for the proposed systems and the baselines. It can be seen that concat3 shows the best performance in F-measure among all systems, having a statistically better performance over all systems except for concat2. The CSHMMs with concatenated training were all better than ergodic0–3. Here, the performance (and output) of ergodic0–3 was exactly the same. This happened because of the broad distributions in their common states; no paths went through the common states and all paths went through the SHMMs of the six domains instead.

The evaluation results in Table 4 may be rather in favor of our systems because the summarization lengths were set by the proposed systems. Therefore, we performed another experiment to investigate the performance of the baselines with varying compression rates and compared their performance with the proposed systems in F-measure. We found that the best performance was achieved by BL-DD when the compression rate was 0.4 with the F-measure of 0.191, which concat3 significantly outperformed by the Wilcoxon signed-rank test ($p<0.01$). Note that the performance shown in Table 4 may seem low. However, we found that the maximum recall is 0.355 (cal-

CAL1	When I order a product from you, I get a confirmation email
CAL2	**Puffer fish**
CAL3	**Sets I have ordered, but I haven't received the confirmation email**
OPE1	**Order**
OPE2	**I will make a confirmation whether you have ordered**
OPE3	Ten sets of Shimonoseki puffer fish by dropship
OPE4	"Yoriai" (name of the product)
OPE5	**Two kilos of bony parts of tiger puffer fish**
OPE6	Baked fins for fin sake
OPE7	**600 milliliter of puffer fish soy sauce**
OPE8	**And, grated radish and red pepper**
OPE9	Your desired delivery date is the 13th of February
CAL4	Yes, all in small cases
CAL5	This is q in alphabet right?
CAL6	Hyphen g
CAL7	**You mean that the order was successful**
OPE10	Yes, it was Nomura at JDS call center

Figure 6: Example output of concat3 for MO task No. 3 (cf Fig. 5). The utterances were translated by the authors. The compression rate for this dialogue was 0.24.

culated by using summaries with no compression). This means that the maximum F-measure we could attain is lower than 0.524 (when the precision is ideal with 1). This is because of the differences between the scenarios and the actual dialogues. We want to pursue ways to improve our evaluation methodology in the future.

Despite such issues in evaluation, from the results, we conclude that our extractive summarization method is effective and that having the common states and training CSHMMs with concatenated training are useful in modeling domain-specific sequences of contact center dialogues.

5.6 Example of System Output

Figure 6 shows an example output of concat3 for the scenario MO task No. 3 (cf. Fig. 5). **Bold font** indicates utterances that were NOT included in the summary of concat3's same-length-BF-DD counterpart. It is clear that sequences related to the MO domain were successfully extracted. When we look at the summary of BF-DD, we see such utterances as "*Can I have your address from the postcode*" and "*Finally, can I have your email address*", which are obvious cliches in contact center dialogues. This indicates the usefulness of common states for ignoring such common exchanges.

6 Summary and Future Work

This paper proposed a novel extractive summarization method for contact center dialogues. We devised a particular type of HMM called CSHMM, which processes operator/caller utterance sequences of multiple domains simultaneously to model domain-specific utterance sequences and common sequences at the same time. We trained a CSHMM using the transcripts of simulated contact center dialogues and verified its effectiveness for the summarization of calls.

There still remain several limitations in our approach. One is its inability to change the compression rate, which we aim to solve in the next step using the forward-backward algorithm (Rabiner and Juang, 1986). This algorithm can calculate the posterior probability of each state at each time frame given an input dialogue sequence, enabling us to extract top-N domain-specific sequences. We also need to find the appropriate topic number for the topic model. In our implementation, we used a tentative value of 100, which may not be appropriate. In addition, we believe the topic model and the CSHMM can be unified because these models are fundamentally similar, especially when LDA is employed. Model topologies may also have to be reconsidered. In our CSHMM with concatenated training, the states in domain-specific SHMMs are only connected to the common states, which may be inappropriate because there could be a case where a domain changes from one to another without having a common sequence. Applying CSHMMs to speech and other NLP tasks is another challenge. As a near-term goal, we aim to apply our method to the summarization of meetings, where we will need to extend our CSHMMs to deal with more than two participants. Finally, we also want to build a contact center dialogue agent by extending the CSHMMs to partially observable Markov decision processes (POMDPs) (Williams and Young, 2007) by following the recent work on building POMDPs from dialogue data in the dynamic Bayesian network (DBN) framework (Minami et al., 2009).

Acknowledgments

We thank the members of the Spoken Dialog System Group, especially Noboru Miyazaki and Satoshi Kobashikawa, for their effort in dialogue data collection.

References

Barzilay, Regina and Lillian Lee. 2004. Catching the drift: Probabilistic content models, with applications to generation and summarization. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL)*, pages 113–120.

Byrd, Roy J., Mary S. Neff, Wilfried Teiken, Youngja Park, Keh-Shin F. Cheng, Stephen C. Gates, and Karthik Visweswariah. 2008. Semi-automated logging of contact center telephone calls. In *Proceeding of the 17th ACM conference on Information and knowledge management (CIKM)*, pages 133–142.

Chu-Carroll, Jennifer and Bob Carpenter. 1999. Vector-based natural language call routing. *Computational Linguistics*, 25(3):361–388.

Gillick, Dan and Benoit Favre. 2009. A scalable global model for summarization. In *Proceedings of the Workshop on Integer Linear Programming for Natural Language Processing*, pages 10–18.

Gorin, Allen L., Giuseppe Riccardi, and Jerry H. Wright. 1997. How may I help you? *Speech Communication*, 23(1-2):113–127.

Hori, Chiori and Sadaoki Furui. 2003. A new approach to automatic speech summarization. *IEEE Transactions on Multimedia*, 5(3):368–378.

Kupiec, Julian, Jan Pedersen, and Francine Chen. 1995. A trainable document summarizer. In *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR)*, pages 68–73.

Lee, Kai-Fu. 1989. *Automatic speech recognition: the development of the SPHINX system*. Kluwer Academic Publishers.

Lin, Chin-Yew and Eduard Hovy. 2003. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology (NAACL-HLT)*, pages 71–78.

Liu, Feifan and Yang Liu. 2008. Correlation between ROUGE and human evaluation of extractive meeting summaries. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies (HLT)*, pages 201–204.

Mani, Inderjeet. 2001. *Automatic summarization*. John Benjamins Publishing Company.

Meguro, Toyomi, Ryuichiro Higashinaka, Kohji Dohsaka, Yasuhiro Minami, and Hideki Isozaki. 2009. Analysis of listening-oriented dialogue for building listening agents. In *Proceedings of the SIGDIAL 2009 conference*, pages 124–127.

Minami, Yasuhiro, Akira Mori, Toyomi Meguro, Ryuichiro Higashinaka, Kohji Dohsaka, and Eisaku Maeda. 2009. Dialogue control algorithm for ambient intelligence based on partially observable Markov decision processes. In *Proceedings of the 1st international workshop on spoken dialogue systems technology (IWSDS)*, pages 254–263.

Murray, Gabriel, Steve Renals, and Jean Carletta. 2005. Extractive summarization of meeting recordings. In *Proceedings of the 9th European Conference on Speech Communication and Technology (EUROSPEECH)*, pages 593–596.

Osborne, Miles. 2002. Using maximum entropy for sentence extraction. In *Proceedings of the ACL-02 Workshop on Automatic Summarization*, pages 1–8.

Rabiner, Lawrence R. and Biing-Hwang Juang. 1986. An introduction to hidden Markov models. *IEEE ASSP Magazine*, 3(1):4–16.

Rabiner, Lawrence R. 1990. A tutorial on hidden Markov models and selected applications in speech recognition. *Readings in speech recognition*, 53(3):267–296.

Radev, Dragomir R., Hongyan Jing, Małgorzata Styś, and Daniel Tam. 2004. Centroid-based summarization of multiple documents. *Information Processing & Management*, 40(6):919–938.

Reynolds, Douglas A., Thomas F. Quatieri, and Robert B. Dunn. 2000. Speaker verification using adapted Gaussian mixture models. *Digital Signal Processing*, 10(1-3):19 – 41.

Subramaniam, L. Venkata, Tanveer A. Faruquie, Shajith Ikbal, Shantanu Godbole, and Mukesh K. Mohania. 2009. Business intelligence from voice of customer. In *Proceedings of the 2009 IEEE International Conference on Data Engineering (ICDE)*, pages 1391–1402.

Takeuchi, Hironori, L Venkata Subramaniam, Tetsuya Nasukawa, Shourya Roy, and Sreeram Balakrishnan. 2007. A conversation-mining system for gathering insights to improve agent productivity. In *Proceedings of the IEEE International Conference on E-Commerce Technology and the IEEE International Conference on Enterprise Computing, E-Commerce, and E-Services*, pages 465–468.

Tam, Yik-Cheung and Tanja Schultz. 2005. Dynamic language model adaptation using variational Bayes inference. In *Proceedings of the 9th European Conference on Speech Communication and Technology (EUROSPEECH)*, pages 5–8.

Šingliar, Tomas and Milos Hauskrecht. 2006. Noisy-OR component analysis and its application to link analysis. *The Journal of Machine Learning Research*, 7:2189–2213.

Williams, Jason D. and Steve Young. 2007. Partially observable Markov decision processes for spoken dialog systems. *Computer Speech & Language*, 21(2):393–422.

Recognizing Relation Expression between Named Entities based on Inherent and Context-dependent Features of Relational words

Toru Hirano[†], **Hisako Asano**[‡], **Yoshihiro Matsuo**[†], **Genichiro Kikui**[†]
[†]NTT Cyber Space Laboratories, NTT Corporation
[‡]Innovative IP Architecture Center, NTT Communications Corporation
hirano.tohru@lab.ntt.co.jp
hisako.asano@ntt.com
{matsuo.yoshihiro,kikui.genichiro}@lab.ntt.co.jp

Abstract

This paper proposes a supervised learning method to recognize expressions that show a relation between two named entities, e.g., person, location, or organization. The method uses two novel features, 1) whether the candidate words inherently express relations and 2) how the candidate words are influenced by the past relations of two entities. These features together with conventional syntactic and contextual features are organized as a tree structure and are fed into a boosting-based classification algorithm. Experimental results show that the proposed method outperforms conventional methods.

1 Introduction

Much attention has recently been devoted to using enormous amount of web text covering an exceedingly wide range of domains as a huge knowledge resource with computers. To use web texts as knowledge resources, we need to extract information from texts that are merely sequences of words and convert them into a structured form. Although extracting information from texts as a structured form is difficult, relation extraction is a way that makes it possible to use web texts as knowledge resources.

The aim of relation extraction is to extract semantically related named entity pairs, X and Y, and their relation, R, from a text as a structured form [X, Y, R]. For example, the triple [Yukio Hatoyama, Japan, prime minister] would be extracted from the text "Yukio Hatoyama is the prime minister of Japan". This extracted triple provides important information used in information retrieval (Zhu et al., 2009) and building an ontology (Wong et al., 2010).

It is possible to say that all named entity pairs that co-occur within a text are semantically related in some way. However, we define that named entity pairs are semantically related if they satisfy either of the following rules:

- One entity is an attribute value of the other.
- Both entities are arguments of a predicate.

Following the above definition, explicit and implicit relations should be extracted. An explicit relation means that there is an expression that shows the relation between a named entity pair in a given text, while an implicit relation means that there is no such expression. For example, the triple [Yukio Hatoyama, Kunio Hatoyama, brother] extracted from the text "Yukio Hatoyama, the Democratic Party, is Kunio Hatoyama's brother" is an explicit relation. In contrast, the triple [Yukio Hatoyama, the Democratic Party, member] extracted from the same text is an implicit relation because there is no expression showing the relation (e.g. member) between "Yukio Hatoyama" and "the Democratic Party" in the text.

Extracting triples [X, Y, R] from a text involves two tasks. One is detecting semantically related pairs from named entity pairs that co-occur in a text and the other is determining the relation between a detected pair. For the former task, various supervised learning methods (Culotta and Sorensen, 2004; Zelenko et al., 2003; Hirano et al., 2007) and bootstrapping methods (Brin, 1998; Pantel and Pennacchiotti, 2006) have been explored to date. In contrast, for the latter task,

Coling 2008: Poster Volume, pages 409–417,
Beijing, August 2010

only a few methods have been proposed so far (Hasegawa et al., 2004; Banko and Etzioni, 2008; Zhu et al., 2009). We therefore addressed the problem of how to determine relations between a given pair.

We used a three-step approach to address this problem. The first step is to recognize an expression that shows explicit relations between a given named entity pair in a text. If no such expression is recognized, the second step is to estimate the relationship that exists between a given named entity pair that has an implicit relation. The last step is to identify synonyms of the relations that are recognized or estimated in the above steps. In this paper, we focus on the first step. The task is selecting a phrase from the text that contains a relation expression linking a given entity pair and outputting the expression as one showing the relationship between the pair.

In our preliminary experiment, it was found that using only structural features of a text, such as syntactic or contextual features, is not good enough for a number of examples. For instance, the two Japanese sentences shown in Figure 1 have the same syntactic structure but (a) contains a relation expression and (b) does not. We therefore assume there are clues for recognizing relation expressions other than conventional syntactic and contextual information. In this paper, we propose a supervised learning method that includes two novel features of relational words as well as conventional syntactic and contextual features. The novel features of our method are:

Inherent Feature: Some words are able to express the relations between named entities and some are not. Thus, it would be useful to know the words that inherently express these relations.

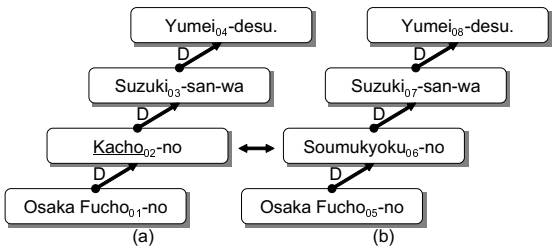
Context-dependent Feature: There are a number of typical relationships that change as time passes, such as "dating" ⇒ "engagement" ⇒ "marriage" between persons. Furthermore, present relations are influenced by the past relations of a given named entity pair. Thus, it would be useful to know the past relations between a given pair and how the relations change as time passes.

In the rest of this paper, Section 2 references related work, Section 3 outlines our method's main features and related topics, Section 4 describes our experiments and experimental results, and Section 5 briefly summarizes key points and future work to be done.

2 Related Work

The "Message Understanding Conference" and "Automatic Content Extraction" programs have promoted relational extraction. The task was studied so as to extract predefined semantic relations of entity pairs in a text. Examples include the supervised learning method cited in (Kambhatla, 2004; Culotta and Sorensen, 2004; Zelenko et al., 2003) and the bootstrapping method cited in (Pantel and Pennacchiotti, 2006; Agichtein and Gravano, 2000). Recently, open information extraction (Open IE), a novel domain-independent extraction paradigm, has been suggested (Banko and Etzioni, 2008; Hasegawa et al., 2004). The task is to detect semantically related named entity pairs and to recognize the relation between them without using predefined relations.

Our work is a kind of open IE, but our approach differs from that of previous studies. Banko (2008) proposed a supervised learning method using conditional random fields to recognize the relation expressions from words located between a given pair. Hasegawa (2004) also proposed a rule-based method that selects all words located between a given pair as a relation expression if a given named entities appear within ten words. The point of these work is that they selected relation expressions only from the words located between



(a) Mr.Suzuki$_{03}$, a manager$_{02}$ of Osaka Prefectural Government$_{01}$, is famous$_{04}$.
(b) Mr.Suzuki$_{07}$, administration office$_{06}$ in Osaka Prefectural Government$_{05}$, is famous$_{08}$.

Figure 1: Same syntactic examples

given entities in the text, because as far as English texts are concerned, 86% of the relation expressions of named entity pairs appear between the pair (Banko and Etzioni, 2008). However, our target is Japanese texts, in which only 26% of entity pair relation expressions appear between the pair. Thus, it is hard to incorporate previous approaches into a Japanese text.

To solve the problem, our task was to select a phrase from the entire text that would include a relation expression for connecting a given pair.

3 Recognizing Relation Expressions between Named Entities

To recognize the relation expression for a given pair, we need to select a phrase that includes an expression that shows the relation between a given entity pair from among all noun and verb phrases in a text. Actually, there are two types of candidate phrases in this case. One is from a sentence that contains a given pair (intra-sentential), and the other is from a sentence that does not (inter-sentential). For example, the triple [Miyaji$_{21}$, Ishii$_{22}$, taiketsu$_{12}$] extracted from the following text is inter-sentential.

(S-1) Chumokoku$_{11}$-no taiketsu$_{12}$-ga mamonaku$_{13}$ hajimaru$_{14}$.
(The showcase$_{11}$ match$_{12}$ will start$_{14}$ soon$_{13}$.)

(S-2) Ano Miyaji$_{21}$-to Ishii$_{22}$-toiu kanemochi$_{23}$-niyoru yume$_{24}$-no kikaku$_{25}$.
(The dream$_{24}$ event$_{25}$ between the rich mens$_{23}$, Miyaji$_{21}$ and Ishii$_{22}$.)

According to our annotated data shown in Table 2, 53% of the semantically-related named entity pairs are intra-sentential and 12% are inter-sentential. Thus, we first select a phrase from those in a sentence that contains a given pair, and if no phrase is selected, select one from the rest of the sentences in a text.

We propose a supervised learning method that uses two novel features of relational words as well as conventional syntactic and contextual features. These features are organized as a tree structure and are fed into a boosting-based classification algorithm (Kudo and Matsumoto, 2004). The highest-scoring phrase is then selected if the score exceeds a given threshold. Finally, the head of the selected phrase is output as the relation expression of a given entity pair.

The method consists of four parts: preprocessing (POS tagging and parsing), feature extraction, classification, and selection. In this section, we describe the idea behind using our two novel features and how they are implemented to recognize the relation expressions of given pairs. Before that, we will describe our proposed method's conventional features.

3.1 Conventional Features

Syntactic feature

To recognize the intra-sentential relation expressions for a given pair, we assume that there is a discriminative syntactic structure that consists of given entities and their relation expression. For example, there is a structure for which the common parent phrase of the given pair, X = "Hatoyama Yukio$_{32}$" and Y = "Hatoyama Kunio$_{33}$", has the relation expression, R = "ani$_{34}$" in the Japanese sentence S-3. Figure 2 shows the dependency tree of sentence S-3.

(S-3) Minshuto$_{31}$-no Hatoyama Yukio$_{32}$-wa Hatoyama Kunio$_{33}$-no ani$_{34}$-desu.
(Yukio Hatoyama$_{32}$, the Democratic Party$_{31}$, is Kunio Hatoyama$_{33}$'s brother$_{34}$.)

To use a discriminative structure for each candidate, we make a minimum tree that consists of given entities and the candidate where each phrase is represented by a case marker "CM", a dependency type "DT", an entity class, and the string and POS of the candidate (See Figure 3).

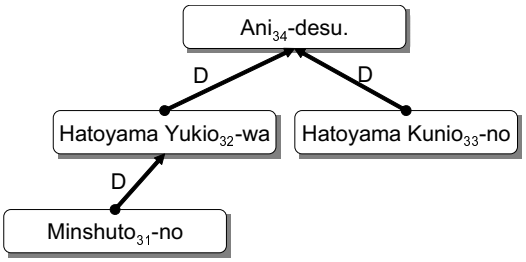


Figure 2: Dependency tree of sentence S-3

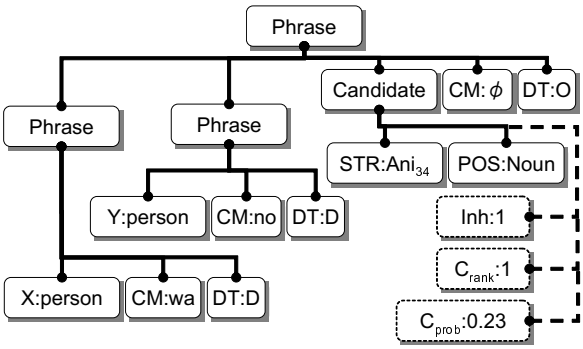


Figure 3: Intra-sentential feature tree

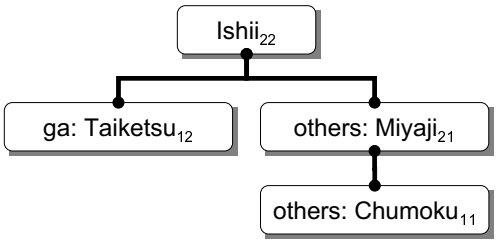


Figure 4: Salient referent list tree

Contextual Feature

To recognize the inter-sentential relation expressions for a given pair, we assume that there is a discriminative contextual structure that consists of given entities and their relation expression. Here, we use a Salient Referent List (SRL) to obtain contextual structure. The SRL is an empirical sorting rule proposed to identify the antecedent of (zero) pronouns (Nariyama, 2002), and Hirano (2007) proposed a way of applying SRL to relation detection. In this work, we use this way to apply SRL to recognize inter-sentential relation expressions.

We applied SRL to each candidate as follows. First, from among given entities and the candidate, we choose the one appearing last in the text as the root of the tree. We then append noun phrases, from the chosen one to the beginning of the text, to the tree depending on case markers, "wa" (topicalised subject), "ga" (subject), "ni" (indirect object),"wo" (object), and "others", with the following rules. If there are nodes of the same case marker already in the tree, the noun phrase is appended as a child of the leaf node of them. In other cases, the noun phrase is appended as a child of the root node. For example, we get the SRL tree shown in Figure 4 with the given entity pair, X = "Miyaji$_{21}$" and Y = "Ishii$_{22}$", and the candidate, "taiketsu$_{12}$", with the text (S-1, S-2).

To use a discriminative SRL structure, we make a minimum tree that consists of given entities and the candidate where each phrase is represented by an entity class, and the string and POS of the candidate (See Figure 5). In this way, there is a problem when the candidate is a verb phrase, because only noun phrases are appended to the SRL tree. If the candidate is a verb phrase, we cannot make a minimum tree that consists of given entities and the candidate.

To solve this problem, a candidate verb phrase is appended to the feature tree using a syntactic structure. In a dependency tree, almost all verb phrases have some parent or child noun phrases that are in the SRL tree. Thus, candidate verb phrases are appended as offspring of these noun phrases represented syntactically as "parent" or "child". For example, when given the entity pair, X = "Miyaji$_{21}$" and Y = "Ishii$_{22}$", and the candidate, "hajimaru$_{14}$" from the text (S-1, S-2), a feature tree cannot be made because the candidate is not in an SRL tree. By extending the way the syntactic structure is used, "hajimaru$_{14}$" has a child node "taiketsu$_{12}$", which is in an SRL tree, and this makes it possible to make the feature tree shown in Figure 6.

3.2 Proposed Features

To recognize intra-sentential or inter-sentential relation expressions for given pairs, we assume there are clues other than syntactic and contextual information. Thus, we propose inherent and

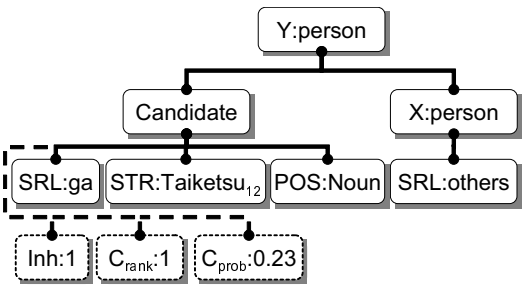


Figure 5: Inter-sentential feature tree

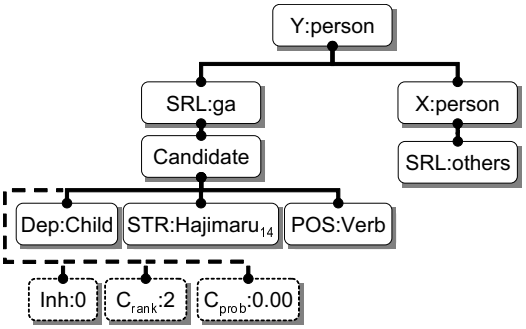


Figure 6: Extended inter-sentential feature tree

context-dependent features of relational words.

Inherent Feature of Relational words

Some words are able to express the relations between named entities and some are not. For example, the word "mother" can express a relation, but the word "car" cannot. If there were a list of words that could express relations between named entities, it would be useful to recognize the relation expression of a given pair. As far as we know, however, no such list exists in Japanese. Thus, we estimate which words are able to express relations between entities. Here, we assume that almost all verbs are able to express relations, and accordingly we focus on nouns.

When the relation expression, R, of an entity pair, X and Y, is a noun, it is possible to say "Y is R of X" or "Y is X's R". Here, we can say noun R takes an argument X. In linguistics, this kind of noun is called a relational noun. Grammatically speaking, a relational noun is a simple noun, but because its meaning describes a "relation" rather than a "thing", it is used to describe relations just as prepositions do. To estimate which nouns are able to express the relations between named entities, we use the characteristics of relational nouns. In linguistics, many researchers describe the relationship between possessives and relational nouns (Chris, 2008). Thus, we use the knowledge that in the patterns "B of A" or "A's B", if word B is a relational noun, the corresponding word A belongs to a certain semantic category. In contrast, if word B is not a relational noun, the corresponding word A belongs to many semantic categories (Tanaka et al., 1999). Figure 7 shows scattering of the semantic categories of "mother" and "car"

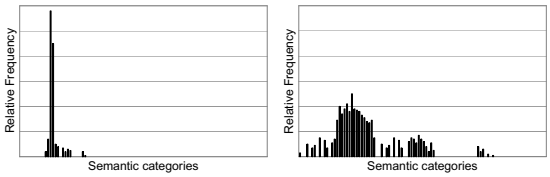


Figure 7: Scattering of semantic category of "mother" (left) and "car" (right).

acquired by the following way.

First, we acquired A and B using the patterns "A no B"[1] from a large Japanese corpus, then mapped words A into semantic categories C= $\{$ $c_1, c_2, \cdots, c_m$ $\}$ using a Japanese lexicon (Ikehara et al., 1999). Next, for each word B, we calculated a scattering score $\mathcal{H}c(B)$ using the semantic category of corresponding words A. Finally, we estimated whether a word is a relational noun by using k-NN estimation with positive and negative examples. As estimated results, "Inh:1" shows that it is a relational noun and "Inh:0" shows that it is not. In both cases, the result is appended to the feature tree as a child of the candidate node (See Figure 3, 5, or 6).

$$\mathcal{H}c(B) = -\sum_{c \in C} P(c|B) log_m P(c|B)$$

$$P(c|B) = \frac{freq(c, B)}{freq(B)}$$

In our experiments, we acquired 55,412,811 pairs of A and B from 1,698,798 newspaper articles and 10,499,468 weblog texts. As training data, we used the words of relation expressions as positive examples and other words as negative examples.

Context-dependent Feature of Relational words

There are a number of typical relationships that change as time passes, such as "dating" ⇒ "engagement" ⇒ 'marriage" between persons. Furthermore, present relations are affected by the past relations of a given named entity pair. For instance, if the past relations of a given pair are "dating" and "engagement" and one of the candidates is "marriage", "marriage" would be selected as the relation expression of the given pair. Therefore, if

[1] "B of A" or "A's B" in English.

Pair of entity class	r_m	r_n	$P_T(r_n\|r_m)$	$Count(r_m, r_n)$
⟨person,person⟩	dating	dating	0.050	102
		marriage	0.050	101
		engagement	0.040	82
⟨person,person⟩	engagement	marriage	0.157	786
		engagement	0.065	325
		wedding	0.055	276
⟨person,organization⟩	vice president	president	0.337	17,081
		vice president	0.316	16,056
		CEO	0.095	4,798
⟨person,organization⟩	researcher	fellow	0.526	61
		manager	0.103	12
		member	0.078	9
⟨organization,organization⟩	alliance	alliance	0.058	8,358
		accommodated	0.027	3,958
		acquisition	0.027	3,863
⟨location,location⟩	neighbour	mutual consultation	0.022	2,670
		support	0.015	1,792
		visit	0.012	1,492
⟨location,location⟩	war	war	0.077	78,170
		mutual consultation	0.015	15,337
		support	0.010	10,226

Table 1: Examples of calculated relation trigger model between entity classes defined by IREX

we know the past relations of the given pair and the typical relational change that occurs as time passes, it would be useful to recognize the relation expression of a given pair.

In this paper, we represent typical relational changes that occur as time passes by a simple relation trigger model $P_T(r_n|r_m)$. Note that r_m is a past relation and r_n is a relation affected by r_m. This model disregards the span between r_n and r_m. To make the trigger model, we automatically extract triples [X, Y, R] from newspaper articles and weblog texts, which have time stamps of the document creation. Using these triples with time stamps for each entity pair, we sort relations in order of time and count pairs of present and previous relations. For example, if we extract "dating" occurring for an entity pair on January 10, 1998, "engagement" occurring on February 15, 2001, and "marriage" occurring on December 24, 2001, the pairs ⟨dating, engagement⟩, ⟨dating, marriage⟩, and ⟨engagement, marriage⟩ are counted. The counted score is then summed up by the pair of entity class and the trigger model is calculated by the following formula.

$$P_T(r_n|r_m) = \frac{Count(r_m, r_n)}{\sum_{r_n} Count(r_m, r_n)}$$

For the evaluation, we extracted triples by named entity recognition (Suzuki et al., 2006), relation detection (Hirano et al., 2007), and the proposed method using the inherent features of relational words described in Section 3.2. A total of 10,463,232 triples were extracted from 8,320,042 newspaper articles and weblog texts with time stamps made between January 1, 1991 and June 30, 2006. As examples of the calculated relation trigger model, Table 1 shows the top three probability relations r_n of several relations r_m between Japanese standard named entity classes defined in the IREX workshop[2]. For instance, the relation "fellow" has the highest probability of being changed from the relation "researcher" between person and organization as time passes.

[2]http://nlp.cs.nyu.edu/irex/

To obtain the past relations of a given pair in the input text, we again used the triples with time stamps extracted as above. The only relations we use as past relations, $R_m = \{r_{m_1}, r_{m_2}, \cdots, r_{m_k}\}$, are those of a given pair whose time stamps are older than the input text. Finally, we calculated probabilities with the following formula using the past relations R_m and the trigger model $P_T(r_n|r_m)$.

$$P_T(r_n|R_m) = max\{P_T(r_n|r_{m_1}), P_T(r_n|r_{m_2}), \cdots, P_T(r_n|r_{m_k})\}$$

Using this calculated probability, we ranked candidates and appended the rank "C$_{rank}$" and the probability score "C$_{prob}$" to the feature tree as a child of the candidate node (See Figure 3, 5, or 6). For example, if the past relations R_m were "dating" and "engagement" and candidates were "marriage", "meeting', "eating", or "drinking", the candidates probabilities were calculated and ranked as "marriage" (C$_{prob}$:0.15, C$_{rank}$:1), "meeting" (C$_{prob}$:0.08, C$_{rank}$:2), etc.

3.3 Classification Algorithms

Several structure-based learning algorithms have been proposed so far (Collins and Duffy, 2002; Suzuki et al., 2003; Kudo and Matsumoto, 2004). The experiments tested Kudo and Matsumoto's boosting-based algorithm using sub-trees as features, which is implemented as a BACT system.

Given a set of training examples each of which is represented as a tree labeling whether the candidate is the relation expression of a given pair or not, the BACT system learns that a set of rules is effective in classifying. Then, given a test instance, the BACT system classifies using a set of learned rules.

4 Experiments

We conducted experiments using texts from Japanese newspaper articles and weblog texts to test the proposed method for both intra- and inter-sentential tasks. In the experiments, we compared the following methods:

Conventional Features: trained by conventional syntactic features for intra-sentential tasks as described in Section 3.1, and contextual features for inter-sentential tasks as described in Section 3.1.

+Inherent Features: trained by conventional features plus inherent features of relational words described in Section 3.2.

++Context-dependent Features$_{\mathrm{TM}}$: trained by conventional and inherent features plus context-dependent features of relational words with the **trigger model** described in Section 3.2.

++Context-dependent Features$_{\mathrm{CM}}$: trained by conventional and inherent features plus context-dependent features of relational words with a **cache model**. We evaluated this method to compare it with Context-dependent Features$_{\mathrm{TM}}$ to show the effectiveness of the proposed trigger model. The cache model is a simple way to use past relations in which the probability $P_C(r_{cand})$ calculated by the following formula and the rank based on the probability is appended to every candidate feature tree.

$$P_C(r_{cand}) = \frac{|r_{cand} \text{ in past relations}|}{|\text{past relations}|}$$

Relation Types		#
Explicit	Intra-sentential	9,178
	Inter-sentential	2,058
Implicit		5,992
Total		17,228

Table 2: Details of the annotated data

4.1 Settings

We used 6,200 texts from Japanese newspapers and weblogs dated from January 1, 2004 to June 30, 2006, manually annotating the semantic relations between named entities for experiment purposes. There were 17,228 semantically-related entity pairs as shown in Table 2. In an intra-sentential experiment, 17,228 entity pairs were given, but only 9,178 of them had relation expressions. In contrast, in an inter-sentential experiment, 8,050 entity pairs excepted intra-sentential

	$Precision$	$Recall$	F
Conventional Features	63.5 % (3,436/5,411)	37.4 % (3,436/9,178)	0.471
+Inherent Features	67.2 % (4,036/6,001)	43.9 % (4,036/9,178)	0.531
++Context-dependent Features$_{\mathbf{TM}}$	70.7 % (4,460/6,312)	48.6 % (4,460/9,178)	**0.576**
++Context-dependent Features$_{\mathbf{CM}}$	67.5 % (4,042/5,987)	44.0 % (4,042/9,178)	0.533

Table 3: Experimental results of intra-sentential

	$Precision$	$Recall$	F
Conventional Features	70.1 % (579/825)	28.1 % (579/2,058)	0.401
+Inherent Features	77.1 % (719/932)	34.9 % (719/2,058)	0.480
++Context-dependent Features$_{\mathbf{TM}}$	75.2 % (794/1,055)	38.5 % (794/2,058)	**0.510**
++Context-dependent Features$_{\mathbf{CM}}$	74.3 % (732/985)	35.5 % (732/2,058)	0.481

Table 4: Experimental result of inter-sentential

were given, but only 2,058 of them had relation expressions.

We conducted five-fold cross-validation over 17,228 entity pairs so that sets of pairs from a single text were not divided into the training and test sets. In the experiments, all features were automatically acquired using a Japanese POS tagger (Fuchi and Takagi, 1998) and dependency parser (Imamura et al., 2007).

4.2 Results

Tables 3 and 4 show the performance of several methods for intra-sentential and inter-sentential. $Precision$ is defined as the percentage of correct relation expressions out of recognized ones. $Recall$ is the percentage of correct relation expressions from among the manually annotated ones. The F measure is the harmonic mean of precision and recall.

A comparison with the Conventional Features and Inherent Features method for intra-/inter-sentential tasks indicates that the proposed method using inherent features of relational words improved intra-sentential tasks F by 0.06 points and inter-sentential tasks F by 0.08 points. Using a statistical test (McNemar Test) demonstrably showed the proposed method's effectiveness.

A comparison with the Inherent Features and Context-dependent Features$_{\mathbf{TM}}$ method showed that the proposed method using context-dependent features of relational words improved intra-/inter-sentential task performance by 0.045 and 0.03 points, respectively. McNemar test results also showed the method's effectiveness.

To further compare the usage of context-dependent features, trigger models, and cache models, we also used Context-dependent Features$_{\mathbf{CM}}$ method for comparison. Tables 3 and 4 show that our proposed trigger model performed better than the cache model, and McNemar test results showed that there was a significant difference between the models. The reason the trigger model performed better than the cache model is that the trigger model correctly recognized the relation expressions that did not appear in the past relations of a given pair. Thus, we can conclude that using typical relationships that change as time passes helps to recognize relation expressions between named entities.

5 Conclusion

We proposed a supervised learning method that employs inherent and context-dependent features of relational words and uses conventional syntactic or contextual features to improve both intra- and inter-sentential relation expression recognition. Our experiments demonstrated that the method improves the F measure and thus helps to recognize relation expressions between named entities.

In future work, we plan to estimate implicit relations between named entities and to identify relational synonyms.

References

Agichtein, Eugene and Luis Gravano. 2000. Snowball: Extracting relations from large plain-text collections. In *Proceedings of the 5th ACM conference on Digital libraries*, pages 85–94.

Banko, Michele and Oren Etzioni. 2008. The tradeoffs between open and traditional relation extraction. In *Proceedings of the 46th Annual Meeting on Association for Computational Linguistics: Human Language Technologies*, pages 28–36.

Brin, Sergey. 1998. Extracting patterns and relations from the world wide web. In *WebDB Workshop at 6th International Conference on Extending Database Technology*, pages 172–183.

Chris, Barker, 2008. *Semantics: An international handbook of natural language meaning*, chapter Possessives and relational nouns. Walter De Gruyter Inc.

Collins, Michael and Nigel Duffy. 2002. Convolution kernels for natural language. *Advances in Neural Information Processing Systems*, 14:625–632.

Culotta, Aron and Jeffrey Sorensen. 2004. Dependency tree kernels for relation extraction. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, pages 423–429.

Fuchi, Takeshi and Shinichiro Takagi. 1998. Japanese morphological analyzer using word co-occurrence - jtag. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, volume 1, pages 409–413.

Hasegawa, Takaaki, Satoshi Sekine, and Ralph Grishman. 2004. Discovering relations among named entities from large corpora. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, pages 415–422.

Hirano, Toru, Yoshihiro Matsuo, and Genichiro Kikui. 2007. Detecting semantic relations between named entities in text using contextual features. In *Proceedings of the 45th Annual Meeting on Association for Computational Linguistics*, pages 157–160.

Ikehara, Satoru, Masahiro Miyazaki, Satoru Shirai, Akio Yoko, Hiromi Nakaiwa, Kentaro Ogura, Masafumi Oyama, and Yoshihiko Hayashi. 1999. *Nihongo Goi Taikei (in Japanese)*. Iwanami Shoten.

Imamura, Kenji, Genichiro Kikui, and Norihito Yasuda. 2007. Japanese dependency parsing using sequential labeling for semi-spoken language. In *Proceedings of the 45th Annual Meeting on Association for Computational Linguistics*, pages 225–228.

Kambhatla, Nanda. 2004. Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, pages 178–181.

Kudo, Taku and Yuji Matsumoto. 2004. A boosting algorithm for classification of semi-structured text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 301–308.

Nariyama, Shigeko. 2002. Grammar for ellipsis resolution in japanese. In *Proceedings of the 9th International Conference on Theoretical and Methodological Issues in Machine Translation*, pages 135–145.

Pantel, Patrick and Marco Pennacchiotti. 2006. Espresso: Leveraging generic patterns for automatically harvesting semantic relations. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 113–120.

Suzuki, Jun, Tsutomu Hirao, Yutaka Sasaki, and Eisaku Maeda. 2003. Hierarchical directed acyclic graph kernel: Methods for structured natural language data. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pages 32–39.

Suzuki, Jun, Erik McDermott, and HIdeki Isozaki. 2006. Training conditional random fields with multivariate evaluation measures. In *Proceedings of the 43th Annual Meeting on Association for Computational Linguistics*.

Tanaka, Shosaku, Yoichi Tomiura, and Toru Hitaka. 1999. Classification of syntactic categories of nouns by the scattering of semantic categories (in japanese). *Transactions of Information Processing Society of Japan*, 40(9):3387–3396.

Wong, Wilson, Wei Liu, and Mohammed Bennamoun. 2010. Acquiring semantic relations using the web for constructing lightweight ontologies. In *Proceedings of the 13th Pacific-Asia Conference on Knowledge Discovery and Data Mining*.

Zelenko, Dmitry, Chinatsu Aone, and Anthony Richardella. 2003. Kernel methods for relation extraction. *Journal of Machine Learning Research*, 3:1083–1106.

Zhu, Jun, Zaiqing Nie, Xiaojing Liu, Bo Zhang, and Ji-Rong Wen. 2009. Statsnowball: a statistical approach to extracting entity relationships. In *Proceedings of the 18th international conference on World Wide Web*, pages 101–110.

Word Sense Disambiguation-based Sentence Similarity

Chukfong Ho[1], Masrah Azrifah Azmi Murad[2]
Department of Information System
University Putra Malaysia
hochukfong@yahoo.com[1],
masrah@fsktm.upm.edu.my[2]

Rabiah Abdul Kadir, Shyamala C. Doraisamy
Department of Multimedia
University Putra Malaysia
{rabiah, shyamala}@fsktm.upm.edu.my

Abstract

Previous works tend to compute the similarity between two sentences based on the comparison of their nearest meanings. However, the nearest meanings do not always represent their actual meanings. This paper presents a method which computes the similarity between two sentences based on a comparison of their actual meanings. This is achieved by transforming an existing most-outstanding corpus-based measure into a knowledge-based measure, which is then integrated with word sense disambiguation. The experimental results on a standard data set show that the proposed method outperforms the baseline and the improvement achieved is statistically significant at 0.025 levels.

1 Introduction

Although measuring sentence similarity is a complicated task, it plays an important role in natural language processing applications. In text categorization (Yang and Wen, 2007), documents are retrieved based on similar or related features. In text summarization (Zhou et al., 2006) and machine translation (Kauchak and Barzilay, 2006), summaries comparison based on sentence similarity has been applied for automatic evaluation. In text coherence (Lapata and Barzilay, 2005), different sentences are linked together based on the sequence of similar or related words.

Two main issues are investigated in this paper: 1) the performance between corpus-based measure and knowledge-based measure, and 2) the influence of word sense disambiguation (WSD) on measuring sentence similarity. WSD is the task of determining the sense of a polysemous word within a specific context (Wang et al., 2006). Corpus-based methods typically compute sentence similarity based on the frequency of a word's occurrence or the co-occurrence between collocated words. Although these methods benefit from the statistical information derived from the corpus, this statistical information is closer to syntactic representation than to semantic representation. In comparison, knowledge-based methods compute the similarity between two sentences based on the semantic information collected from knowledge bases. However, this semantic information is applied in a way that, for any two sentences, the comparison of their nearest meanings is taken into consideration instead of the comparison of their actual meanings. More importantly, the nearest meaning does not always represent the actual meaning. In this paper, a solution is proposed that seeks to address these two issues. Firstly, the most outstanding existing corpus-based sentence similarity measure is transformed into a knowledge-based measure. Then, its underlying concept, which is the comparison of the nearest meanings, is replaced by another underlying concept, the comparison of the actual meanings.

The rest of this paper is organized into five sections. Section 2 presents an overview of the related works. Section 3 details the problem of the existing method and the improvement of the proposed method. Section 4 describes the experimental design. In Section 5, the experimental results are discussed. Finally, the implications and contributions are addressed in Section 6.

Coling 2008: Poster Volume, pages 418–426,
Beijing, August 2010

2 Related Work

In general, related works can be categorized into corpus-based, knowledge-based and hybrid-based methods. Islam and Inkpen (2008) proposed a corpus-based sentence similarity measure as a function of string similarity, word similarity and common word order similarity (CWO). They claimed that a corpus-based measure has the advantage of large coverage when compared to a knowledge-based measure. However, the judgment of similarity is situational and time dependent (Feng et al., 2008). This suggests that the statistical information collected from the past corpus may not be relevant to sentences present in the current corpus. Apart from that, the role of string similarity is to identify any misspelled word. A malfunction may occur whenever string similarity deals with any error-free sentences because the purpose for its existence is no longer valid.

For knowledge-based methods, Li et al. (2009) adopted an existing word similarity measure to deal with the similarities of verbs and nouns while the similarities of adjectives and adverbs were measured only based on simple word overlaps. However, Achananuparp et al. (2008) previously showed that the word overlap-based method performed badly in measuring text similarity. Liu et al. (2007) integrated the Dynamic Time Warping (DTW) technique into the similarity measure to identify the distance between words. The main drawback of DTW is that the computational cost and time will increase proportionately with the sentence's length. Wee and Hassan (2008) proposed a method that takes into account the directionality of similarity in which the similarity of any two words is treated as asymmetric. The asymmetric issue between a pair of words was resolved by considering both the similarity of the first word to the second word, and vice versa.

Corley and Mihalcea (2005) proposed a hybrid method by combining six existing knowledge-based methods. Mihalcea et al. (2006) further combined those six knowledge-based methods with two corpus-based methods and claimed that they usually achieved better performance in terms of precision and recall respectively. However, those methods were only combined by using simple average calculation.

Perhaps the most closely related work is a recently proposed query extension technique. Perez-Agüera and Zaragoza (2008) made use of WSD information to map the original query words and the expansion words to WordNet senses. However, without the presence of or considering the surrounding words, the meaning of the expansion words alone tend to be represented by their most general meanings instead of the disambiguated meanings, which results in the possibility of WSD information not being useful for word expansions. In contrast to their work, which is more suitable to be applied on word-to-word similarity task, the method proposed in this paper is more suitable for application on sentence-to-sentence similarity tasks.

Overall, the above-mentioned related works compute similarity based either on statistical information or on a comparison of the nearest meanings in terms of words. None of them compute sentence similarity based on the comparison of actual meanings. Our proposed method, which is a solution to this issue, will be explained in detail in the next section.

3 Sentence Similarity

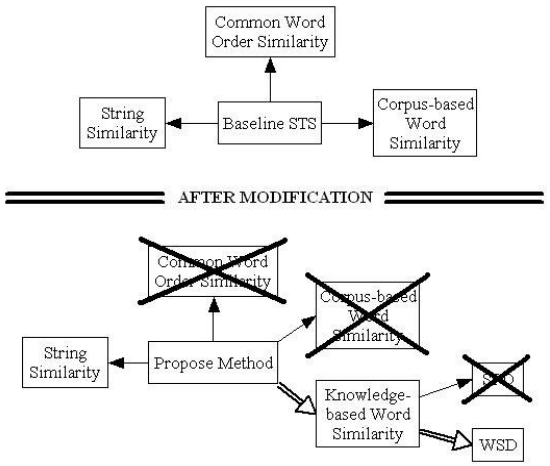


Figure 1. The proposed method

Our proposed method shown in Figure 1, is the outcome of some modifications on an existing method, which is also the most outstanding method, the Semantic Text Similarity (STS) model (Islam and Inkpen, 2008). First of all, CWO is removed from STS as the previous works (Islam and Inkpen, 2007; Islam and Inkpen, 2008) have shown that the presence of CWO has no influence on the outcome. Then,

the corpus-based word similarity function of STS is replaced by an existing knowledge-based word similarity measure called YP (Yang and Powers, 2005). Finally, the underlying concept of YP is modified by the integration of WSD and is based on the assumption that any disambiguated sense of a word represents its actual meaning. Thus, the proposed method is also called WSD-STS.

3.1 String similarity measure

The string similarity between two words is measured by using the following equations:

$$v_1 = NLCS(w_i^a, w_j^b) = \frac{l(LCS(w_i^a, w_j^b))^2}{l(w_i) \times l(w_j)} \quad (1)$$

$$v_2 = NMCLCS_1(w_i^a, w_j^b) = \frac{l(MCLCS_1(w_i^a, w_j^b))^2}{l(w_i) \times l(w_j)} \quad (2)$$

$$v_3 = NMCLCS_n(w_i^a, w_j^b) = \frac{l(MCLCS_n(w_i^a, w_j^b))^2}{l(w_i) \times l(w_j)} \quad (3)$$

$$Sim_{string}(X,Y) = 0.33v_1 + 0.33v_2 + 0.33v_3 \quad (4)$$

where $l(x)$ represents the length of x; a and b represent the lengths of sentences X and Y respectively after removing stop words; w_i represents the i-th word in sequence a; w_j represents the j-th word in sequence b; and $Sim_{string}(X,Y)$ represents the overall string similarity. The underlying concept of string similarity is based on character matching. *NLCS* represents the normalized version of the traditional longest common subsequence (LCS) technique in which the lengths of the two words are taken into consideration. $MCLCS_1$ represents the modified version of the traditional LCS in which the string matching must start from the first character while $MCLCS_n$ represents the modified version of the traditional LCS in which the string matching may start from any character. $NMCLCS_1$ and $NMCLCS_n$ represent the normalized versions of $MCLCS_1$ and $MCLCS_n$ respectively. More detailed information regarding string similarity measure can be found in the original paper (Islam and Inkpen, 2008).

3.2 Adopted word similarity measure

Yang and Powers (2005) proposed YP based on the assumptions that every single path in the hierarchical structure of WordNet 1) is identical; and 2) represents the shortest distance between any two connected words. The similarity between two words in sequence a and sequence b can be represented by the following equation:

$$Sim_{word}(w_i^a, w_j^b) = \begin{cases} \alpha_t \prod_{i=1}^{l-1} \beta_{t_i}, & l < \gamma \\ 0, & l \geq \gamma \end{cases} \quad (5)$$

where $0 \leq Sim_{word}(w_i^a, w_j^b) \leq 1$; d is the depth of LCS; l is the length of path between disambiguated w_i^a and w_j^b; t represents the type of path (hypernyms/hyponym, synonym or holonym/meronym) which connects them; α_t represents their path type factor; β_t represents their path distance factor; and γ represents an arbitrary threshold on the distance introduced for efficiency, representing human cognitive limitations. The values of α_t, β_t and γ have already been empirically tuned as 0.9, 0.85 and 12 respectively. More detailed information regarding YP can be found in the original paper (Yang and Powers, 2005).

In order to adapt a different underlying concept, which is the comparison of actual meanings, l has to be redefined as the path distance between disambiguated words, w_i^a and w_j^b. Since YP only differs from the modified version of YP (MYP) in terms of the definition of l, MYP can also be represented by equation (5).

3.3 The proposed measure

The gap

Generally, all the related works in Section 2 can be abstracted as a function of word similarity. This reflects the importance of a word similarity measure in measuring sentence similarity. However, measuring sentence similarity is always a more complicated task than measuring word similarity. The reason is that while a word similarity measure only involves a single pair of words, a sentence similarity measure has to deal with multiple pairs of words. In addition, due to the presence of the surrounding words in a sentence, the possible meaning of a word is always being restricted (Kolte and Bhirud, 2008). Thus, without some modifications, the traditional word similarity measures, which are based on the concept of a comparison of the nearest meanings, are inapplicable in the context of sentence similarity measures.

The importance of WSD in reducing the gap

Before performing the comparison of actual meanings, WSD has to be integrated so that the most suitable sense can be assigned to any polysemous word. The importance of WSD can be investigated by using a simple example. Consider a pair of sentences, collected from WordNet 2.1, which use two words, "*dog*" and "*cat*":

X: The dog barked all night.

Y: What a cat she is!

Based on the definition in WordNet 2.1, the word "*dog*" in *X* is annotated as the first sense which means "*a member of the genus Canis* (*probably descended from the common wolf*) *that has been domesticated by man since prehistoric times*". Meanwhile, the word "*cat*" in *Y* is annotated as the third sense with the definition of "*a spiteful woman's gossip*". The path distance between "*cat*" and "*dog*" based on their actual senses is equal to 7. However, their shortest path distance (SPD), which is based on their nearest senses, is equal to 4. SPD is the least number of edges connecting two words in the hierarchical structure of WordNet. In other words, "*cat*" and "*dog*" in *X* and *Y* respectively, are not as similar as the one measured by using SPD. The presence of the additional path distances is significant as it is almost double the actual path distance between "*cat*" and "*dog*".

WSD-STS

The adopted sentence similarity measure, STS, can be represented by the following equations:

$$Sim_{semantic}(X,Y) = \frac{(\delta + \sum_{i=1}^{c} \tau_i) \times (a+b)}{2ab} \quad (6)$$

$$SIM(X,Y) = \frac{Sim_{smeantic}(X,Y) + Sim_{string}(X,Y)}{2} \quad (7)$$

where for equation (6): δ represents the number of overlapped words between the words in sequence *a* and sequence *b*; *c* represents the number of semantically matched words between the words in sequence *a* and sequence *b*, in which *c* = *a* if *a* < *b* or *c* = *b* if *b* < *a*, τ_i represents the highest matching similarity score of *i*-th word in the shorter sequence with respect to one of the words in the longer sequence; and $\Sigma\tau$ represents the sum of the highest matching similarity score between the words in sequence *a* and sequence *b*.

For STS, the similarity between two words is measured by using a corpus-based measure. For WSD-STS, this corpus-based measure is replaced by MYP. Finally, the overall sentence similarity is represented by equation (7).

4 Experimental Design

4.1 Data set

Li et al., (2006) constructed a data set which consists of 65 pairs of human-rated sentences by applying the similar experimental design for creating the standard data set for the word similarity task (Rubenstein and Goodenough, 1965). These 65 sentence pairs were the definitions collected from the Collin Cobuild Dictionary. Out of these, 30 sentence pairs with rated similarity scores that ranged from 0.01 to 0.96 were selected as test data set. The corresponding 30 word pairs for these 30 sentence pairs are shown in the second column of Table 1. A further set of 66 sentence pairs is still under development and it will be combined with the existing data set in the future (O'Shea et al., 2008b).

4.2 Procedure

Firstly, Stanford parser[1] is used to parse each sentence and to tag each word with a part of speech (POS). Secondly, Structural Semantic Interconnections[2] (SSI), which is an online WSD system, is used to disambiguate and to assign a sense for each word in the 30 sentences based on the assigned POS. SSI is applied based on the assumption that it is able to perform WSD correctly. The main reason for choosing SSI to perform WSD is its promising results reported in a study by Navigli and Verladi (2006). Thirdly, all the stop words which exist in these 30 pairs of sentences are removed. It is important to note that the 100 most frequent words collected from British National Corpus (BNC) were applied as the stop words list on the baseline, STS. However, due to the limited accessibility to BNC, a different stop words list[3], which is available online, is applied in this paper.

[1] http://nlp.stanford.edu/software/lex-parser.shtml

[2] http://lcl.uniroma1.it/ssi

[3] http://www.translatum.gr/forum/index.php?topic=2476.0

No.	The Corresponding Word Pairs of the Test Data Set	Human Similarity (Mean)	(Li et al., 2006)	(Feng et al., 2008)	(O'Shea et al., 2008a)	(Liu et al., 2007)	(Islam and Inkpen, 2008) STS	Experimental Conditions		
								OLP-STS	WSD-STS	SPD-STS
1	Cord-Smile	0.01	0.33	0.15	0.51	0.03	0.06	0.000	0.026	0.089
5	Autograph-Shore	0.01	0.29	0.28	0.53	0.00	0.11	0.000	0.061	0.061
9	Asylum-Fruit	0.01	0.21	0.31	0.51	0.00	0.07	0.000	0.035	0.043
13	Boy-Rooster	0.11	0.53	0.40	0.54	0.12	0.16	0.000	0.088	0.137
17	Coast-Forest	0.13	0.36	0.13	0.58	0.02	0.26	0.208	0.266	0.284
21	Boy-Sage	0.04	0.51	0.36	0.53	0.14	0.16	0.000	0.113	0.140
25	Forest-Graveyard	0.07	0.55	0.23	0.60	0.18	0.33	0.146	0.199	0.201
29	Bird-Woodland	0.01	0.33	0.16	0.51	0.01	0.12	0.000	0.054	0.059
33	Hill-Woodland	0.15	0.59	0.21	0.81	0.47	0.29	0.208	0.252	0.246
37	Magician-Oracle	0.13	0.44	0.31	0.58	0.05	0.20	0.000	0.081	0.092
41	Oracle-Sage	0.28	0.43	0.20	0.58	0.16	0.09	0.000	0.025	0.045
47	Furnace-Stove	0.35	0.72	0.29	0.72	0.06	0.30	0.000	0.094	0.136
48	Magician-Wizard	0.36	0.65	0.36	0.62	0.22	0.34	0.143	0.229	0.294
49	Hill-Mound	0.29	0.74	0.18	0.54	0.45	0.15	0.000	0.149	0.130
50	Cord-String	0.47	0.68	0.50	0.68	0.16	0.49	0.222	0.246	0.340
51	Glass-Tumbler	0.14	0.65	0.27	0.73	0.16	0.28	0.156	0.188	0.246
52	Grin-Smile	0.49	0.49	0.43	0.70	0.18	0.32	0.250	0.273	0.330
53	Serf-Slave	0.48	0.39	0.49	0.83	0.18	0.44	0.436	0.472	0.458
54	Journey-Voyage	0.36	0.52	0.32	0.61	0.19	0.41	0.225	0.260	0.260
55	Autograph-Signature	0.41	0.55	0.30	0.70	0.33	0.19	0.258	0.315	0.332
56	Coast-Shore	0.59	0.76	0.31	0.78	0.46	0.47	0.375	0.489	0.489
57	Forest-Woodland	0.63	0.70	0.25	0.75	0.39	0.26	0.208	0.264	0.342
58	Implement-Tool	0.59	0.75	0.25	0.83	0.34	0.51	0.511	0.560	0.560
59	Cock-Rooster	0.86	1.00	0.92	0.99	0.85	0.94	0.750	0.866	0.866
60	Boy-Lad	0.58	0.66	0.61	0.83	0.69	0.60	0.250	0.554	0.570
61	Cushion-Pillow	0.52	0.66	0.29	0.63	0.45	0.29	0.139	0.182	0.255
62	Cemetery-Graveyard	0.77	0.73	0.91	0.74	0.65	0.51	0.402	0.487	0.587
63	Automobile-Car	0.56	0.64	0.45	0.87	0.38	0.52	0.321	0.378	0.378
64	Midday-Noon	0.96	1.00	0.99	1.00	1.00	0.93	0.750	0.862	0.862
65	Gem-Jewel	0.65	0.83	0.64	0.86	0.60	0.65	0.450	0.566	0.566

Table 1. Data Set Results

Finally, the remaining content words are lemmatized by using Natural Language Toolkit[4] (NLTK). Nevertheless, those words which can be found in WordNet and which have different definitions from their lemmatized form will be excluded from lemmatization. For instance,

Cooking[NN] can be a great art.

The word in the bracket represents the tagged POS for its corresponding word. Since based on the definitions provided by WordNet, "*cooking*", which is tagged as a noun, has a different meaning from its lemmatized form "*cook*", which is also tagged as a noun. Therefore, "*cooking*" is excluded from lemmatization.

4.3 Experimental conditions

Sentence similarity is measured under the following three conditions:

- **OLP-STS**: A modified version of the baseline, STS (Islam and Inkpen, 2008), in which it only relies on the presence of overlapped words. This means that the component $\sum_{i=1}^{c}\tau_i$, which represents the word similarity, is removed from equation (6).
- **SPD-STS**: The corpus-based word similarity measure of the baseline, STS, which is represented by $\sum_{i=1}^{c}\tau_i$ in equation (6), is replaced by a knowledge-based word similarity measure, YP.
- **WSD-STS**: A modified version of SPD-STS in which the knowledge-based measure, YP, is replaced by MYP.

As mentioned in Section 4.2, different stop words lists were applied between the baseline and the proposed methods under different ex-

[4] http://www.nltk.org/

perimental conditions in this paper. Since this issue may be questioned due to the unfair comparison, the performance of WSD-STS is evaluated on top of a number of different stop words lists which are available online in order to investigate any influence which may be caused by stop words list.

5 Results and Discussion

Table 1 presents the similarity scores obtained from the mean of human ratings, the benchmarks, and different experimental conditions of the proposed methods. Figure 2 presents the corresponding Pearson correlation coefficients of various measures as listed in Table 1.

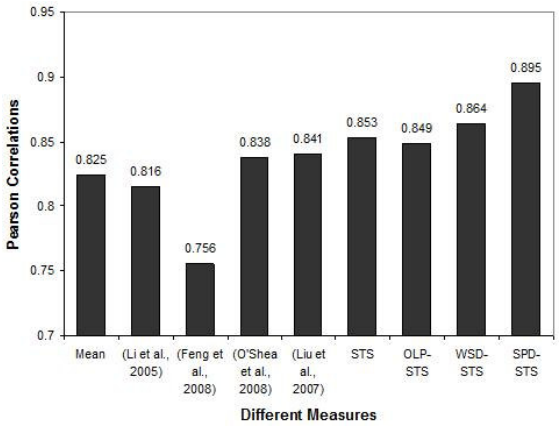


Figure 2. Pearson Correlation Coefficient

Figure 2 shows that STS appears to be the most outstanding measure among the existing works with a correlation coefficient of 0.853. However, Figure 2 also shows that both the proposed methods in this paper, WSD-STS and SPD-STS, outperform STS. This result indicates that knowledge-based method tends to perform better than a corpus-based method. The reason is that a knowledge base is much closer to human representation of knowledge (WordNet is the knowledge base applied in this paper) than a corpus. A corpus only reflects the usage of languages and words while WordNet is a model of human knowledge constructed by many expert lexicographers (Li et al., 2006). In other words, a corpus is more likely to provide unprocessed raw data while a knowledge base tends to provide ready-to-use information.

The results of the performance of the two proposed methods are as expected. SPD-STS achieved a bigger but statistically insignificant improvement while WSD-STS achieved a smaller but statistically significant improvement at 0.01 levels. The significance of a correlation is calculated by using an online calculator, *VassarStats*[5]. The reason for the variance in the outcomes between SPD-STS and WSD-STS is obvious; it is the difference in terms of their underlying concepts. In other words, sentence similarity computation, which is based on a comparison of the nearest meanings, results in insignificant improvement while sentence similarity computation, which is based on a comparison of actual meanings, achieves statistically significant improvement. These explanations indicate that WSD is essential in confirming the validity of the task of measuring sentence similarity.

Figure 2 also reveals that a relatively low correlation is achieved by OLP-STS. This is not at all surprising since Achananuparp et al. (2008) has already demonstrated that the overlapped word-based method tends to perform badly in measuring sentence similarity. However, it is interesting to find that the difference in performance between STS and OLP-STS is very small. This indirectly suggests that the presence of the string similarity measure and the corpus-based word similarity measure has only a slight improvement on the performance of OLP-STS.

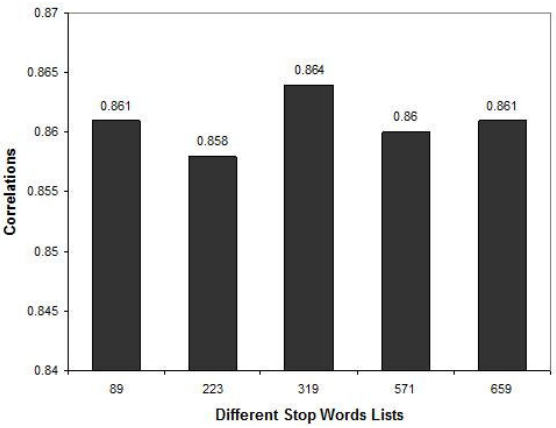


Figure 3. The performance of the WSD-SPD versus different stop words lists

Next, in order to address the issue of unfair comparison due to the usage of different stop words lists, the performance of WSD-SPD has been evaluated on top of a number of different

[5] http://faculty.vassar.edu/lowry/rdiff.html?

stop words lists. A total of five stop words lists with different lengths (89[6], 223[7], 319, 571[8] and 659[9]) of stop words were applied. The performances of WSD-SPD with respect to these stop words lists are portrayed in Figure 3. They are found to be in a comparable condition. This result connotes that the influence caused by the usage of different stop words lists is small and can be ignored. Hence, the unfair comparison between our proposed method and the baseline should not be treated as an issue for the benchmarking purpose of this paper.

On the other hand, although an assumption is made that SSI performs WSD correctly, we noticed that not all the words were disambiguated confidently. The confident scores which were assigned to the disambiguated words by SSI range between 30% and 100%. These confident scores reflect the confidence of SSI in performing WSD. Thus, it is possible that some of those words which were assigned with low confident scores were disambiguated incorrectly. Consequently, the final sentence similarity score is likely to be affected negatively. In order to reduce the negative effect which may be caused by incorrect WSD, any words pair which is not confidently disambiguated is assigned the similarity score based on the concept of comparing the nearest meanings instead of comparing the actual meanings. In other words, WSD-STS and SPD-STS are combined and results in WSD-SPD. The performance of WSD-SPD across a range of confident scores is essential in revealing the impact of WSD and SPD on the task of measuring sentence similarity.

Figure 4 outlines the performance achieved by WSD-SPD across different confident scores assigned by SSI. The confident score of at least 0.7 is identified as the threshold in which SSI optimizes its performance. The performance of WSD-SPD is found to be statistically insignificant for those confident scores above the threshold. The explanation for this phenomenon can be found in Figure 5. Figure 5 illustrates the percentage of the composition between WSD and SPD in WSD-SPD. It is obvious that once the portion of WSD exceeds the portion of SPD, the performance of WSD-SPD is found to be statistically insignificant. This finding suggests that SPD, which reflects the application of the concept of nearest meaning comparison, is likely to decrease the validity of sentence similarity measurement while WSD, which reflects the application of the concept of actual meaning comparison, is essential in confirming the validity of sentence similarity measurement.

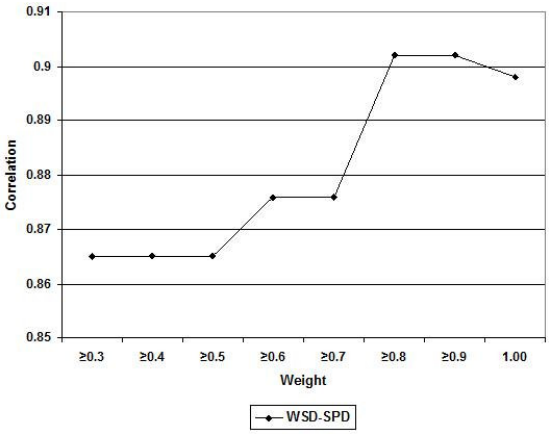


Figure 4. The performance of WSD-SPD versus confident scores

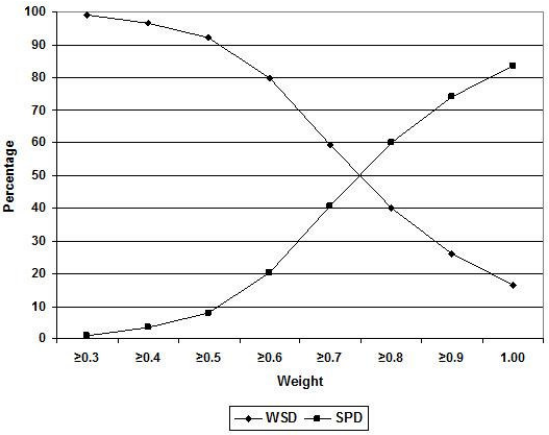


Figure 5. The percentage of WSD/SPD versus confident score

The trend of the performance of string similarity measure and word similarity measure with respect to different weight assignments is delineated in Figure 6. The lowest correlation of 0.856 is obtained when only the string similarity function is considered while the word similarity

[6] http://msdn.microsoft.com/en-us/library/bb164590.aspx
[7] http://snowball.tartarus.org/algorithms/english/stop.txt
[8] http://truereader.com/manuals/onix/stopwords2.html
[9] http://www.link-assistant.com/seo-stop-words.html

function is excluded. A better performance is achieved by taking the two measures into consideration where more weight is given to the measure of word similarity. This trend intimates that the string similarity measure offers a smaller contribution in measuring sentence similarity than word similarity measure. In contrast to a word similarity measure, a string similarity measure is purposely proposed to address the issue of misspelled words. Since the data set applied in this experiment does not contain any misspelled words, it is obvious that a string similarity measure performs badly. In addition, the underlying concept of string similarity is questionable. Does it make sense to determine the similarity of two words based on the matching between their characters or the matching of the sequence of characters? Consider four pairs of words: "*play*" versus "*pray*", "*plant*" versus "*plane*", "*plane*" versus "*plan*" and "*stationary*" versus "*stationery*". These word pairs are highly similar in terms of characters but they are semantically dissimilar or unrelated.

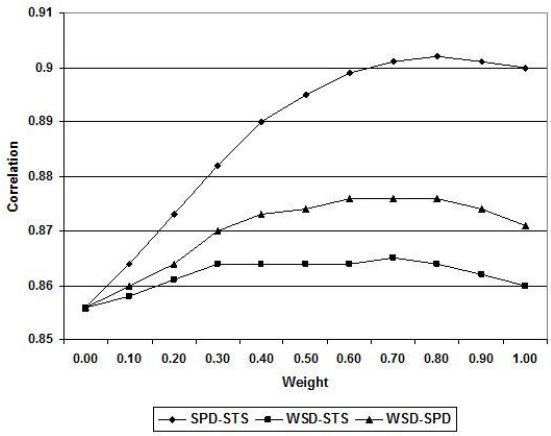


Figure 6. The performance of the different measures versus the weight between string similarity and word similarity

Figure 6 also depicts that the combination of word similarity measure (70%) and string similarity measure (30%) performs better than the measure which is solely based on word similarity function. It is obvious that the difference is caused by the presence of string similarity measure. The combination assigns similarity scores to all word pairs while the word similarity measure only assigns similarity scores to those word pairs which fulfill two requirements: 1) any two words which share an identical POS, and 2) any two words which must either be a pair of nouns or a pair of verbs. In fact, adjectives and adverbs do contribute to representing the meaning of a sentence although their contribution is relatively smaller than the contribution of nouns and verbs (Liu et al., 2007; Li et al., 2009). Therefore, by ignoring the presence of adjectives and adverbs, the performance will definitely be affected negatively.

6 Conclusion

This paper has presented a knowledge-based method which measures the similarity between two sentences based on their actual meaning comparison. The result shows that the proposed method, which is a knowledge-based measure, performs better than the baseline, which is a corpus-based measure. The improvement obtained is statistically significant at 0.025 levels. This result also shows that the validity of the output of measuring the similarity of two sentences can be improved by comparing their actual meanings instead of their nearest meanings. These are achieved by transforming the baseline into a knowledge-based method and then by integrating WSD into the adopted knowledge-based measure.

Although the proposed method significantly improves the quality of measuring sentence similarity, it has a limitation. The proposed method only measures the similarity between two words with an identical part of speech (POS) and these two words must either be a pair of nouns or a pair of verbs. By ignoring the importance of adjectives and adverbs, and the relationship between any two words with different POS, a slight decline is observed in the obtained result. In future research, these two issues will be addressed by taking into account the relatedness between two words instead of only considering their similarity.

References

Achananuparp, Palakorn, Xiao-Hua Hu, and Xiao-Jiong Shen. 2008. The Evaluation of Sentence Similarity Measures. *In Proceedings of the 10th International Conference on Data Warehousing*

and Knowledge Discovery (DaWak), pages 305-316, Turin, Italy.

Corley, Courtney, and Rada Mihalcea. 2005. Measuring the Semantic Similarity of Texts. *In Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment*, pages 48-55, Ann Arbor.

Feng, Jin, Yi-Ming Zhou, and Trevor Martin. 2008. Sentence Similarity based on Relevance. *In Proceedings of IPMU*, pages 832-839.

Islam, Aminul, and Diana Inkpen. 2007. Semantic Similarity of Short Texts. *In Proceedings of RANLP*, pages 291-297.

Islam, Aminul, and Diana Inkpen. 2008. Semantic Text Similarity Using Corpus-Based Word Similarity and String Similarity. *ACM Transactions on Knowledge Discovery from Data*, 2(2):10.

Kauchak, David, and Regina Barzilay. 2006. Paraphrasing for Automatic Evaluation. *In Proceedings of HLT-NAACL*, pages 455-462, New York.

Kolte, Sopan Govind, and Sunil G. Bhirud. 2008. Word Sense Disambiguation using WordNet Domains. *In The First International Conference on Emerging Trends in Engineering and Technology*, pages 1187-1191.

Lapata, Mirella, and Regina Barzilay. 2005. Automatic Evaluation of Text Coherence: Models and Representations. *In Proceedings of the 19th International Joint Conference on Artificial Intelligence*.

Li, Lin, Xia Hu, Bi-Yun Hu, Jun Wang, and Yi-Ming Zhou. 2009. Measuring Sentence Similarity from Different Aspects. *In Proceedings of the Eighth International Conference on Machine Learning and Cybernetics*, pages 2244-2249.

Li, Yu-Hua, David McLean, Zuhair A. Bandar, James D.O'Shea, and Keeley Crockett. 2006. Sentence Similarity Based on Semantic Nets and Corpus Statistics. *IEEE Transactions on Knowledge and Data Engineering*, 18(8):1138-50.

Liu, Xiao-Ying, Yi-Ming Zhou, and Ruo-Shi Zheng. 2007. Sentence Similarity based on Dynamic Time Warping. *In The International Conference on Semantic Computing*, pages 250-256.

Mihalcea, Rada, Courtney Corley, and Carlo Strapparava. 2006. Corpus-based and Knowledge-based Measures of Text Semantic Similarity. *In Proceedings of the American Association for Artificial Intelligence*.

Navigli, Roberto, and Paola Velardi. 2005. Structural Semantic Interconnections: A Knowledge-Based Approach to Word Sense Disambiguation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27(7):1075-86.

O'Shea, James, Zuhair Bandar, Keeley Crockett, and David McLean. 2008a. A Comparative Study of Two Short Text Semantic Similarity Measures. *In KES-AMSTA, LNAI: Springer Berlin / Heidelberg*.

O'Shea, James, Zuhair Bandar, Keeley Crockett, and David McLean. 2008b. Pilot Short Text Semantic Similarity Benchmark Data Set: Full Listing and Description.

Perez-Aguera, Jose R., and Hugo Zaragoza. 2008. UCM-Y!R at Clef 2008 Robust and WSD Tasks. *In Working Notes for CLEF Workshop*.

Rubenstein, Herbert, and John B. Goodenough. 1965. Contextual Correlates of Synonymy. *Communications of the ACM*, pages 627-633.

Wang, Yao-Feng, Yue-Jie Zhang, Zhi-Ting Xu, and Tao Zhang. 2006. Research on Dual Pattern of Unsupervised and Supervised Word Sense Disambiguation. *In Proceedings of the Fifth International Conference on Machine Learning and Cybernetics*, pages 2665-2669.

Wee, Leong Chee, and Samer Hassan. 2008. Exploiting Wikipedia for Directional Inferential Text Similarity. *In Proceedings of Fifth International Conference on Information Technology: New Generations*, pages 686-691.

Yang, Cha, and Jun Wen. 2007. Text Categorization Based on Similarity Approach. *In Proceedings of International Conference on Intelligence Systems and Knowledge Engineering (ISKE)*.

Yang, Dong-Qiang, and David M.W. Powers. 2005. Measuring Semantic Similarity in the Taxonomy of WordNet. *In Proceedings of the 28th Australasian Computer Science Conference*, pages 315-332, Australia.

Zhou, Liang, Chin-Yew Lin, Dragos Stefan Munteanu, and Eduard Hovy. 2006. ParaEval: Using Paraphrases to Evaluate Summaries Automatically. *In Proceedings of Human Language Technology Conference of the North American Chapter of the ACL*, pages 447-454, New York.

Towards Automated Related Work Summarization

Cong Duy Vu Hoang and **Min-Yen Kan**
Department of Computer Science
School of Computing
National University of Singapore
{hcdvu,kanmy}@comp.nus.edu.sg

Abstract

We introduce the novel problem of automatic related work summarization. Given multiple articles (*e.g.*, conference/journal papers) as input, a related work summarization system creates a topic-biased summary of related work specific to the target paper. Our prototype **Re**lated **Wo**rk **S**ummarization system, **ReWoS**, takes in set of keywords arranged in a hierarchical fashion that describes a target paper's topics, to drive the creation of an extractive summary using two different strategies for locating appropriate sentences for general topics as well as detailed ones. Our initial results show an improvement over generic multi-document summarization baselines in a human evaluation.

1 Introduction

In many fields, a scholar needs to show an understanding of the context of his problem and relate his work to prior community knowledge. A related work section is often the vehicle for this purpose; it contextualizes the scholar's contributions and helps readers understand the critical aspects of the previous works that current work addresses. Creating such a summary requires the author to position his own work within the contextual research to showcase the advantages of his method.

We envision an NLP application that assists in creating a related work summary. We propose this *related work summarization* task as a challenge to the automatic summarization community. In its full form, it is a topic-biased, multi-document summarization problem that takes as input a target scientific document for which a related work section needs to be drafted. The output goal is to create a related work section that finds the relevant related works and contextually describes them in relationship to the scientific document at hand.

We dissect the full challenge as bringing together work of disparate interests; 1) in finding relevant documents; 2) in identifying the salient aspects of these documents in relation to the current work worth summarizing; and 3) in generating the final topic-biased summary. While it is clear that current NLP technology does not let us build a complete solution for this task, we believe that tackling the individual components will help bring us towards an eventual solution.

In fact, existing works in the NLP and recommendation systems communities have already begun work that fits towards the completion of the first two tasks. Citation prediction (Nallapati et al., 2008) is a growing research area that has aimed both at predicting citation growth over time within a community and at individual paper citation patterns. This year, an automatic keyphrase extraction task from scientific articles was first fielded in SemEval-2, partially addressing Task 1[1]. Also, automatic survey generation (Mohammad et al., 2009) is becoming a growing field within the summarization community. However, to date, we have not yet seen any work that examines topic-biased summarization of multiple scientific articles. For these reasons, we focus on Task 3 here – *the creation of a related work section, given a structured input of the topics for summary.*. The remaining contributions of our paper

[1] http://semeval2.fbk.eu/semeval2.php

Coling 2008: Poster Volume, pages 427–435,
Beijing, August 2010

consists of work towards this goal:

• We conduct a study of the argumentative patterns used in related work sections, to describe the plausible summarization tactics for their creation in Section 3.

• We describe our approach to generate an extractive related work summary given an input topic hierarchy tree, using two separate strategies to differentiate between summarizing shallow internal nodes from deep detailed leaf nodes of the topic tree in Section 4.

2 Related Work

Fully automated related work summarization is significantly different from traditional summarization. While there are no existing studies on this specific problem, there are closely related endeavors. The iOPENER[2] project works towards automated creation of technical surveys, given a research topic (Mohammad et al., 2009). Standard generic multi-document summarization algorithms were applied to generate technical surveys. They showed that citation information was effective in the generation process. This was also validated earlier in (Nakov et al., 2004), which showed that the citing sentences in other papers can give a useful description of a target work.

Other studies focus mainly on single-document scientific article summarization. The pioneers of automated summarization (Luhn, 1958; Baxendale, 1958; Edmundson, 1969) had envisioned their approaches being used for the automatic creation of scientific summaries. They examined various features specific to scientific texts (*e.g.*, frequency-based, sentence position, or rhetorical clues features) which were proven effective for domain-specific summarization tasks.

Further, Mei and Zhai (2008) and Qazvinian and Radev (2008) utilized citation information in creating summaries for a single scientific article in computational linguistics domain. Also, Schwartz and Hearst (2006) also utilized the citation sentences to summarize the key concepts and entities in bioscience texts, and argued that citation sentences may contain informative contributions of a paper that complement its original abstract.

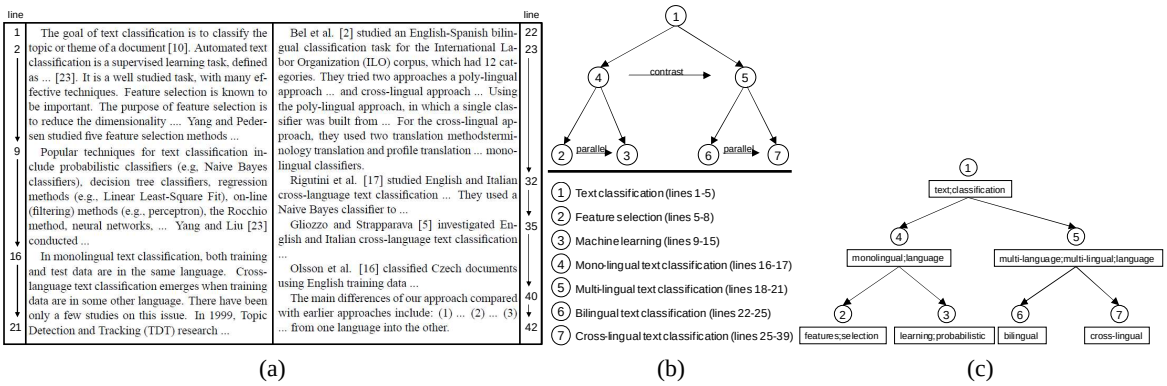
[2]http://clair.si.umich.edu/clair/iopener/

These works all center on the role of citations and their contexts in creating a summary, using citation information to rank content for extraction. However, they did not study the rhetorical structure of the intended summaries, targeting more on deriving useful content. For working along this vein, we turn to studies on the rhetorical structure of scientific articles. Perhaps the most relevant is work by (Teufel, 1999; Teufel and Moens, 2002) who defined and studied argumentative zoning of texts, especially ones in computational linguistics. While they studied the structure of an entire article, it is clear from their studies that a related work section would contain general background knowledge (BACKGROUND zone) as well as specific information credited to others (OTHER and BASIS zones). This vein of work has been followed by many, including Teufel et al. (2009; Angrosh et al. (2010).

3 Structure of Related Work Section

We first extend the work on rhetorical analysis, concentrating on related work sections. By studying examples in detail, we gain insight on how to approach related work summarization. We focus on a concrete related work example for illustration, an excerpt of which is shown in Figure 1a. Focusing on the argumentative progression of the text, we note the flow through different topics is hierarchical and can be represented as a topic tree as in Figure 1b.

This summary provides background knowledge for a paper on text classification, which is the root of the topic tree (node 1; lines 1–5). Two topics ("feature selection" and "machine learning") are then presented in parallel (nodes 2 & 3; lines 5–8 & 9–15), where specific details on relevant works are selected to describe two topics. These two topics are implicitly understood as subtopics of a more general topic, namely "mono-lingual text classification" (node 4; lines 16–17). The authors use the monolingual topic to contrast it with the subsequent subtopic "multi-lingual text classification" (node 5; lines 18–21). This topic is described by elaborating its details through two subtopics: "bilingual text classification" and "cross-lingual text classification" (nodes 6 & 7; lines 22–25 & 25–39) where again, various example works



(a) (b) (c)

Figure 1: a) A related work section extracted from (Wu and Oard, 2008); b) An associated topic hierarchy tree of a); c) An associated topic tree, annotated with key words/phrases.

are described and cited. The authors then conclude by contrasting their proposed approach with the introduced relevant approaches (lines 40–42).

This summary illustrates three important points. First, the topic tree is an essential input to the summarization process. The topic tree can be thought of as a high-level rhetorical structure for which a process then attaches content. While it is certainly non-trivial to build such a tree, modifications to hierarchical topic modeling (M. et al., 2004) or keyphrase extraction algorithms (Witten et al., 1999) we believe can be used to induce a suitable form. A resulting topic hierarchy from such a process would provide an associated set of key words or phrases that would describe the node, as shown in Figure 1c.

Second, while summaries can be structured in many ways, they can be viewed as moves along the topic hierarchy tree. In the example, nodes 2 and 3 are discussed before their parent, as the parent node (node 4) serves as a useful contrast to introduce its sibling (node 5). We find variants of depth-first traversal common, but breadth-first traversals of nodes with multiple descendants are more rare. They may be structured this way to ease the reader's burden on memory and attention. This is in line with other summary genres where information is ordered by high-level logical considerations that place macro level constraints (Barzilay et al., 2002).

Third, there is a clear distinction between sentences that describe a general topic and those that describe work in detail. Generic topics are often represented by background information, which is not tied to a particular prior work. These include definitions or descriptions of a topic's purpose. In contrast, detailed information forms the bulk of the summary and often describes key related work that is attributable to specific authors. Recently, Jaidka et al. (2010) also present the beginnings of a corpus study of related work sections, where they differentiate integrative and descriptive strategies in presenting discourse work. We see our differentiation between general and detailed topics as a natural parallel to their notion of integrative and descriptive strategies.

To introspect on these findings further, we created a related work data set (called **RWSData**[3]), which includes 20 articles from well-respected venues in NLP and IR, namely SIGIR, ACL, NAACL, EMNLP and COLING. We extracted the related work sections directly from those research articles as well as references the sections cited. References to books and Ph.D. theses were removed, as their verbosity would change the problem drastically (Mihalcea and Ceylan, 2007). Since we view each related work summary as a topic-biased summary originating from a topic hierarchy tree, annotation of such topical information for our data set is necessary. Each article's data consists of the reference related work summary, the collection of the input research articles

[3]To be made available at `http://wing.comp.nus.edu.sg/downloads/rwsdata`.

	SbL−RW	WbL−RW	No−RAs	SbL−RA	WbL−RA	TS	TD
average	17.9	522.4	10.9	2386.0	51739.6	3.3	1.8
stdev	7.9	216.5	5.6	1306.7	26682.3	1.7	0.6
min	6	179	2	348	8580	1	1
max	40	922	26	5549	112267	7	3

Table 1: **The demographics of RWSData**. No, RW, RA, SbL, WbL, TS, and TD are labeled as (N)umber (o)f, (R)elated (W)orks, (R)eferenced (A)rticles, (S)entence-(b)ased (L)ength of, (W)ord-(b)ased (L)ength of, (T)ree (S)ize, and (T)ree (D)epth, respectively.

that were referenced and a manually-constructed topic descriptions in a hierarchical fashion (topic tree). More details on the demographics of RWSData are shown in Table 1. RWSData summaries average 17.9 sentences, 522 words in length, citing an average of 10.9 articles. While hierarchical, the topic trees are simple, averaging 3.3 topic nodes in size and average depth of 1.8. Their simplicity furthers our claim that automated methods would be able to create such trees.

4 ReWoS: Paired General and Specific Summarization

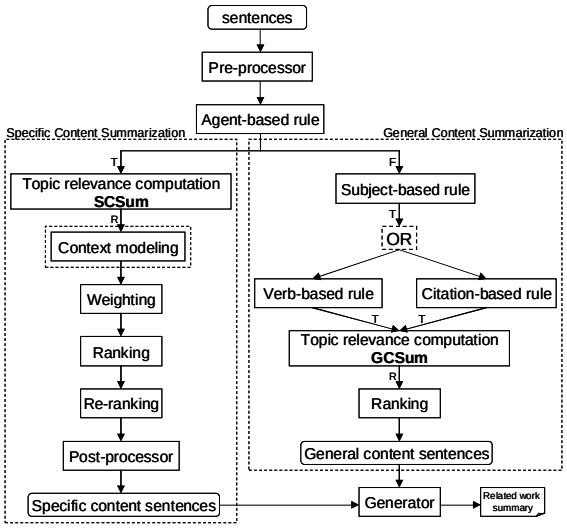


Figure 2: **The ReWoS architecture**. Decision edges labeled as **T**rue, **F**alse and **R**elevant.

Inspired by the above observations, we propose a novel strategy for related work summarization with respect to a given topic tree. Note that while the construction of the topic tree is central to the process, we consider this outside the scope of the current work (see §1); our investigation focuses on how such input could be utilized to construct a reasonable topic-biased related work summary.

We posit that sentences within a related work section come about by means of two separate processes – a process that gives general background information and another that describes specific author contributions. A key realization in our work is that these processes are easily mapped to the topic tree topologically: general content is described in tree-internal nodes, whereas leaf nodes contribute detailed specifics. In our approach, these two processes are independent, and combined to construct the final summary.

We have implemented our idea in **ReWoS** (**Re**lated **Wo**rk **S**ummarizer), whose general architecture is shown in Figure 2. ReWoS is a largely heuristic system, featuring both a **G**eneral **C**ontent **Sum**marization (**GCSum**) and a **S**pecific **C**ontent **Sum**marization (**SCSum**) modules, prefixed by preprocessing. A natural language template **generation** system fills out the end of the summary.

ReWoS first applies a set of preprocessing steps (shown in the top of Figure 2). Input sentences (*i.e.*, the set of sentences from each related/cited article) first removes sentences that are too short (< 7 tokens) or too long (> 80 tokens), ones that use future tense (possibly future work), and example and navigation sentences. This last category is filtered out by checking for the presence of a cue phrase among a lexical pattern database: *e.g.*, "in the section", "figure x shows", "for instance". Lowercasing and stemming are also performed.

We then direct sentences to either GCSum or SCSum based on whether it describes the author's own work or not, similar in spirit and execution to (Teufel et al., 2009). If sentence contains indicative pronouns or cue phrases (*e.g.*, "we", "this ap-

proach"), the sentence is deemed to describe own work and is directed to SCSum; otherwise the sentence is directed to the GCSum workflow.

4.1 (G)eneral (C)ontent (Sum)marization

GCSum extracts sentences containing useful background information on the topics of the internal node in focus. Since general content sentences do not specifically describe work done by the authors, we only take sentences that do not have the author-as-agent as input for GCSum.

We divide such general content sentences into two groups: indicative and informative. Informative sentences give detail on a specific aspect of the problem. They often give definitions, purpose or application of the topic (*"Text classification is a task that assigns a certain number of predefined labels for a given text."*). In contrast, indicative sentences are simpler, inserted to make the topic transition explicit and rhetorically sound (*"Many previous studies have studied monolingual text classification."*).

Indicative sentences can be easily generated by templates, as the primary information that is transmitted is the identity of the topic itself. Informative sentences, on the other hand, are better extracted from the source articles themselves, requiring a specific strategy. As informative sentences contain more content, our strategy with GCSum is to attempt to locate informative sentences to describe the internal nodes, failing which GCSum falls back to using predefined templates to generate an indicative placeholder.

To implement GCSum's informative extractor, we use a set of heuristics in a decision cascade to first filter inappropriate sentences (as shown on the RHS of Figure 2). Remaining candidates (if any) are then ranked by relevance and the top n are selected for the summary.

The heuristic cascade's purpose is to ensure sentences fit the syntactic structure of commonly-observed informative sentences. A useful sentence should discuss the topic directly, so GCSum first checks the subject of each candidate sentence, filtering sentences whose subject do not contain at least one topic keyword. We observed that background sentences often feature specific verbs or citations. GCSum thus also checks whether stock verb phrases (*i.e.*, "based on", "make use of" and 23 other patterns) are used as the main verb. Otherwise, GCSum checks for the presence of at least one citation – general sentences may list a set of citations as examples. If both the cue verb and citation checks fail, the sentence is dropped.

GCSum's topic relevance computation ranks remaining sentences based on keyword content. We state that the topic of an internal node is affected by its surrounding nodes – ancestor, descendants and siblings. Based on this idea, the score of a sentence is computed in a discriminative way using the following linear combination:

$$score_S \rightarrow score_S^{QA} + score_S^{Q} - score_S^{QR} \quad (1)$$

where $score_S$ is the final relevance score, and $score_S^{QA}$, $score_S^{Q}$, and $score_S^{QR}$ are the component scores of the sentence S with respect to the ancestor, current or other remaining nodes. We give positive credit to a sentence that contains keywords from an ancestor node, but penalize sentences with keywords from other topics (as such sentences would be better descriptors for those other topics). Component relevance scores are calculated using Term Frequency $\times$ Inverse Sentence Frequency (TF$\times$ISF) (Otterbacher et al., 2005):

$$\begin{aligned} score_S^{Q} &= \frac{rel(S,Q)}{\sum_{Q'} rel(S,Q')} \quad (2) \\ &= \frac{\sum_{w \in Q} \log(tf_w^S + 1) \times \log(tf_w^Q + 1) \times isf_w}{Norm} \end{aligned}$$

where $rel(S,Q)$ is the relevance of S with respect to topic Q, $Norm$ is a normalization factor of $rel(S,Q)$ over all input sentences, tf_w^S and tf_w^Q are the term frequencies of token w within S or sentences that discuss topic Q, respectively. isf_w is the inverse sentence frequency of w.

4.2 (S)pecific (C)ontent (Sum)marization

SCSum aims to extract sentences that contain detailed information about a specific author's work that is relevant to the input leaf node's topic from the set of sentences that exhibit the author-as-agent. SCSum starts by computing the topic relevance of each candidate sentence as shown in Equation (3). This process is identical to the step in GCSum, except that the term $score_S^{QR}$ in Equation (1) is replaced by $score_S^{QS}$, which is the relevance of S with respect to its sibling nodes. We

hypothesize that given a leaf node, sibling node topics may have an even more pronounced negative effect than other remaining nodes in the topic tree.

$$score_S \rightarrow score_S^{QA} + score_S^{Q} - score_S^{QS} \quad (3)$$

Context Modeling. We note that single sentences occasionally do not contain enough contexts to clearly express the idea mentioned in original articles. In fact, an agent-based sentence often introduces a concept but pertinent details are often described later. Extracting just the agent-based sentence may incompletely describe a concept and lead to false inferences. Consider the example in Figure 3. Here Sentences 0-5 are an contiguous extract of a source article being summarized, where Sentence 0 is an identified agent-based sentence. Sentence 6 shows a related work section sentence from a citing article that describes the original article. It is clear that the citing description is composed of information taken not only from the agent-based sentence but its context in the following sentences as well. This observation

0) To evaluate the quality of produced paraphrases, **we** picked at random 500 paraphrasing pairs from the lexical paraphrases produced by our algorithm.

---

1) These **pairs** were used as test data and also to evaluate whether **humans** agree on **paraphrasing judgments**.
2) The judges were given a page of guidelines, defining **paraphrase** as **approximate conceptual equivalence.**
3) The main dilemma in designing the evaluation is whether to include the context: should the **human judge** see only a paraphrase pair or should a pair of sentences containing these paraphrases also be given?
4) In a similar MT task evaluation of word-to-word translation context is usually included (Melamed, 2001).
5) Although paraphrasing is considered to be context dependent, there is no agreement on the extent.

---

6) (Barzilay and McKeown 2001) **evaluated** their **paraphrases** by asking **judges** whether **paraphrases** were "**approximately conceptually equivalent**".

Figure 3: A context modeling example.

motivates us to choose nearby sentences within a contextual window after the agent-based sentence to represent the topic. We set the contextual window to 5 and extract a maximum of 2 additional sentences. These additions are chosen based on their relevance scores to the topic, using Equation (3). Sentences with non-zero scores are then added as contexts of the anchor agent-based sentence, otherwise they are excluded. As a result, some topics may contain only a single sentence, but others may be described by additional contextual sentences.

Weighting. The score of a candidate content sentence is computed from topic relevance computation (SCSum) that includes contributions for keywords present in the current, ancestor and sibling nodes. We observe that the presence of one or more of current, ancestor and sibling nodes may affect the final score from the computation. Thus, to partially address this, we add a new weighting coefficient for the score computed from the topic relevance computation (SCSum) (Equation (3)) as follows:

$$score_S^* = w_S^{QA,Q,QS} \times score_S \quad (4)$$

where: $w_S^{QA,Q,QS}$ is a weighting coefficient that takes on differing values based on the presence of keywords in the sentence. Q, QA, and QS denote keywords from current, ancestor and sibling nodes. If the sentence contains keywords from other sibling nodes, we assign a penalty of 0.1. Otherwise, we assign a weight of 1.0, 0.5, or 0.25, based on whether keywords are present from both the ancestor node and current node, just the current node or just the ancestor node.

To build the final summary, ReWoS selects the top scoring sentence and iteratively adds the next most highly ranked sentence, until the n sentence budget is reached. We use SimRank (Li et al., 2008) to remove the next sentence to be added, if it is too similar to the sentences already in the summary.

4.3 Generation

ReWoS generates its summaries by using depth-first traversal to order the topic nodes, as in RWS-Data we observed this to be the most prevalent discourse pattern. It calls GCSum and SCSum to summarize individual nodes, distributing the total sentence budget equally among nodes.

ReWoS post-processes sentences to improve fluency where possible. We first replace agentive forms with a citation to the articles (*e.g.*, "we" $\rightarrow$ "(Wu and Oard, 2008)"). ReWoS also replaces found abbreviations with their corresponding long forms, by connecting abbreviation with their expansions by utilizing dependency relation output from the Stanford dependency parser.

System	ROUGE Recall Scores				Human Evaluation Scores			
	ROUGE-1	ROUGE-2	ROUGE-S4	ROUGE-SU4	Correctness	Novelty	Fluency	Usefulness
LEAD	0.501	0.096	0.116	0.181	3.027	2.764	3.082	2.745
MEAD	0.663	0.178	0.211	0.287	3.009	3.109	2.591	2.700
ReWoS−WCM	0.584	0.127	0.154	0.227	3.618	3.391	3.391	3.609
ReWoS−CM	0.698	0.183	0.218	0.298	3.691	3.618	2.955	3.573

Table 2: Evaluation results for ReWoS variants and baselines.

5 Evaluation

We wish to assess the quality of ReWoS, comparing to state-of-the-art generic summarization systems. We first detail our baseline systems used for performance comparison, and defined evaluation measures specific to related work summary evaluation. In our evaluation, we use our manually-compiled RWSData data set.

We benchmark ReWoS against two baseline systems: LEAD and MEAD. The LEAD baseline represents each of the cited article with an equal number of sentences. The first n sentences are drawn from the article, meaning that the title and abstract are usually extracted. The order of the article leads used in the resulting summary was determined by the order of articles to be processed. MEAD is a well-documented baseline extractive multi-document summarizer, developed in (Radev et al., 2004). MEAD offers a set of different features that can be parameterized to create resulting summaries. We conducted an internal tuning of MEAD to maximize its performance on the RWSData. The optimal configuation uses just two tuned features of *centroid* and *cosine similarity*. Note that the MEAD baseline does use the topic tree keywords in computing cosine similarity score. Our ReWoS system is the only system that leverages the topic tree *structure* which is central to our approach. In our experiments, we used MEAD toolkit[4] to produce the summaries for LEAD and MEAD baseline systems.

Automatic evaluation was performed with ROUGE (Lin, 2004), a widely used and recognized automated summarization evaluation method. We employed a number of ROUGE variants, which have been proven to correlate with human judgments in multi-document summarization (Lin, 2004). However, given the small size of our summarization dataset, we can only draw notional evidence from such an evaluation; it is not possible to find statistically significant conclusion from our evaluation.

To partially address this, we also conducted a human evaluation to assess more fine-grained qualities of our system. We asked 11 human judges to follow an evaluation guideline that we prepared, to evaluate the summary quality, consisting of the following evaluation measures:

• **Correctness:** Is the summary content actually relevant to the hierarchical topics given?

• **Novelty**: Does the summary introduce novel information that is significant in comparison with the human created summary?

• **Fluency**: Does the summary's exposition flow well, in terms of syntax as well as discourse?

• **Usefulness**: Is the summary useful in supporting the researchers to quickly grasp the related works given hierarchical topics?

Each judge was asked to grade the four summaries according to the measures on a 5-point scale of 1 (very poor) to 5 (very good). Summaries 1 and 2 come from LEAD-based and MEAD systems, respectively. Summaries 3 and 4 come from our proposed ReWoS systems, without (ReWoS−WCM) and with (ReWoS−CM) the context modeling in SCSum. All summarizers were set to yield a summary with the same length (1% of the original relevant articles, measured in sentences). Due to limited time, only 10 out of 20 evaluation sets were assessed by the evaluators. Each set was graded at least 3 times by 3 different evaluators; evaluators did not know the identities of the systems, which were randomized for each set examined.

6 Results

ROUGE results are summarized in Table 2. Surprisingly, the MEAD baseline system outperforms both LEAD baseline and ReWoS–WCM (without context modeling). Only ReWoS–CM (with

[4] `http://www.summarization.com/mead/`

context modeling) is significantly better than others, in terms of all ROUGE variants. Here are some possible reasons to explain this. First, ROUGE evaluation seems to work unreasonably when dealing with verbose summaries, often produced by MEAD. Second, related work summaries are multi-topic summaries of multi-article references. This may cause miscalculation from overlapping n-grams that occur across multiple topics or references.

Since automatic evaluation with ROUGE does not allow much introspection, we turn to our human evaluation. Results are also summarized in Table 2. They show that both ReWoS–WCM and ReWoS–CM perform significantly better than baselines in terms of correctness, novelty, and usefulness. This is because our system utilized features developed specifically for related work summarization. Also, our proposed systems compare favorably with LEAD, showing that necessary information is not only located in titles or abstracts, but also in relevant portions of the research article body.

ReWoS–CM (with context modeling) performed equivalent to ReWoS–WCM (without it) in terms of correctness and usefulness. For novelty, ReWoS–CM is better than ReWoS–WCM. It proved that the proposed component of context moding is useful in providing new information that is necessary for the related work summaries. For fluency, only ReWoS–CM is better than baseline systems. This is a negative result, but is not surprising because the summaries from the ReWoS–CM which uses context modeling seems to be longer than others. It makes the summaries quite hard to digest; some evaluators told us that they preferred the shorter summaries. A future extension is that using information fusion techniques to fuse the contextual sentences with its anchor agentive sentence.

A detailed error analysis of the results revealed that there are three main types of errors produced by our systems. The first issue is in calculating topic relevance. In the context of related work summarization, our heuristics-based strategies for sentence extraction cannot capture fully this issue. Some sentences that have high relevant scores to topics are not actually semantically relevant to the topics. The second issue of anaphoric expression is more addressable. Some extracted sentences still contain anaphoric expression (*e.g.*, "they", "these", "such", ...), making final generated summaries incoherent. The third issue is paraphrasing, where substituted paraphrases replace the original words and phrases in the source articles.

7 Conclusion and Future Work

According to the best of our knowledge, automated related work summarization has not been studied before. In this paper, we have taken the initial steps towards solving this problem, by dividing the task into general and specific summarization processes. Our initial results show an improvement over generic multi-document summarization baselines in human evaluation. However, our work shows that there is much room for additional improvement, for which we have outlined a few challenges.

A shortcoming of our current work is that we assume that a topic hierarchy tree is given as input. We feel that this is an acceptable limitation because we feel existing techniques will be able to create such input, and that the topic trees used in this study were quite simple. We plan to validate this by generating these topic trees automatically in our future work.

Exploring related work summarization comes at a timely moment, as scholars now have access to a preponderous amount of scholarly literature. Automated assistance in interpreting and organizing scholarly work will help build future applications for integration with digital libraries and reference management tools.

References

Angrosh, M. A., Stephen Cranefield, and Nigel Stanger. 2010. Context identification of sentences in related work sections using a conditional random field: towards intelligent digital libraries. In *JCDL '10: Proceedings of the 10th annual joint conference on Digital libraries*, pages 293–302. ACM.

Barzilay, Regina, Noemie Elhadad, and Kathleen R. McKeown. 2002. Inferring strategies for sentence ordering in multidocument news summariza-

tion. In *Journal of Artificial Intelligence Research*, volume 17, pages 35–55.

Baxendale, P. B. 1958. Machine-made index for technical literature - an experiment. *IBM Journal of Research Development*, 2(4):354–361.

Edmundson, H. P. 1969. New methods in automatic extracting. *Journal of the ACM*, 16(2):264–285.

Jaidka, Kokil, Christopher S. G. Khoo, and Jin-Cheon Na. 2010. Imitating human literature review writing: An approach to multi-document summarization. In *ICADL*, pages 116–119.

Li, Wenjie, Furu Wei, Qin Lu, and Yanxiang He. 2008. PNR2: Ranking sentences with positive and negative reinforcement for query-oriented update summarization. In *Proceedings of 22nd International Conference on Computational Linguistics*, pages 489–496, Manchester, UK, August.

Lin, Chin-Yew. 2004. Rouge: A package for automatic evaluation of summaries. In *Proceedings of the ACL-04 Workshop Text Summarization Branches Out*, pages 74–81, Spain, July.

Luhn, H. P. 1958. The automatic creation of literature abstracts. *IBM Journal of Research Development*, 2(2):159–165.

M., Blei D., Griffiths T. L., Jordan M. I., and Tenenbaum J. B. 2004. Hierarchical topic models and the nested chinese restaurant process. In *Advances in Neural Information Processing Systems (NIPS)*.

Mei, Qiaozhu and ChengXiang Zhai. 2008. Generating impact-based summaries for scientific literature. In *Proceedings of the 46th Annual Meeting of the Association of Computational Linguistics (ACL)*, pages 816–824, Columbus, Ohio, June.

Mihalcea, Rada and Hakan Ceylan. 2007. Explorations in automatic book summarization. In *Proceedings of Empirical Methods in Natural Language Processing - Conference on Natural Language Learning (EMNLP-CoNLL)*, pages 380–389, Prague, Czech Republic, June.

Mohammad, S., B. Dorr, M. Egan, A. Hassan, P. Muthukrishan, V. Qazvinian, D. Radev, and D. Zajic. 2009. Using citations to generate surveys of scientific paradigms. In *Proceedings of Human Language Technologies - North American Association for Computational Linguistics (HLT-NAACL)*, pages 584–592, Boulder, Colorado, June.

Nakov, Preslav I., Ariel S. Schwartz, and Marti A. Hearst. 2004. Citances: Citation sentences for semantic analysis of bioscience text. In *Workshop on Search and Discovery in Bioinformatics*.

Nallapati, R. M., A. Ahmed, E. P. Xing, and W. W. Cohen. 2008. Joint latent topic models for text and citations. In *Proceeding of the 14th ACM SIGKDD International Conference on Knowledge Discovery in Data and Data Mining*, pages 542–550.

Otterbacher, Jahna, Güneş Erkan, and Dragomir R. Radev. 2005. Using random walks for question-focused sentence retrieval. In *Proceedings of Human Language Technologies - Empirical Methods in Natural Language Processing (HLT-EMNLP '05)*, pages 915–922. ACL.

Qazvinian, Vahed and Dragomir R. Radev. 2008. Scientific paper summarization using citation summary networks. In *Proceedings of International Conference on Computational Linguistics (COLING)*, pages 689–696, Manchester, UK, August.

Radev, Dragomir R., Hongyan Jing, Malgorzata Sty, and Daniel Tam. 2004. Centroid-based summarization of multiple documents. *Information Processing & Management (IPM)*, 40(6):919–938.

Schwartz, Ariel S. and Marti Hearst. 2006. Summarizing key concepts using citation sentences. In *Proceedings of Natural language processing of biology text (BioNLP '06)*, pages 134–135. ACL.

Teufel, Simone and Marc Moens. 2002. Summarizing scientific articles: experiments with relevance and rhetorical status. *Computational Linguistics*, 28(4):409–445.

Teufel, Simone, Advaith Siddharthan, and Colin Batchelor. 2009. Towards domain-independent argumentative zoning: Evidence from chemistry and computational linguistics. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1493–1502, Singapore, August. Association for Computational Linguistics.

Teufel, Simone. 1999. *Argumentative Zoning: Information Extraction from Scientific Text*. Ph.D. thesis, University of Edinburgh.

Witten, Ian H., Gordon Paynter, Eibe Frank, Carl Gutwin, and Craig G. Nevill-Manning. 1999. Kea: Practical automatic keyphrase extraction. In *Proceedings of Digital Libraries 99 (DL'99)*, pages 254–255. ACM Press.

Wu, Yejun and Douglas W. Oard. 2008. Bilingual topic aspect classification with a few training examples. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 203–210, New York, NY, USA. ACM.

Negative Feedback: The Forsaken Nature Available for Re-ranking

Yu Hong, Qing-qing Cai, Song Hua, Jian-min Yao, Qiao-ming Zhu
School of Computer Science and Technology, Soochow University
jyao@suda.edu.cn

ABSTRACT

Re-ranking for Information Retrieval aims to elevate relevant feedbacks and depress negative ones in initial retrieval result list. Compared to relevance feedback-based re-ranking method widely adopted in the literature, this paper proposes a new method to well use three features in known negative feedbacks to identify and depress unknown negative feedbacks. The features include: 1) the minor (lower-weighted) terms in negative feedbacks; 2) hierarchical distance (HD) among feedbacks in a hierarchical clustering tree; 3) obstinateness strength of negative feedbacks. We evaluate the method on the TDT4 corpus, which is made up of news topics and their relevant stories. And experimental results show that our new scheme substantially outperforms its counterparts.

1. INTRODUCTION

When we start out an information retrieval journey on a search engine, the first step is to enter a query in the search box. The query seems to be the most direct reflection of our information needs. However, it is short and often out of standardized syntax and terminology, resulting in a large number of negative feedbacks. Some researches focus on exploring long-term query logs to acquire query intent. This may be helpful for obtaining information relevant to specific interests but not to daily real-time query intents. Especially it is extremely difficult to determine whether the interests and which of them should be involved into certain queries. Therefore, given a query, it is important to "locally" ascertain its intent by using the real-time feedbacks.

Intuitively it is feasible to expand the query using the most relevant feedbacks (Chum et al., 2007). Unfortunately search engines just offer "farraginous" feedbacks (viz. pseudo-feedback) which may involve a great number of negative feedbacks. And these negative feedbacks never honestly lag behind relevant ones in the retrieval results, sometimes far ahead because of their great literal similarity to query. These noisy feedbacks often mislead the process of learning query intent.

For so long, there had no effective approaches to confirm the relevance of feedbacks until the usage of the web click-through data (Joachims et al., 2003). Although the data are sometimes incredible due to different backgrounds and habits of searchers, they are still the most effective way to specify relevant feedbacks. This arouses recent researches about learning to rank based on supervised or semi-supervised machine learning methods, where the click-through data, as the direct reflection of query intent, offer reliable training data to learning the ranking functions.

Although the learning methods achieve substantial improvements in ranking, it can be found that lots of "obstinate" negative feedbacks still permeate retrieval results. Thus an interesting question is why the relevant feedbacks are able to describe what we really need, but weakly repel what we do not need. This may attribute to the inherent characteristics of pseudo-feedback, i.e. their high literal similarity to queries. Thus no matter whether query expansion or learning to rank, they may fall in the predicament that "favoring" relevant feedbacks may result in "favoring" negative ones, and that "hurting" negative feedbacks may result in "hurting" relevant ones.

However, there are indeed some subtle differences between relevant and negative feedbacks, e.g. the minor terms (viz. low-weighted terms in texts). Although these terms are often ignored in

Coling 2008: Poster Volume, pages 436–444,
Beijing, August 2010

relevance measurement because their little effect on mining relevant feedbacks that have the same topic or kernel, they are useful in distinguishing relevant feedbacks from negative ones. As a result, these minor terms provides an opportunity to differentiate the true query intent from its counterpart intents (called "opposite intents" thereafter in this paper). And the "opposite intents" are adopted to depress negative feedbacks without "hurting" the ranks of relevant feedbacks. In addition, hierarchical clustering tree is helpful to establish the natural similarity correlation among information. So this paper adopts the hierarchical distance among feedbacks in the tree to enhance the "opposite intents" based division of relevant and negative feedbacks. Finally, an obstinateness factor is also computed to deal with some obstinate negative feedbacks in the top list of retrieval result list. In fact, Teevan (Teevan et al., 2008) observed that most searchers tend to browse only a few feedbacks in the first one or two result pages. So our method focuses on improving the precision of highly ranked retrieval results.

The rest of the paper is organized as follows. Section 2 reviews the related work. Section 3 describes our new irrelevance feedback-based re-ranking scheme and the HD measure. Section 4 introduces the experimental settings while Section 5 reports experimental results. Finally, Section 6 draws the conclusion and indicates future work.

2. RELATED WORK

Our work is motivated by information search behaviors, such as eye-tracking and click through (Joachims, 2003). Thereinto, the click-through behavior is most widely used for acquiring query intent. Up to present, several interesting features, such as click frequency and hit time on click graph (Craswell et al., 2007), have been extracted from click-through data to improve search results. However, although effective on query learning, they fail to avoid the thorny problem that even when the typed query and the click-through data are the same, their intents may not be the same for different searchers.

A considerable number of studies have explored pseudo-feedback to learn query intent, thus refining page ranking. However, most of them focus on the relevant feedbacks. It is until recently that negative ones begin to receive some attention. Zhang (Zhang et al., 2009) utilize the irrelevance distribution to estimate the true relevance model. Their work gives the evidence that negative feedbacks are useful in the ranking process. However, their work focuses on generating a better description of query intent to attract relevant information, but ignoring that negative feedbacks have the independent effect on repelling their own kind. That is, if we have a king, we will not refuse a queen. In contrast, Wang (Wang et al., 2008) benefit from the independent effect from the negative feedbacks. Their method represents the opposite of query intent by using negative feedbacks and adopts that to discount the relevance of each pseudo-feedback to a query. However, their work just gives a hybrid representation of opposite intent which may overlap much with the relevance model. Although another work (Wang et al., 2007) of them filters query terms from the opposite intent, such filtering makes little effect because of the sparsity of the query terms in pseudo-feedback.

Other related work includes query expansion, term extraction and text clustering. In fact, query expansion techniques are often the chief beneficiary of click-through data (Chum et al., 2007). However, the query expansion techniques via clicked feedbacks fail to effectively repel negative ones. This impels us to focus on un-clicked feedbacks. Cao (Cao et al., 2008) report the effectiveness of selecting good expansion terms for pseudo-feedback. Their work gives us a hint about the shortcomings of the one-sided usage of high-weighted terms. Lee (Lee et al., 2008) adopt a cluster-based re-sampling method to emphasize the core topic of a query. Their repeatedly feeding process reveals the hierarchical relevance of pseudo-feedback.

3. RE-RANKING SCHEME

3.1 Re-ranking Scheme

The re-ranking scheme, as shown in Figure 1, consists of three components: acquiring negative feedbacks, measuring irrelevance feedbacks and re-ranking pseudo-feedback.

Given a query and its search engine results, we start off the re-ranking process after a trigger point. The point may occur at the time when searchers click on "next page" or any hyperlink.

All feedbacks before the point are assumed to have been seen by searchers. Thus the un-clicked feedbacks before the point will be treated as the known negative feedbacks because they attract no attention of searchers. This may be questioned because searchers often skip some hyperlinks that have the same contents as before, even if the links are relevant to their interests. However, such skip normally reflects the true searching intent because novel relevant feedbacks always have more attractions after all.

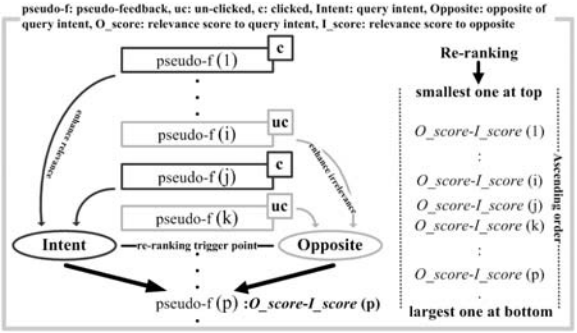


Figure 1. Re-ranking scheme

Another crucial step after the trigger point is to generate the opposite intent by using the known negative feedbacks. But now we temporarily leave the issue to Section 3.2 and assume that we have obtained a good representation of the opposite intent, and meanwhile that of query intent has been composed of the highly weighted terms in the known relevant feedbacks and query terms. Thus, given an unseen pseudo-feedback, we can calculate its overall ranking score predisposed to the opposite intent as follows:

$$R\_score = O\_score - \alpha \cdot I\_score \quad (1)$$

where the *O_score* is the relevance score to the opposite intent, *I_score* is that to the query intent and α is a weighting factor. On the basis, we re-rank the unseen feedbacks in ascending order. That is, the feedback with the largest score appears at the bottom of the ranked list.

It is worthwhile to emphasize that although the overall ranking score, i.e. *R_score*, looks similar to Wang (Wang et al., 2008) who adopts the inversely discounted value (i.e. the relevance score is calculated as $I\_score - \alpha \cdot O\_score$) to re-rank feedbacks in descending order, they are actually quite different because our overall ranking score as shown in Equation (1) is designed to depress negative feedbacks, thereby achieving the similar effect to filtering.

3.2 Representing Opposite Intent

It is necessary for the representation of opposite intent to obey two basic rules: 1) the opposite intent should be much different from the query intent; and 2) it should reflect the independent effect of negative feedbacks.

Given a query, it seems easy to represent its opposite intent by using a vector of high-weighted terms of negative feedbacks. However, the vector is actually a "close relative" of query intent because the terms often have much overlap with that of relevant feedbacks. And the overlapping terms are exactly the source of the highly ranked negative feedbacks. Thus we should throw off the overlapping terms and focus on the rest instead.

In this paper, we propose two simple facilities in representing opposite intent. One is a vector of the weighted terms (except query terms) occurring in the known negative feedbacks, named as $O(-q)$, while another further filters out the high-weighted terms occurring in the known relevant feedbacks, named as $O(-q-r)$. Although $O(-q)$ filters out query terms, the terms are so sparse that they contribute little to opposite intent learning. Thus, we will not explore $O(-q)$ further in this paper (Our preliminary experiments confirm our reasoning). In contrast, $O(-q-r)$ not only differs from the representation of query intent due to its exclusion of query terms but also emphasize the low-weighted terms occurring in negative feedbacks due to exclusion of high-weighted terms occurring in the known relevant feedbacks.

3.3 Employing Opposite Intent

Another key issue in our re-ranking scheme is how to measure the relevance of all the feedbacks to the opposite intent, i.e. *O_score*, thereby the ranking score *R_score*. For simplicity, we only consider Boolean measures in employing opposite intent to calculate the ranking score *R_score*.

Assume that given a query, there are N known relevant feedbacks and $\overline{N}$ known negative ones. First, we adopt query expansion to acquire the representation of query intent. This is done by pouring all terms of the N relevant feedbacks and query terms into a bag of words, where all the occurring weights of each term are

accumulated, and extracting *n* top-weighted terms to represent the query intent as $I(+q+r)$. Then, we use the $\bar{N}$ negative feedbacks to represent the *n*-dimensional opposite intents $O(-q-r)$. For any unseen pseudo-feedback *u*, we also represent it using an *n*-dimensional vector $V(u)$ which contains its *n* top-weighted terms. In all the representation processes, the TFIDF weighting is adopted.

Thus, for an unseen pseudo-feedback *u*, the relevance scores to the query intent and the opposite intent can be measured as:

$$\begin{aligned} I\_score(u) &= B\{\ V(u),\ I(+q+r)\ \} \\ O\_score(u) &= B\{\ V(u),\ O(-q-r)\ \} \end{aligned} \tag{2}$$

where $B\{*,*\}$ indicates Boolean calculation:

$$\begin{aligned} B\{X,Y\} &= \sum b\{x_i,Y\},\ x_i \in X \\ b\{x_i,Y\} &= \begin{cases} 1, & if\ \ x_i \in Y \\ 0, & if\ \ x_i \notin Y \end{cases} \end{aligned} \tag{3}$$

In particular, we simply set the factor α, as mentioned in Equation (1), to 1 so as to balance the effect of query intent and its opposite intent on the overall ranking score. The intuition is that if an unseen pseudo-feedback has more overlapping terms with $O(-q-r)$ than $I(+q+r)$, it will has higher probability of being depressed as an negative feedback.

Two alternatives to the above Boolean measure are to employ the widely-adopted VSM cosine measure and Kullback-Liebler (KL) divergence (Thollard et al., 2000). However, such term-weighting alternatives will seriously eliminate the effect of low-weighted terms, which is core of our negative feedback-based re-ranking scheme.

3.4 Hierarchical Distance (HD) Measure

The proposed method in Section 3.3 ignores two key issues. First, given a query, although search engine has thrown away most opposite intents, it is unavoidable that the pseudo-feedback still involves more than one opposite intent. However, the representation $O(-q-r)$ has the difficulty in highlighting all the opposite intents because the feature fusion of the representation smoothes the independent characteristics of each opposite intent. Second, given several opposite intents, they have different levels of effects on the negative score $O\_score(u)$. And the effects cannot be measured by the unilateral score.

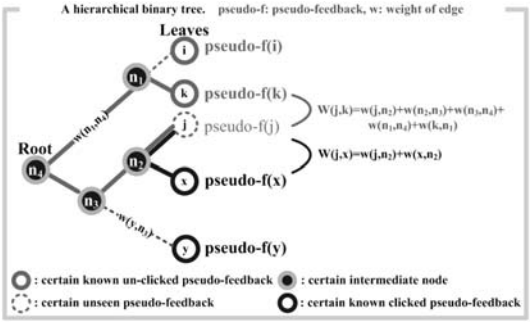


Figure 2. Weighted distance calculation

To solve the issues, we propose a hierarchical distance based negative measure, abbr. HD, which measures the distances among feedbacks in a hierarchical clustering tree, and involves them into hierarchical division of relevance score. Given two random leaves *u* and *v* in the tree, their HD score is calculated as:

$$HD\_score(u,v) = \frac{rel(u,v)}{W(u,v)} \tag{4}$$

where $rel(*,*)$ indicates textual similarity, $W(*,*)$ indicates the weighted distance in the tree, which is calculated as:

$$W(u,v) = \sum_{i \in m} w_i(u,v) \tag{5}$$

where *m* is the total number of the edges between two leaves, $w_i(*,*)$ indicates the weight of the *i-th* edge. In this paper, we adopt CLUTO to generate the hierarchical binary tree, and simply let each $w_i(*,*)$ equal 1. Thus the $W(*,*)$ becomes to be the number of edges *m*, for example, the $W(j,k)$ equals 5 in Figure 2.

On the basis, given an unseen feedback *u*, we can acquire its modified re-ranking score R'_score by following steps. First, we regard each known negative feedback as an opposite intent, following the two generative rules (mentioned in section 3.2) to generate its *n*-dimensional representation $O(-q-r)$. Additionally we represent both the known relevant feedbacks and the unseen feedback *u* as *n*-dimensional term vectors. Second, we cluster these feedbacks to generate a hierarchical binary tree and calculate the HD score for each pair of $(u,*)$, where $*$ denotes a leaf in the tree except *u*. Thus the modified ranking score is calculated as:

$$R'\_score = \sum_{i \in \bar{N}} HDI\_score(u,\bar{v}_i) - \sum_{j \in N} HDI\_score(u,v_j) \tag{6}$$

where $\bar{v}_i$ indicates the *i-th* known negative feedback in the leaves, $\bar{N}$ is the total number of

$\bar{v}$, v_j indicates the *j-th* known relevant feedback, N is the total number of v. Besides, we still adopt Boolean value to measure the textual similarity $rel(*,*)$ in both clustering process and ranking score calculation, thus the HD score in the formula (6) can be calculated as follows:

$$HD\_score(u,v) = \frac{B\{\ V(u),\ V(v)\ \}}{W(u,v)}$$
$$HD\_score(u,\bar{v}) = \frac{O\_score(u)}{W(u,\bar{v})} \qquad (7)$$

3.5 Obstinateness Factor

Additionally we involve an interesting feature, i.e. the obstinate degree, into our re-ranking scheme. The degree is represented by the rank of negative feedbacks in the original retrieval results. That is, the more "topping the list" an negative feedback is, the more obstinate it is.

Therefore we propose a hypothesis that if a feedback is close to the obstinate feedback, it should be obstinate too. Thus given an unseen feedback *u*, its relevance to an opposite intent in HD can be modified as:

$$O\_score(u)' = (1 + \frac{\beta}{rnk}) \cdot O\_score(u) \qquad (8)$$

where rnk indicates the rank of the opposite intent in original retrieval results (Note: in HD, every known negative feedback is an opposite intent), β is a smoothing factor. Because ascending order is used in our re-ranking process, by the weighting coefficient, i.e. $(1 + \beta / rnk)$, the feedback close to the obstinate opposite intents will be further depressed. But the coefficient is not commonly used. In HD, we firstly ascertain the feedback closest to *u*, and if the feedback is known to be negative, set to $\bar{v}_{\max}$, we will use the Equation (8) to punish the pair of (u, $\bar{v}_{\max}$) alone, otherwise without any punishment.

4. EXPERIMENTAL SETTING

4.1 Data Set

We evaluate our methods with two TDT collections: TDT 2002 and TDT 2003. There are 3,085 stories in the TDT 2002 collection are manually labeled as relevant to 40 news topics, 30,736 ones irrelevant to any of the topics. And 3,083 news stories in the TDT 2003 collection are labeled as relevant to another 40 news topics, 15833 ones irrelevant to them. In our evaluation, we adopt TDT 2002 as training set, and TDT 2003 as test set. Besides, only English stories are used, both Mandarin and Arabic ones are replaced by their machine-translated versions (i.e. mttkn2 released by LDC).

Corpus	good	fair	poor
TDT 2002	26	7	7
TDT 2003	22	10	8

Table 1. Number of queries referring to different types of feedbacks (Search engine: Lucene 2.3.2)

In our experiments, we realize a simple search engine based on Lucene 2.3.2 which applies document length to relevance measure on the basis of traditional literal term matching. To emulate the real retrieval process, we extract the title from the interpretation of news topic and regard it as a query, and then we run the search engine on the TDT sets and acquire the first 1000 pseudo-feedback for each query. All feedbacks will be used as the input of our re-ranking process, where the hand-crafted relevant stories default to the clicked feedbacks. By the search engine, we mainly obtain three types of pseudo-feedback: "*good*", "*fair*" and "*poor*", where "*good*" denotes that more than 5 clicked (viz. relevant) feedbacks are in the top 10, "*fair*" denotes more than 2 but less than 5, "*poor*" denotes less than 2. Table 1 shows the number of queries referring to different types of feedbacks.

4.2 Evaluation Measure

We use three evaluation measures in experiments, *P@n*, *NDCG@n* and *MAP*. Thereinto, *P@n* denotes the precision of top *n* feedbacks. On the basis, *NDCG* takes into account the influence of position to precision. *NDCG* at position *n* is calculated as:

$$NDCG@n = \frac{1}{Z_n} \cdot DCG@N = \frac{\sum_{i=1}^{n} \frac{2^{r(u_i)} - 1}{\log(1+i)}}{Z_n} \qquad (9)$$

where i is the position in the result list, Z_n is a normalizing factor and chosen so that for the perfect list *DCG* at each position equals one, and $r(u_i)$ equals 1 when u_i is relevant feedback, else 0. While *MAP* additionally takes into account recall, calculated as:

$$MAP = \frac{1}{m} \sum_{i=1}^{m} \frac{1}{R_i} (\sum_{j=1}^{k} r_i(u_j) \cdot (p@j)_i) \qquad (10)$$

where m is the total number of queries, so *MAP* gives the average measure of precision and recall

for multiple queries, R_i is the total number of feedbacks relevant to query *i*, and *k* is the number of pseudo-feedback to the query. Here *k* is indicated to be 1000, thus *Map* can give the average measure for all positions of result list.

4.3 Systems

We conduct experiments using four main systems, in which the search engine based on Lucene 2.3.2, regarded as the basic retrieval system, provides the pseudo-feedback for the following three re-ranking systems.

Exp-sys: Query is expanded by the first *N* known relevant feedbacks and represented by an *n*-dimensional vector which consists of *n* distinct terms. The standard TFIDF-weighted cosine metric is used to measure the relevance of the unseen pseudo-feedback to query. And the relevance-based descending order is in use.

Wng-sys: A system realizes the work of Wang (Wang et al., 2008), where the known relevant feedbacks are used to represent query intent, and the negative feedbacks are used to generate opposite intent. Thus, the relevance score of a feedback is calculated as $I\_score_{wng} - \alpha_w \cdot O\_score_{wng}$, and the relevance-based descending order is used in re-ranking.

Our-sys: A system is approximately similar to *Wng-sys* except that the relevance is measured by $O\_score_{our} - \alpha \cdot I\_score_{our}$ and the pseudo-feedback is re-ranked in ascending order.

Additionally both *Wng-sys* and *Our-sys* have three versions. We show them in Table 2, where "*I*" corresponds to the generation rule of query intent, "*O*" to that of opposite intent, *Rel*. means relevance measure, *u* is an unseen feedback, *v* is a known relevant feedback, $\bar{v}$ is a known negative feedback.

5. RESULTS

5.1 Main Training Result

We evaluate the systems mainly in two circumstances: when both N and $\bar{N}$ equal 1 and when they equal 5. In the first case, we assume that retrieval capability is measured under given few known feedbacks; in the second, we emulate the first page turning after several feedbacks have been clicked by searchers. Besides, the approximately optimal value of *n* for the *Exp-sys*, which is trained to be 50, is adopted as the global value for all other systems. The training results are shown in Figure 3, where the *Exp-sys* never gains much performance improvement when *n* is greater than 50. In fairness to effects of "*I*" and "*O*" on relevance measure, we also make $\bar{n}$ equal 50. In addition, all the discount factors (viz. α, α_{w2} and α_{w3}) initially equal 1, and the smoothing factor β is trained to be 0.5.

Wng-sys1	***"I"***	*n*-dimensional vector for each *v*, Number of *v* in use is *N*
	"O"	None
	Rel.	$R\_score_{w1} = (\sum_{i=1}^{N} \cos(u,v))/N$
Wng-sys2	***"I"***	Number of *v* in use is *N*, all *v* combine into a *n*-dimensional bag of words b_{w2}
	"O"	Number of $\bar{v}$ in use is *N*, all *v* combine into a *n*-dimensional words bag $\bar{b}_{w2}$
	Rel.	$R\_score_{w2} = \cos(u, b_{w2}) - \alpha_{w2} \cdot \cos(u, \bar{b}_{w2})$
Wng-sys3	***"I"***	Similar generation rules to *Wng-sys2* except that query terms are removed from bag of words b_{w3} and $\bar{b}_{w3}$
	"O"	
	Rel.	$R\_score_{w3} = \cos(u, b_{w3}) - \alpha_{w3} \cdot \cos(u, \bar{b}_{w3})$
Our-sys1	***"I"***	$I(+q+r)$ in section 3.3
	"O"	$O(-q-r)$ in section 3.2
	Rel.	$R\_score = O\_score - \alpha \cdot I\_score$
Our-sys2	***"I"***	The same generation rules to *Our-sys1*
	"O"	
	Rel.	HD algorithm: $R'\_score = \sum_{i \in \bar{N}} HDI\_score(u, \bar{v}_i) - \sum_{j \in N} HDI\_score(u, v_j)$
Our-sys3	***"I"***	The same generation rules to *Our-sys1*
	"O"	
	Rel.	HD algorithm + obstinateness factor: $O\_score(u)' = (1 + \frac{\beta}{rnk}) \cdot O\_score(u)$

Table 2. All versions of both *Wngs* and *Ours*

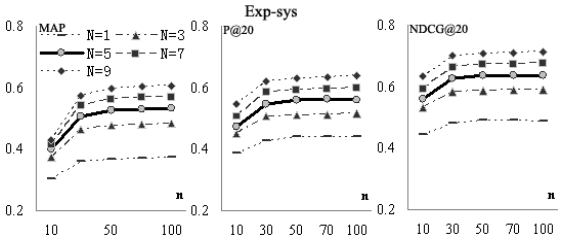


Figure 3. Parameter training of *Exp-sys*

For each query we re-rank all the pseudo-feedback, including that defined as known, so P@20 and NDCG@20 are in use to avoid over-fitting (such as P@10 and NDCG@10 given both N and $\bar{N}$ equal 5). We show the main training results in Table 3, where our methods achieve much better performances than the re-ranking methods based on relevant feedback learning when ***N***= $\bar{N}$ =5. Thereinto, our basic system, i.e. *Our-sys1*, at least achieves approximate 5% improvement on P@20, 3% on NDCG@20 and 1% on MAP than the optimal *wng-sys* (viz. *wng-sys1*). And obvi-

ously the most substantial improvements are contributed by the HD measure which even increases the P@20 of *Our-sys1* by 8.5%, NDCG@20 by 13% and MAP by 9%. But it is slightly disappointing that the obstinateness factor only has little effectiveness on performance improvement, although *Our-sys3* nearly wins the best retrieval results. This may stem from "soft" punishment on obstinateness, that is, for an unseen feedback, only the obstinate companion closest to the feedback is punished in relevance measure.

-	*Our-sys1*	*Our-sys2*	*Exp-sys*	*Wng-sys1*	*Basic*
P@20	0.6603	0.8141	0.63125	0.7051	0.6588
NDCG@20	0.7614	0.8587	0.8080	0.7797	0.6944
MAP	0.6583	0.7928	0.5955	0.7010	0.6440

Table 3. Main training results

It is undeniable that all the re-ranking systems work worse than the basic search engine when the known feedbacks are rare, such as $N = \overline{N} = 1$. This motivates an additional test on the higher values of both N and $\overline{N}$ ($N = \overline{N} = 9$), as shown in Table 4. Thus it can be found that most of the re-ranking systems achieve much better performance than the basic search engine. An important reason for this is that more key terms can be involved into representations of both query intent and its opposite intent. So it seems that more manual intervention is always reliable. However in practice, seldom searchers are willing to use an unresponsive search engine that can only offer relatively satisfactory feedbacks after lots of click-through and page turning. And in fact at least two pages (if one page includes 10 pseudo-feedback) need to be turned in the training corpus when both N and $\overline{N}$ equal 9. So we just regard the improvements benefiting from high click-through rate as an ideal status, and still adopt the practical numerical value of N and $\overline{N}$, i.e. $N = \overline{N} = 5$, to run following test.

5.2 Constraint from Query

A surprising result is that *Exp-sys* always achieves the worst MAP value, even worse than the basic search engine even if high value of *N* is in use, such as the performance when *N* equal 9 in Table 4. It seems to be difficult to question the reasonability of the system because it always selects the most key terms to represent query intent by query expansion. But an obvious difference between *Exp-sys* and other re-ranking systems could explain the result. That is the query terms consistently involved in query representation by *Exp-sys*.

systems	$N = \overline{N}$	*P@20*	*NDCG@20*	*MAP*	*Factor*
Basic	-	**0.6588**	**0.6944**	**0.6440**	-
Exp-sys	1	0.4388	0.4887	0.3683	-
	5	**0.5613**	**0.6365**	**0.5259**	-
Wng-sys1	1	0.5653	0.6184	0.5253	-
	5	**0.6564**	**0.7361**	**0.6506**	-
Wng-sys2	1	0.5436	0.6473	0.4970	α_{w2} =1
	5	**0.5910**	**0.7214**	**0.5642**	α_{w2} =1
Wng-sys3	1	0.5436	0.6162	0.4970	α_{w3} =1
	5	**0.5910**	**0.6720**	**0.5642**	α_{w3} =1
Our-sys1	1	0.5628	0.6358	0.4812	α =1
	5	**0.7031**	**0.7640**	**0.6603**	α =1
Our-sys2	1	0.6474	0.6761	0.5967	α =1
	5	**0.7885**	**0.8381**	**0.7499**	α =1
Our-sys3	1	0.6026	0.6749	0.5272	β =0.5
	5	**0.7897**	**0.8388**	**0.7464**	β =0.5

Table 4. Effects of N and $\overline{N}$ on re-ranking performance (when $N = \overline{N} = 9$, $n = \overline{n} = 50$)

In fact, *Wng-sys1* never overly favor the query terms because they are not always the main body of an independent feedback, and our systems even remove the query terms from the opposite intent directly. Conversely *Exp-sys* continuously enhances the weights of query terms which result in over-fitting and bias. The visible evidence for this is shown in Figure 4, where *Exp-sys* achieves better Precision and NDCG than the basic search engine at the top of result list but worse at the subsequent parts. The results illustrate that too much emphasis placed on query terms in query expansion is only of benefit to elevating the originally high-ranked relevant feedback but powerless to pull the straggler out of the bottom of result list.

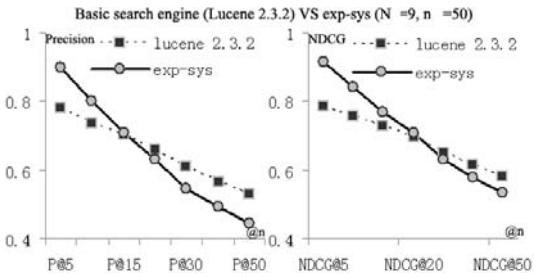


Figure 4. MAP comparison (basic vs *Exp*)

5.3 Positive Discount Loss

Obviously Wang (Wang et al., 2008) has noticed the negative effects of query terms on re-ranking. Therefore his work (reproduced by *Wng-sys1*, *2*, *3* in this paper) avoids arbitrarily enhancing the terms in query representation, even removes them as *Wng-sys3*. This indeed contributes to the

improvement of the re-ranking system, such as the better performances of *Wng-sys1, 2, 3* shown in Table 3, although *Wng-sys3* has no further improvement than *Wng-sys2* because of the sparsity of query terms. On the basis, the work regards the terms in negative feedbacks as noises and reduces their effects on relevance measure as much as possible. This should be a reasonable scheme, but interestingly it does not work well in our experiments. For example, although *Wng-sys2* and *Wng-sys3* eliminate the relevance score calculated by using the terms in negative feedbacks, they perform worse than *Wng-sys1* which never make any discount.

systems	$\alpha_* = 0.5$	$\alpha_* = 1$	$\alpha_* = 2$
Our-sys1	0.4751	0.6603	0.6901
Wng-sys2	0.6030	0.5642	0.4739
Wng-sys3	0.6084	0.5642	0.4739

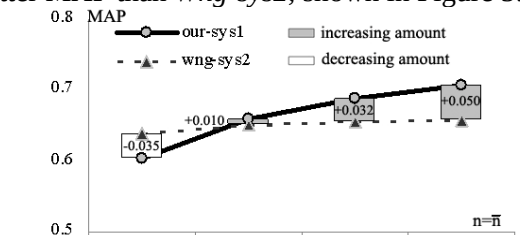
Table 5. Effects on MAP

Additionally when we increase the discount factor α_{w2} and α_{w3}, as shown in Table 5, the performances (MAP) of *Wng-sys2* and *Wng-sys3* further decrease. This illustrates that the high-weighted terms of high-ranked negative feedbacks are actually not noises. Otherwise why do the feedbacks have high textual similarity to query and even to their neighbor relevant feedbacks? Thus it actually hurts real relevance to discount the effect of the terms.

Conversely *Our-sys1* can achieve further improvement when the discount factor α increases, as shown in Table 5. It is because the discount contributes to highlighting minor terms of negative feedbacks, and these terms always have little overlap with the kernel of relevant feedbacks. Additionally the minor terms are used to generate the main body of opposite intent in our systems, thus the discount can effectively separate opposite intent from positive query representation. Thereby we can use relatively pure representation of opposite intent to detect and repel subsequent negative feedbacks.

5.4 Availability of Minor Terms

Intuitively we can involve more terms into query representation to alleviate the positive discount loss. But it does not work in practice. For example, *Wng-sys2* shown in Figure 5 has no obvious improvement no matter how many terms are included in query representation. Conversely *Our-sys1* can achieve much more improvement when it involves more terms into the opposite intent. For example, when the number of terms increases to 150, *Our-sys1* has approximately 5% better MAP than *Wng-sys2*, shown in Figure 5.



	30	50	100	150
our-sys1	0.6034	0.6603	0.6873	0.706
wng-sys2	0.6386	0.6506	0.6556	0.6563

Figure 5. Effects on MAP in modifying the dimensionality n (when $N=\overline{N}=5$, $\alpha=1$)

This result illustrates that minor terms are available for repelling negative feedbacks, but too weak to recall relevant feedbacks. In fact, the minor terms are just the low-weighted terms in text. Current text representation techniques often ignore them because of their marginality. However minor terms can reflect fine distinctions among feedbacks, even if they have the same topic. And the distinctions are of great importance when we determine why searchers say "Yes" to some feedbacks but "No" to others.

systems	*metric*	*good*	*fair*	*poor*	*global*	*Factor*
Wng-sys1	**P@20**	0.7682	0.5450	0.2643	0.6205	-
	NDCG@20	0.8260	0.6437	0.4073	0.7041	
	MAP	**0.6634**	**0.4541**	**0.9549**	**0.6620**	
Our-sys1	**P@20**	0.8273	0.5700	0.2643	0.6603	$\alpha=2$, $\beta=0.5$
	NDCG@20	0.8679	0.6620	0.4017	0.7314	
	MAP	**0.6740**	**0.4573**	**0.9184**	**0.6623**	
Our-sys2	**P@20**	0.8523	0.7600	0.2714	0.7244	$\alpha=2$, $\beta=0.5$
	NDCG@20	0.8937	0.8199	0.4180	0.7894	
	MAP	**0.7148**	**0.6313**	**0.9897**	**0.7427**	
Our-sys3	**P@20**	0.8523	0.7600	0.2714	0.7244	$\alpha=2$, $\beta=0.5$
	NDCG@20	0.8937	0.8200	0.4180	0.7894	
	MAP	**0.7145**	**0.6292**	**0.9897**	**0.7420**	

Table 6. Main test results

5.5 Test Result

We run all systems on test corpus, i.e. TDT2003, but only report four main systems: *Wng-sys1*, *Our-sys1*, *Our-sys2* and *Our-sys3*. Other systems are omitted because of their poor performances. The test results are shown in Table 6 which includes not only global performances for all test queries but also local ones on three distinct types of queries, i.e. "good", "fair" and "poor". Thereinto, *Our-sys2* achieves the best performance around all types of queries. So it is believable

that hierarchical distance of clustering tree always plays an active role in distinguishing negative feedbacks from relevant ones. But it is surprising that *Our-sys3* achieves little worse performance than *Our-sys2*. This illustrates poor robustness of obstinateness factor.

Interestingly, the four systems all achieve very high MAP scores but low P@20 and NDCG@20 for "poor" queries. This is because the queries have inherently sparse relevant feedbacks: less than 6‰ averagely. Thus the highest p@20 is only approximate 0.3, i.e. 6/20. And the low NDCG@20 is in the same way. Besides, all MAP scores for "fair" queries are the worst. We find that this type of query involves more macroscopic features which results in more kernels of negative feedbacks. Although we can solve the issue by increasing the dimensionality of opposite intent, it undoubtedly impairs the efficiency of re-ranking.

6. CONCLUSION

This paper proposes a new re-ranking scheme to well explore the opposite intent. In particular, a hierarchical distance-based (HD) measure is proposed to differentiate the opposite intent from the true query intent so as to repel negative feedbacks. Experiments show substantial outperformance of our methods.

Although our scheme has been proven effective in most cases, it fails on macroscopic queries. In fact, the key difficulty of this issue lies in how to ascertain the focal query intent given various kernels in pseudo-feedback. Fortunately, click-through data provide some useful information for learning real query intent. Although it seems feasible to generate focal intent representation by using overlapping terms in clicked feedbacks, such representation is just a reproduction of macroscopic query since the overlapping terms can only reflect common topic instead of focal intent. Therefore, it is important to segment clicked feedbacks into different blocks, and ascertain the block of greatest interest to searchers.

References

Allan, J., Lavrenko, V., and Nallapati, R. 2002. UMass at TDT 2002, Topic Detection and Tracking: Workshop.

Craswell, N., and Szummer, M. Random walks on the click graph. 2007. In Proceedings of the Conference on Research and Development in Information Retrieval. SIGIR '30. ACM Press, New York, NY, 239-246.

Cao, G. H., Nie, J. Y., and Gao, J. F. 2008. Stephen Robertson. Selecting Good Expansion Terms for Pseudo-Relevance Feedback. In Proceedings of the Conference on Research and Development in Information Retrieval. SIGIR '31. ACM Press, New York, NY, 243-250.

Chum, O., Philbin, J., Sivic, J., and Zisserman, A. 2007. Automatic query expansion with a generative feature model for object retrieval. In Proceedings of the 11th International Conference on Computer Vision, Rio de Janeiro, Brazil, 1–8.

Joachims, T., Granka, L., and Pan, B. 2003. Accurately Interpreting Clickthrough Data as Implicit Feedback. In Proceedings of the Conference on Research and Development in Information Retrieval. SIGIR '28. New York, NY, 154-161.

Lee, K. S., Croft, W. B., and Allan, J. 2008 A Cluster-Based Resampling Method for Pseudo-Relevance Feedback. In Proceedings of the Conference on Research and Development in Information Retrieval. SIGIR '31. ACM Press, New York, NY, 235-242.

Thollard, F., Dupont, P., and Higuera, L.2000. Probabilistic DFA Inference Using Kullback-Leibler Divergence and Minimality. In Proceedings of the 17th Int'l Conf on Machine Learning. San Francisco: Morgan Kaufmann, 975-982.

Teevan, J. T., Dumais, S. T., and Liebling, D. J. 2008. To Personalize or Not to Personalize: Modeling Queries with Variation in User Intent. In Proceedings of the Conference on Research and Development in Information Retrieval. SIGIR '31. New York, NY, 163-170.

Wang, X. H., Fang, H., and Zhai, C. X. 2008. A Study of Methods for Negative Relevance Feedback. In Proceedings of the Conference on Research and Development in Information Retrieval. SIGIR '31. ACM Press, New York, NY, 219-226.

Wang, X. H., Fang, H., and Zhai, C. X. 2007. Improve retrieval accuracy for difficult queries using negative feedback. In Proceedings of the sixteenth ACM conference on Conference on information and knowledge management. ACM press, New York, NY, USA, 991-994.

Zhang, P., Hou, Y. X., and Song, D. 2009. Approximating True Relevance Distribution from a Mixture Model based on Irrelevance Data. In Proceedings of the Conference on Research and Development in Information Retrieval. SIGIR '31. ACM Press, New York, NY, 107-114.

Morphological analysis can improve a CCG parser for English

Matthew Honnibal, Jonathan K. Kummerfeld and James R. Curran
School of Information Technologies
University of Sydney
{mhonn, jono, james}@it.usyd.edu.au

Abstract

Because English is a low morphology language, current statistical parsers tend to ignore morphology and accept some level of redundancy. This paper investigates how costly such redundancy is for a lexicalised grammar such as CCG.

We use morphological analysis to split verb inflectional suffixes into separate tokens, so that they can receive their own lexical categories. We find that this improves accuracy when the splits are based on correct POS tags, but that errors in gold standard or automatically assigned POS tags are costly for the system. This shows that the parser can benefit from morphological analysis, so long as the analysis is correct.

1 Introduction

English is a configurational language, so grammatical functions are mostly expressed through word order and function words, rather than with inflectional morphology. Most English verbs have four forms, and none have more than five. Most of the world's languages have far richer inflectional morphology, some with millions of possible inflection combinations.

There has been much work on addressing the sparse data problems rich morphology creates, but morphology has received little attention in the English statistical parsing literature. We suggest that English morphology may prove to be an underutilised aspect of linguistic structure that can improve the performance of an English parser. English also has a rich set of resources available, so an experiment that is difficult to perform with another language may be easier to conduct in English, and a technique that makes good use of English morphology may transfer well to a morphologically rich language. under-exploited in English natural language

In this paper, we show how morphological information can improve an English statistical parser based on a lexicalised formalism, Combinatory Categorial Grammar (CCG, Steedman, 2000), using a technique suggested for Turkish (Bozsahin, 2002) and Korean (Cha et al., 2002). They describe how a morphologically rich language can be analysed efficiently with CCG by splitting off inflectional affixes as morphological tokens. This allows the affix to receive a category that performs the feature coercion. For instance, *sleeping* would ordinarily be assigned the category $S[ng]\backslash NP$: a sentence with the $[ng]$ feature requiring a leftward NP argument. We split the word into two tokens:

sleep	-ing
$S[b]\backslash NP$	$(S[ng]\backslash NP)\backslash(S[b]\backslash NP)$

The additional token creates a separate space for inflectional information, factoring it away from the argument structure information.

Even with only 5 verb forms in English, we found that accurate morphological analysis improved parser accuracy. However, the system had trouble recovering from analysis errors caused by incorrect POS tags.

We then tested how inflection categories interacted with *hat categories*, a linguistically-motivated extension to the formalism, proposed by Honnibal and Curran (2009), that introduces some sparse data problems but improves parser effiency. The parser's accuracy improved by 0.8% when gold standard POS tags were used, but not with automatic POS tags. Our method addresses problems caused by even low morphology, and future work will make the system more robust to POS tagging errors.

Coling 2008: Poster Volume, pages 445–453,
Beijing, August 2010

2 Combinatory Categorial Grammar

Combinatory Categorial Grammar (CCG, Steedman, 2000) is a lexicalised grammar, which means that each word in the sentence is associated with a category that specifies its argument structure and the type and features of the constituent that it heads. For instance, *in* might head a PP-typed constituent with one NP-typed argument, written as PP/NP. The $/$ operator denotes an argument to the right; $\backslash$ denotes an argument to the left. For example, a transitive verb is a function from a rightward NP to and a leftward NP to a sentence, $(S\backslash NP)/NP$. The grammar consists of a few schematic rules to combine the categories:

$$\begin{array}{rcll} X/Y & Y & \Rightarrow_{>} & X \\ Y & X\backslash Y & \Rightarrow_{<} & X \\ X/Y & Y/Z & \Rightarrow_{>\mathbf{B}} & X/Z \\ Y\backslash Z & X\backslash Y & \Rightarrow_{<\mathbf{B}} & X\backslash Z \\ Y/Z & X\backslash Y & \Rightarrow_{<\mathbf{B}_{\times}} & X/Z \end{array}$$

CCGbank (Hockenmaier and Steedman, 2007) extends this grammar with a set of type-changing rules, designed to strike a better balance between sparsity in the category set and ambiguity in the grammar. We mark such productions **TC**.

In wide-coverage descriptions, categories are generally modelled as typed feature structures (Shieber, 1986), rather than atomic symbols. This allows the grammar to include head indices, and to unify under-specified features. In our notation features are annotated in square-brackets, e.g. $S[dcl]$. Head-finding indices are annotated on categories as subscripts, e.g. $(NP_y\backslash NP_y)/NP_z$. We occasionally abbreviate $S\backslash NP$ as VP, and $S[adj]\backslash NP$ as ADJ.

2.1 Statistical CCG parsing and morphology

In CCGbank, there are five features that are largely governed by the inflection of the verb:

writes/wrote	$(S[dcl]\backslash NP)/NP$
(was) written	$(S[pss]\backslash NP)/NP$
(has) written	$(S[pt]\backslash NP)/NP$
(is) writing	$(S[ng]\backslash NP)/NP$
(to) write	$(S[b]\backslash NP)/NP$

The features are necessary for satisfactory analyses. Without inflectional features, there is no way to block over-generation like *has running* or *was ran*. However, the inflectional features also create a level of redundancy if the different inflected forms are treated as individual lexical entries. The different inflected forms of a verb will all share the same set of potential argument structures, so some way of grouping the entries together is desirable.

Systems like the PET HPSG parser (Oepen et al., 2004) and the XLE LFG parser (Butt et al., 2006) use a set of lexical rules that match morphological operations with transformations on the lexical categories. For example, a lexical rule is used to ensure that an intransitive verb like *sleeping* receives the same argument structure as the base form *sleep*, but with the appropriate inflectional feature. This scheme works well for rule-based parsers, but it is less well suited for statistical parsers, as the rules propose categories but do not help the model estimate their likelihood or assign them feature weights.

Statistical parsers for lexicalised formalisms such as CCG are very sensitive to the number of categories in the lexicon and the complexity of the mapping between words and categories. The sub-task of assigning lexical categories, *supertagging* (Bangalore and Joshi, 1999), is most of the parsing task. Supertaggers mitigate sparse data problems by using a label frequency threshold to prune rare categories from the search space. Clark and Curran (2007) employ a tag dictionary that restricts the model to assigning word/category pairs seen in the training data for frequent words.

The tag dictionary causes some level of undergeneration, because not all valid word/category pairs will occur in the limited training data available. The morphological tokens we introduce help to mitigate this, by bringing together what were distinct verbs and argument structures, using lemmatisation and factoring inflection away from argument structures. The tag dictionaries for the inflectional morphemes will have very high coverage, because there are only a few inflectional categories and a few inflectional types.

3 Inflectional Categories

We implement the morphemic categories that have been discussed in the CCG literature

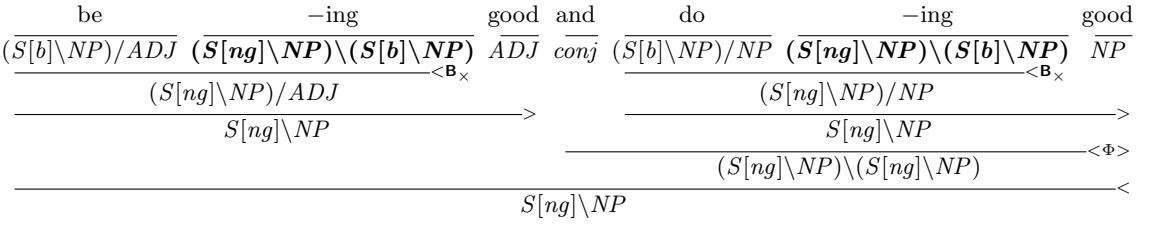


Figure 1: A single inflection category (in bold) can serve many different argument structures.

(Bozsahin, 2002; Cha et al., 2002). The inflected form is broken into two morphemes, and each is assigned a category. The category for the inflectional suffix is a function from a category with the bare-form feature $[b]$ to a category that has an inflectional feature. This prevents verbal categories from having to express their inflectional features directly. Instead, their categories only have to express their argument structure.

The CCG combinators allow multiple argument structures to share a single inflectional category. For instance, the $(S[ng]\NP)\(S[b]\NP)$ category can supply the $[ng]$ feature to all categories that have one leftward NP argument and any number of rightward arguments, via the generalised backward composition combinator. Figure 1 shows this category transforming two different argument structures, using the backward crossed composition rule ($<\mathbf{B}_\times$).

Table 1 shows the most frequent inflection categories we introduce. The majority of inflected verbs in the corpus have a subject and some number of rightward arguments, so we can almost assign one category per feature. The most frequent exceptions are participles that function as pre-nominal modifiers and verbs of speech.

Table 2 shows the inflectional token types we introduce and which features they correspond to. Our scheme largely follows the Penn Treebank tag set (Bies et al., 1995), except we avoided distinguishing past participles from past tense (*-en* vs *-ed*), because this distinction was a significant source of errors for our morphological analysis process, which relies on the part-of-speech tag.

3.1 Creating Training Data

We prepared a version of CCGbank (Hockenmaier and Steedman, 2007) with inflectional tokens. This involved the following steps:

Correcting POS tags: Our morphological analysis relies on the part-of-speech tags provided with CCGbank. We identified and corrected words whose POS tags were inconsistent with their lexical category, as discussed in Section 3.2.

Freq.	Category	Example
32.964	$(S[dcl]\NP)\(S[b]\NP)$	*He ran*
11,431	$(S[pss]\NP)\(S[b]\NP)$	*He was run down*
11,324	$(S[ng]\NP)\(S[b]\NP)$	*He was running*
4,343	$(S[pt]\NP)\(S[b]\NP)$	*He has run*
3,457	$(N/N)\(S[b]\NP)$	*the running man*
2,011	$S[dcl]\S$	*"..", he says*
1,604	$(S[dcl]\S)\(S[b]\S)$	*"..", said the boy*
169	$(S[dcl]\ADJ)\(S[b]\ADJ)$	*Here 's the deal*
55	$(S[dcl]\PP)\(S[b]\PP)$	*On it was a bee*

Table 1: The inflectional categories introduced.

Token	POS	Feat	Example
-es	VBZ	dcl	*He write -es letters*
-e	VBP	dcl	*They write -e letters*
-ed	VBD	dcl	*They write -ed letters*
-ed	VBN	pt	*They have write -ed letters*
-ed	VBN	pss	*Letters were write -ed*
-ing	VBG	ng	*They are write -ing letters*

Table 2: The inflectional token types introduced.

Lemmatising verbs and removing features: We used the morphy WordNet lemmatiser implemented in NLTK[1] to recover the lemma of the inflected verbs, identified by their POS tag (VBP, VBG, VBN or VBZ). The verb's categories were updated by switching their features to $[b]$.

Deriving inflectional categories: The generalised backward composition rules allow a functor to generalise over some sequence of argument categories, so long as they all share the same directionality. For instance, a functor $(S\NP)\(S\NP)$ could backward cross-compose into a category $((S\NP)/NP)/PP$ to its left, generalising over the two rightward arguments that were not specified by the functor's argument. It could not, however, compose into a category like $((S\NP)\NP)/PP$, because the two arguments (NP and PP) have differing direc-

[1] http://www.nltk.org

Freq.	From	To	Examples
1056	VBG	IN	*including, according, following*
379	VBN	JJ	*involved, related, concerned*
351	VBN	IN	*compared, based, given*
274	VBG	NN	*trading, spending, restructuring*
140	VBZ	NN	*is, 's, has*
102	VB	VBP	*sell, let, have*
53	VBZ	MD	*does, is, has*
45	VBG	JJ	*pending, missing, misleading*
41	VBP	MD	*do, are, have*
40	VBD	MD	*did, were, was*
334	All others		
2,815	Total		

Table 3: The most frequent POS tag conversions.

tionalities (leftward and rightward).

Without this restriction, we would only require one inflection category per feature, using inflectional categories like $S[ng]\backslash S[b]$. Instead, our inflectional categories must subcategorise for every argument except the outermost directionally consistent sequence. We discard this outermost consistent sequence, remove all features, and use the resulting category as the argument and result. We then restore the result's feature, and set the argument's feature to $[b]$.

Inserting inflectional tokens: Finally, the inflectional token is inserted after the verb, with a new node introduced to preserve binarisation.

3.2 POS tag corrections

Hockenmaier and Steedman (2007) corrected several classes of POS tag errors in the Penn Treebank when creating CCGbank. We follow Clark and Curran (2007) in using their corrected POS labels, but found that there were still some words with inconsistent POS tags and lexical categories, such as `building|NN|(S[dcl]\NP)/NP`.

In order to make our morphological analysis more consistent, we identify and correct such POS tagging errors as follows. We use two regular expressions to identify verbal lexical categories and verbal POS tags: `^\(*S\[(dcl|pss|ng|pt|b)\]` and `AUX|MD|V..` respectively. If a word has a verbal lexical category and non-verbal POS, we correct its POS tag with reference to its suffix and its category's inflectional feature. If a word has a verbal POS tag and a non-verbal lexical category, we select the POS tag that occurs most frequently with its lexical category.

The only exception are verbs functioning as nominal modifiers, such as *running* in *the running man*, which are generally POS tagged VBG but receive a lexical category of N/N. We leave these POS tagged as verbs, and instead analyse their suffixes as performing a form-function transformation that turns them from $S[b]\backslash NP$ verbs into N/N adjectives — $(N/N)\backslash(S[b]\backslash NP)$.

Table 3 lists the most common before-and-after POS tag pairs from our corrections, and the words that most frequently exemplified the pair. When compiling the table some clear errors came to light, such as the 'correction' of `is|VBZ` to `is|NN`. These errors may explain why the POS tagger's accuracy drops by 0.1% on the corrected set, and suggest that the problem of aligning POS tags and supertags is non-trivial.

In light of these errors, we experimented with an alternate strategy. Instead of correcting the POS tags, we introduced null inflectional categories that compensated for bad morphological tokenisation such as `accord|VBG|(S/S)/PP -ing|VIG|-`. The null inflectional category does not interact with the rest of the derivation, much like a punctuation symbol. This performed little better than the baseline, showing that the POS tag corrections made an important contribution, despite the problems with our technique.

3.3 Impact on CCGbank Lexicon

Verbal categories in CCGbank (Hockenmaier and Steedman, 2007) record both the valency and the inflectional morphology of the verb they are assigned to. This means $v \times i$ categories are required, where v and i are the number of distinct argument structures and inflectional features in the grammar respectively.

The inflectional tokens we propose allow inflectional morphology to be largely factored away from the argument structure, so that roughly $v + i$ verbal categories are required. A smaller category set leads to lower category ambiguity, making the assignment decision easier.

Table 4 summarises the effects of inflection categories on the lexicon extracted from CCGbank. Clark and Curran (2007) extract a set of 425 categories from the training data (Sections 02-21) that

consists of all categories that occur at least 10 times. The frequency cut off is used because the model will not have sufficient evidence to assign the other 861 categories that occur at least once, and their distribution is heavy tailed: together, they only occur 1,426 times. We refer to the frequency filtered set as the lexicon. The parser cannot assign a category outside its lexicon, so gaps in it cause under-generation.

The CCGbank lexicon includes 159 verbal categories. There are 74 distinct argument structures and 5 distinct features among these verbal categories. The grammar Clark and Curran (2007) learn therefore under-generates, because 211 of the 370 (5×74) argument structure and feature combinations are rare or unattested in the training data. For instance, there is a $(S[dcl]\backslash NP)/PP$ category, but no corresponding $(S[b]\backslash NP)/PP$, making it impossible for the grammar to generate a sentence like *I want to talk to you*, as the correct category for *talk* in this context is missing. It would be trivial to add the missing categories to the lexicon, but a statistical model would be unable to reproduce them. There are 8 occurrences of such missing categories in Section 00, the development data.

The reduction in data sparsity brought by the inflection categories causes 22 additional argument structures to cross the frequency threshold into the lexicon. A grammar induced from this corpus is thus able to generate 480 (96×5) argument structure and feature combinations, three times as many as could be generated before.

We introduce 15 inflectional categories in the corpus. The ten most frequent are shown in Table 1. The combinatory rules allow these 15 inflection categories to serve 96 argument structures, reducing the number of verbal categories in the lexicon from 159 to 89 ($74 + 15$).

The statistics at frequency 1 are less reliable, because many of the categories may be linguistically spurious: they may be artefacts caused by annotation noise in the Penn Treebank, or the conversion heuristics used by Hockenmaier and Steedman (2007).

	$\geq$	CCGbank	+Inflect
Inflection categories	10	0	15
Argument structures	10	74	96
Verb categories generated	10	159	480
All categories	10	425	375
Inflection categories	1	0	31
Argument structures	1	283	283
Verbs categories generated	1	498	1415
All categories	1	1285	1120

Table 4: Effect of inflection tokens on the category set for categories with frequency ≥ 10 and ≥ 1

3.4 Configuration of parsing experiments

We conducted two sets of parsing experiments, comparing the impact of inflectional tokens on CCGbank (Hockenmaier and Steedman, 2007) and *hat* CCGbank (Honnibal and Curran, 2009). The experiments allow us to gauge the impact of inflectional tokens on versions of CCGbank with differing numbers of verbal categories.

We used revision 1319 of the C&C parser[2] (Clark and Curran, 2007), using the best-performing configuration they describe, which used the hybrid dependency model. The most important hyper-parameters in their configuration are the β and K values, which control the workflow between the supertagger and parser. We use the Honnibal and Curran (2009) values of these parameters in our hat category experiments, described in Section 5.

Accuracy was evaluated using labelled dependency F-scores (LF). CCG dependencies are labelled by the head's lexical category and the argument slot that the dependency fills. We evaluated the baseline and inflection parsers on the unmodified dependencies, to allow direct comparison. For the inflection parsers, we pre-processed the POS-tagged input to introduce inflection tokens, and post-processed it to remove them.

We follow Clark and Curran (2007) in not evaluating accuracy over sentences for which the parser returned no analysis. The percentage of sentences analysed is described as the parser's *coverage* (C). Speed (S) figures refer to sentences parsed per second (including failures) on a dual-CPU Pentium 4 Xeon with 4GB of RAM.

[2] `http://trac.ask.it.usyd.edu.au/candc`

4 Parsing Results on CCGbank

Table 5 compares the performance of the parser on Sections 00 and 23 with and without inflection tokens. Section 00 was used for development experiments to test different approaches, and Section 23 is the test data. Similar effects were observed on both evaluation sections.

The inflection tokens had no significant impact on speed or coverage, but did improve accuracy by 0.49% F-measure when gold standard POS tags were used, compared to the baseline. However, some of the accuracy improvement can be attributed to the POS tag corrections described in Section 3.2, so the improvement from the inflection tokens alone was 0.39%.

The POS tag corrections caused a large drop in performance when automatic POS tags were used. We attribute this to the imperfections in our correction strategy. The inflection tokens improved the accuracy by 0.39%, but this was not large enough to correct for the drop in accuracy caused by the POS changes.

Another possibility is that our morphological analysis makes POS tagger errors harder to recover from. Instead of an incorrect feature value, POS tag errors can now induce poor morphological splits such as `starl|VBG -ing|VIG`. POS tagging errors are already problematic for the C&C parser, because only the highest ranked tag is forwarded to the supertagger as a feature. Our morphological analysis strategy seems to exacerbate this error propagation problem. Curran et al. (2006) showed that using a beam of POS tags as features in the supertagger and parser mitigated the loss of accuracy from POS tagging errors. Unfortunately, with our morphological analysis strategy, POS tag variations change the tokenisation of a sentence, making parsing more complicated. Perhaps the best solution would be to address the tagging errors in the treebank more thoroughly, and reform the annotation scheme to deal with particularly persistant error cases. This might improve POS tag accuracy to a level where errors are rare enough to be unproblematic.

Despite the limited morphology in English, the inflectional tokens improved the parser's accuracy when gold standard POS tags were supplied. We attribute the increase in accuracy to the more efficient word-to-category mapping caused by replacing inflected forms with lemmas, and feature-bearing verb categories with ones that only refer to the argument structure. We examined this hypothesis by performing a further experiment, to investigate how inflection tokens interact with *hat categories*, which introduce additional verbal categories that represent form-function discrepancies.

		Gold POS			Auto POS		
		LF	S	C	LF	S	C
Baseline	00	87.19	22	99.22	85.28	24	99.11
+POS	00	87.46	24	99.16	85.04	23	99.05
+Inflect	00	87.81	24	99.11	85.33	23	98.95
Baseline	23	87.69	36	99.63	85.50	36	99.58
+POS	23	87.79	36	99.63	85.06	36	99.50
+Inflect	23	88.18	36	99.58	85.42	33	99.34

Table 5: Effect of POS changes and inflection tokens on accuracy (LF), speed (S) and coverage (C) on 00 and 23.

5 Inflection Tokens and Hat Categories

Honnibal and Curran (2009) introduce an extension to the CCG formalism, *hat categories*, as an alternative way to solve the modifier category proliferation (MCP) problem. MCP is caused when a modifier is itself modified by another modifier. For instance, in the sentence *he was injured running with scissors*, *with* modifies *running*, which modifies *injured*. This produces the category $((VP\backslash VP)\backslash(VP\backslash VP))/NP$ for *with*, a rare category that is sensitive to too much of the sentence's structure.

Hockenmaier and Steedman (2007) address MCP by adding type-changing rules to CCGbank. These type-changing rules transform specific categories. They are specific to the analyses in the corpus, unlike the standard combinators, which are schematic and language universal. Honnibal and Curran's (2009) contribution is to extend the formalism to allow these type-changing rules to be lexically specified, restoring universality to the grammar — but at the cost of sparse data problems in the lexicon. Figure 2 shows how a reduced relative clause is analysed using hat categories. The hat category $(S[pss]\backslash NP)^{NP\backslash NP}$ is subject to the *unhat* rule, which unarily replaces it with its hat, $NP\backslash NP$, allowing it to function as a modifier.

Hat categories have a practical advantage for a parser that uses a supertagging phase (Bangalore

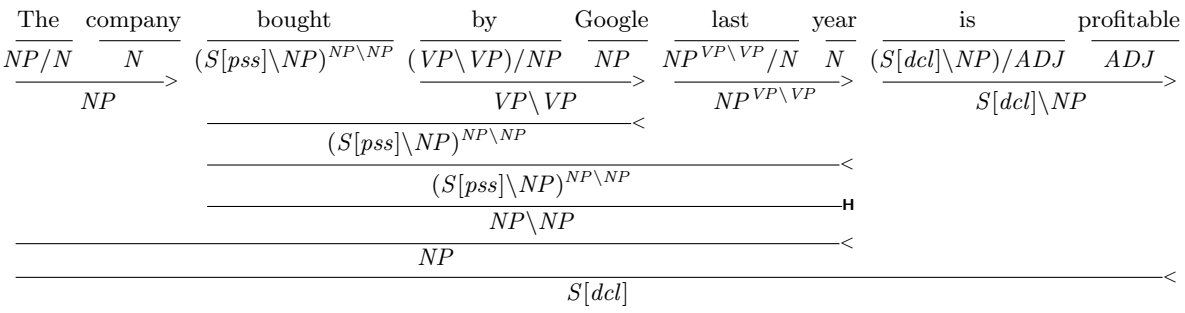


Figure 2: CCG derivation showing hat categories and the unhat rule.

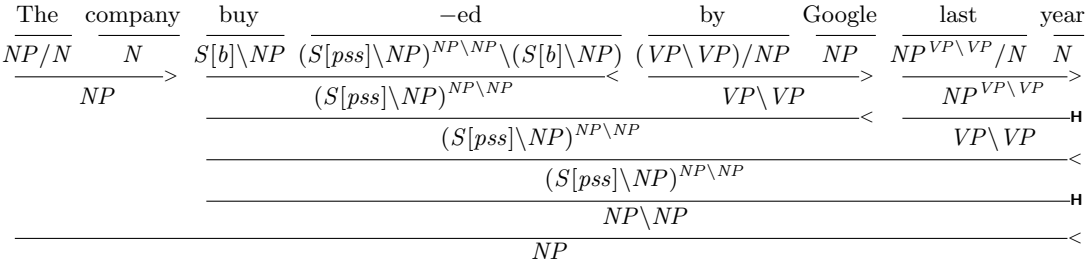


Figure 3: CCG derivation showing how inflectional tokens interact with hat categories.

and Joshi, 1999), such as the C&C system (Clark and Curran, 2007). By replacing type-changing rules with additional lexical categories, more of the work is shifted to the supertagger. The supertagging phase is much more efficient than the chart parsing stage, so redistribution of labour makes the parser considerably faster.

Honnibal and Curran (2009) found that the parser was 37% faster on the test set, at a cost of 0.5% accuracy. They attribute the drop in accuracy to sparse data problems for the supertagger, due to the increase in the number of lexical categories. We hypothesised that inflectional categories could address this problem, as the two analyses interact well.

5.1 Analyses with inflectional hat categories

Using hat categories to lexicalise type-changing rules offers attractive formal properties, and some practical advantages. However, it also misses some generalisations. A type-changing operation such as $S[ng]\backslash NP \rightarrow NP\backslash NP$ must be available to any VP. If we encounter a new word, *The company is blagging its employees*, we can generalise to the reduced relative form, *She works for that company blagging its employees* with no additional information.

This property could be preserved with some form of lexical rule, but a novel word-category pair is difficult for a statistical model to assign. Inflection tokens offer an attractive solution to this problem, as shown in Figure 3. Assigning the hat category to the suffix makes it available to any verb the suffix follows — it is just another function the inflectional suffix can perform. This generality also makes it much easier to learn, because it does not matter whether the training data happens to contain examples of a given verb perfoming that grammatical function.

We prepared a version of the Honnibal and Curran (2009) hat CCGbank, moving hats on to inflectional categories wherever possible. The hat CCGbank's lexicon contained 105 hat categories, of which 77 were assigned to inflected verbs. We introduced 33 inflection hat categories in their place, reducing the number of hat categories by 27.9%. Fewer hat categories were required because different argument structures could be served by the same inflection category. For instance, the $(S[ng]\backslash NP)^{NP\backslash NP}$ and $(S[ng]\backslash NP)^{NP\backslash NP}/NP$ categories were both replaced by the $(S[ng]\backslash NP)^{NP\backslash NP}\backslash(S[b]\backslash NP)$ category. Table 6 lists the most frequent inflection hat categories we introduce.

Freq.	Category
3332	$(S[pss]\backslash NP)^{NP\backslash NP}\backslash(S[b]\backslash NP)$
1518	$(S[ng]\backslash NP)^{NP\backslash NP}\backslash(S[b]\backslash NP)$
1231	$(S[ng]\backslash NP)^{(S\backslash NP)\backslash(S\backslash NP)}\backslash(S[b]\backslash NP)$
360	$((S[dcl]\backslash NP)/NP)^{NP\backslash NP}\backslash((S[b]\backslash NP)/NP)$
316	$(S[ng]\backslash NP)^{NP}\backslash(S[b]\backslash NP)$
234	$((S[dcl]\backslash NP)/S)^{S/S}\backslash((S[b]\backslash NP)/S)$
209	$(S[ng]\backslash NP)^{S/S}\backslash(S[b]\backslash NP)$
162	$(S[dcl]^{NP\backslash NP}\backslash NP)\backslash(S[b]\backslash NP)$
157	$((S[dcl]\backslash NP)/S)^{VP/VP}\backslash((S[b]\backslash NP)/S)$
128	$(S[pss]\backslash NP)^{S/S}\backslash(S[b]\backslash NP)$

Table 6: The most frequent inflection hat categories.

5.2 Parsing results

Table 7 shows the hat parser's performance with and without inflectional categories. We used the values for the β and K hyper-parameters described by Honnibal and Curran (2009). These hyper-parameters were tuned on Section 00, and some over-fitting seems apparent. We also followed their dependency conversion procedure, to allow evaluation over the original CCGbank dependencies and thus direct comparison with Table 5. We also merged the parser changes they described into the development version of the C&C parser we are using, for parse speed comparison.

Interestingly, incorporating the hat changes into the current version has increased the advantage of the hat categories. Honnibal and Curran report a 37% improvement in speed for the hybrid model (which we are using) on Section 23, using gold standard POS tags. With our version of the parser, the improvement is 86% (36 vs. 67 sentences parsed per second).

With gold standard POS tags, the inflection tokens improved the hat parser's accuracy by 0.8%, but decreased its speed by 24%. We attribute the decrease in speed to the increase in sentence length coupled with the new uncertainty on the inflectional tokens. Coverage increased slightly with gold standard POS tags, but decreased with automatic POS tags. We attribute this to the fact that POS tagging errors lead to morphological analysis errors.

The accuracy improvement on the hat corpus was more robust to POS tagging errors than the CCGbank results, however. This may be because POS tagging errors are already quite problematic for the hat category parser. POS tag features are more important for the supertagger than the parser, and the supertagger performs more of the work for the hat parser.

		Gold POS			Auto POS		
		LF	S	C	LF	S	C
Hat baseline	00	87.08	32	99.53	84.67	34	99.32
Hat inflect	00	87.85	37	99.63	84.99	30	98.95
Hat baseline	23	87.26	67	99.50	84.93	53	99.58
Hat inflect	23	88.06	54	99.63	85.25	43	99.38

Table 7: Effect of inflection tokens on accuracy (LF), speed (S) and coverage (C) on Sections 00 and 23.

6 Conclusion

Lexicalised formalisms like CCG (Steedman, 2000) and HPSG (Pollard and Sag, 1994) have led to high-performance statistical parsers of English, such as the C&C CCG parser (Clark and Curran, 2007) and the ENJU HPSG (Miyao and Tsuji, 2008) parser. The performance of these parsers can be partially attributed to their theoretical foundations. This is particularly true of the C&C parser, which exploits CCG's lexicalisation to divide the parsing task between two integrated models (Clark and Curran, 2004).

We have followed this formalism-driven approach by exploiting morphology for English syntactic parsing, using a strategy designed for morphologically rich languages. Combining our technique with hat categories leads to a 20% improvement in efficiency, with a 0.25% loss of accuracy. If the POS tag error problem were addressed, the two strategies combined would improve efficiency by 50%, and improve accuracy by 0.37%. These results illustrate that linguistically motivated solutions can produce substantial practical advantages for language technologies.

Acknowledgments

We would like to thank the anonymous reviewers for their feedback, and the members of the CCG-technicians mailing list for discussion about some of our analyses. Matthew Honnibal was supported by Australian Research Council (ARC) Discovery Grant DP0665973. James Curran was supported by ARC Discovery grant DP1097291 and the Capital Markets Cooperative Research Centre.

References

Srinivas Bangalore and Aravind Joshi. 1999. Supertagging: An approach to almost parsing. *Computational Linguistics*, 25(2):237–265.

Ann Bies, Mark Ferguson, Karen Katz, and Robert MacIntyre. 1995. Bracketing guidelines for Treebank II style Penn Treebank project. Technical report, MS-CIS-95-06, University of Pennsylvania, Philadelphia, PA, USA.

Cem Bozsahin. 2002. The combinatory morphemic lexicon. *Computational Linguistics*, 28(2):145–186.

Miriam Butt, Mary Dalrymple, and Tracy H. King, editors. 2006. CSLI Publications, Stanford, CA.

Jeongwon Cha, Geunbae Lee, and Jonghyeok Lee. 2002. Korean Combinatory Categorial Grammar and statistical parsing. *Computers and the Humanities*, 36(4):431–453.

Stephen Clark and James R. Curran. 2004. The importance of supertagging for wide-coverage CCG parsing. In *Proceedings of 20th International Conference on Computational Linguistics*, pages 282–288. Geneva, Switzerland.

Stephen Clark and James R. Curran. 2007. Wide-coverage efficient statistical parsing with CCG and log-linear models. *Computational Linguistics*, 33(4):493–552.

James R. Curran, Stephen Clark, and David Vadas. 2006. Multi-tagging for lexicalized-grammar parsing. In *Proceedings of the Joint Conference of the International Committee on Computational Linguistics and the Association for Computational Linguistics*, pages 697–704. Sydney, Austrailia.

Julia Hockenmaier and Mark Steedman. 2007. CCGbank: a corpus of CCG derivations and dependency structures extracted from the Penn Treebank. *Computational Linguistics*, 33(3):355–396.

Matthew Honnibal and James R. Curran. 2009. Fully lexicalising CCGbank with hat categories. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1212–1221. Singapore.

Yusuke Miyao and Jun'ichi Tsuji. 2008. Feature forest models for probabilistic HPSG parsing. *Computational Linguistics*, 34(1):35–80.

Stepan Oepen, Daniel Flickenger, Kristina Toutanova, and Christopher D. Manning. 2004. LinGO Redwoods. a rich and dynamic treebank for HPSG. *Research on Language and Computation*, 2(4):575–596.

Carl Pollard and Ivan Sag. 1994. *Head-Driven Phrase Structure Grammar*. The University of Chicago Press, Chicago.

Stuart M. Shieber. 1986. *An Introduction to Unification-Based Approaches to Grammar*, volume 4 of *CSLI Lecture Notes*. CSLI Publications, Stanford, CA.

Mark Steedman. 2000. *The Syntactic Process*. The MIT Press, Cambridge, MA, USA.

What's in a Preposition? Dimensions of Sense Disambiguation for an Interesting Word Class

Dirk Hovy, Stephen Tratz, and Eduard Hovy
Information Sciences Institute
University of Southern California
{dirkh, stratz, hovy}@isi.edu

Abstract

Choosing the right parameters for a word sense disambiguation task is critical to the success of the experiments. We explore this idea for prepositions, an often overlooked word class. We examine the parameters that must be considered in preposition disambiguation, namely context, features, and granularity. Doing so delivers an increased performance that significantly improves over two state-of-the-art systems, and shows potential for improving other word sense disambiguation tasks. We report accuracies of 91.8% and 84.8% for coarse and fine-grained preposition sense disambiguation, respectively.

1 Introduction

Ambiguity is one of the central topics in NLP. A substantial amount of work has been devoted to disambiguating prepositional attachment, words, and names. Prepositions, as with most other word types, are ambiguous. For example, the word *in* can assume both temporal ("in May") and spatial ("in the US") meanings, as well as others, less easily classifiable ("in that vein"). Prepositions typically have more senses than nouns or verbs (Litkowski and Hargraves, 2005), making them difficult to disambiguate.

Preposition sense disambiguation (PSD) has many potential uses. For example, due to the relational nature of prepositions, disambiguating their senses can help with all-word sense disambiguation. In machine translation, different senses of the same English preposition often correspond to different translations in the foreign language. Thus, disambiguating prepositions correctly may help improve translation quality.[1] Coarse-grained PSD can also be valuable for information extraction, where the sense acts as a label. In a recent study, Hwang et al. (2010) identified preposition related features, among them the coarse-grained PP labels used here, as the most informative feature in identifying caused-motion constructions. Understanding the constraints that hold for prepositional constructions could help improve PP attachment in parsing, one of the most frequent sources of parse errors.

Several papers have successfully addressed PSD with a variety of different approaches (Rudzicz and Mokhov, 2003; O'Hara and Wiebe, 2003; Ye and Baldwin, 2007; O'Hara and Wiebe, 2009; Tratz and Hovy, 2009). However, while it is often possible to increase accuracy by using a different classifier and/or more features, adding more features creates two problems: a) it can lead to overfitting, and b) while possibly improving accuracy, it is not always clear where this improvement comes from and which features are actually informative. While parameter studies exist for general word sense disambiguation (WSD) tasks (Yarowsky and Florian, 2002), and PSD accuracy has been steadily increasing, there has been no exploration of the parameters of prepositions to guide engineering decisions.

We go beyond simply improving accuracy to analyze various parameters in order to determine which ones are actually informative. We explore the different options for context and feature se-

[1] See (Chan et al., 2007) for the relevance of word sense disambiguation and (Chiang et al., 2009) for the role of prepositions in MT.

Coling 2008: Poster Volume, pages 454–462,
Beijing, August 2010

lection, the influence of different preprocessing methods, and different levels of sense granularity. Using the resulting parameters in a Maximum Entropy classifier, we are able to improve significantly over existing results. The general outline we present can potentially be extended to other word classes and improve WSD in general.

2 Related Work

Rudzicz and Mokhov (2003) use syntactic and lexical features from the governor and the preposition itself in coarse-grained PP classification with decision heuristics. They reach an average F-measure of 89% for four classes. This shows that using a very small context can be effective. However, they did not include the object of the preposition and used only lexical features for classification. Their results vary widely for the different classes.

O'Hara and Wiebe (2003) made use of a window size of five words and features from the Penn Treebank (PTB) (Marcus et al., 1993) and FrameNet (Baker et al., 1998) to classify prepositions. They show that using high level features, such as semantic roles, significantly aid disambiguation. They caution that using collocations and neighboring words indiscriminately may yield high accuracy, but has the risk of overfitting. O'Hara and Wiebe (2009) show comparisons of various semantic repositories as labels for PSD approaches. They also provide some results for PTB-based coarse-grained senses, using a five-word window for lexical and hypernym features in a decision tree classifier.

SemEval 2007 (Litkowski and Hargraves, 2007) included a task for fine-grained PSD (more than 290 senses). The best participating system, that of Ye and Baldwin (2007), extracted part-of-speech and WordNet (Fellbaum, 1998) features using a word window of seven words in a Maximum Entropy classifier. Tratz and Hovy (2009) present a higher-performing system using a set of 20 positions that are syntactically related to the preposition instead of a fixed window size.

Though using a variety of different extraction methods, contexts, and feature words, none of these approaches explores the optimal configurations for PSD.

3 Theoretical Background

The following parameters are applicable to other word classes as well. We will demonstrate their effectiveness for prepositions.

Analyzing the syntactic elements of prepositional phrases, one discovers three recurring elements that exhibit syntactic dependencies and define a prepositional phrase. The first one is the governing word (usually a noun, verb, or adjective)[2], the preposition itself, and the object of the preposition.

Prepositional phrases can be fronted ("*In May*, prices dropped by 5%"), so that the governor (in this case the verb "drop") occurs later in the sentence. Similarly, the object can be fronted (consider "*a dessert* to die for").

In the simplest version, we can do classification based only on the preposition and the governor or object alone.[3] Furthermore, directly neighboring words can influence the preposition, mostly two-word prepositions such as "out of" or "because of".

To extract the words discussed above, one can either employ a fixed window size, (which has to be large enough to capture the words), or select them based on heuristics or parsing information. The governor and object can be hard to extract if they are fronted, since they do not occur in their unusual positions relative to the preposition. While syntactically related words improve over fixed-window-size approaches (Tratz and Hovy, 2009), it is not clear which words contribute most. There should be an optimal context, i.e., the smallest set of words that achieves the best accuracy. It has to be large enough to capture all relevant information, but small enough to avoid noise words.[4] We surmise that earlier approaches were not utilizing that optimal context, but rather include a lot of noise.

Depending on the task, different levels of sense granularity may be used. Fewer senses increase the likelihood of correct classification, but may in-

[2]We will refer to the governing word, irrespective of class, as governor.

[3]Basing classification on the preposition alone is not feasible, because of the very polysemy we try to resolve.

[4]It is not obvious how much information a sister-PP can provide, or the subject of the superordinate clause.

correctly conflate prepositions. A finer granularity can help distinguish nuances and better fit the different contexts. However, it might suffer from sparse data.

4 Experimental Setup

We explore the different context types (fixed window size vs. selective), the influence of the words in that context, and the preprocessing method (heuristics vs. parsing) on both coarse and fine-grained disambiguation. We use a most-frequent-sense baseline. In addition, we compare to the state-of-the-art systems for both types of granularity (O'Hara and Wiebe, 2009; Tratz and Hovy, 2009). Their results show what has been achieved so far in terms of accuracy, and serve as a second measure for comparison beyond the baseline.

4.1 Model

We use the MALLET implementation (McCallum, 2002) of a Maximum Entropy classifier (Berger et al., 1996) to construct our models. This classifier was also used by two state-of-the-art systems (Ye and Baldwin, 2007; Tratz and Hovy, 2009). For fine-grained PSD, we train a separate model for each preposition due to the high number of possible classes for each individual preposition. For coarse-grained PSD, we use a single model for all prepositions, because they all share the same classes.

4.2 Data

We use two different data sets from existing resources for coarse and fine-grained PSD to make our results as comparable to previous work as possible.

For the coarse-grained disambiguation, we use data from the POS tagged version of the Wall Street Journal (WSJ) section of the Penn TreeBank. A subset of the prepositional phrases in this corpus is labelled with a set of seven classes: beneficial (BNF), direction (DIR), extent (EXT), location (LOC), manner (MNR), purpose (PRP), and temporal (TMP). We extract only those prepositions that head a PP labelled with such a class ($N = 35,917$). The distribution of classes is highly skewed (cf. Figure 1). We compare the results of this task to the findings of O'Hara and Wiebe (2009).

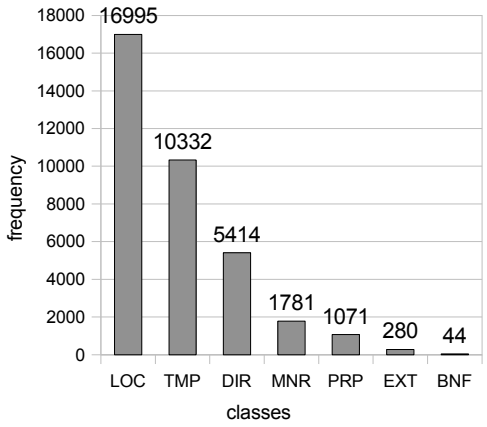


Figure 1: Distribution of Class Labels in the WSJ Section of the Penn TreeBank.

For the fine-grained task, we use data from the SemEval 2007 workshop (Litkowski and Hargraves, 2007), separate XML files for the 34 most frequent English prepositions, comprising $16,557$ training and 8096 test sentences, each instance containing one example of the respective preposition. Each preposition has between two and 25 senses (9.76 on average) as defined by The Preposition Project (Litkowski and Hargraves, 2005). We compare our results directly to the findings from Tratz and Hovy (2009). As in the original workshop task, we train and test on separate sets.

5 Results

In this section we show experimental results for the influence of word extraction method (parsing vs. POS-based heuristics), context, and feature selection on accuracy. Each section compares the results for both coarse and fine-grained granularity. Accuracy for the coarse-grained task is in all experiments higher than for the fine-grained one.

5.1 Word Extraction

In order to analyze the impact of the extraction method, we compare parsing versus POS-based heuristics for word extraction.

Both O'Hara and Wiebe (2009) and Tratz and Hovy (2009) use constituency parsers to preprocess the data. However, parsing accuracy varies,

and the problem of PP attachment ambiguity increases the likelihood of wrong extractions. This is especially troublesome in the present case, where we focus on prepositions.[5] We use the MALT parser (Nivre et al., 2007), a state-of-the-art dependency parser, to extract the governor and object.

The alternative is a POS-based heuristics approach. The only preprocessing step needed is POS tagging of the data, for which we used the system of Shen et al. (2007). We then use simple heuristics to locate the prepositions and their related words. In order to determine the governor in the absence of constituent phrases, we consider the possible governing noun, verb, and adjective. The object of the preposition is extracted as first noun phrase head to the right. This approach is faster than parsing, but has problems with long-range dependencies and fronting of the PP (e.g., the PP appearing earlier in the sentence than its governor).

extraction method	fine	coarse
MALT	84.4	94.0
Heuristics	84.8	90.9
MALT + Heuristics	84.8	91.8

Table 1: Accuracies (%) for Word-Extraction Using MALT Parser or Heuristics.

Interestingly, the extraction method does not significantly affect the final score for fine-grained PSD (see Table 1). The high score achieved when using the MALT parse for coarse-grained PSD can be explained by the fact that the parser was originally trained on that data set. The good results we see when using heuristics-based extraction only, however, means we can achieve high-accuracy PSD even without parsing.

5.2 Context

We compare the effects of fixed window size versus syntactically related words as context. Table 2 shows the results for the different types and sizes of contexts.[6]

[5]Rudzicz and Mokhov (2003) actually motivate their work as a means to achieve better PP attachment resolution.

[6]See also (Yarowsky and Florian, 2002) for experiments on the effect of varying window size for WSD.

Context	coarse	fine
2-word window	91.6	80.4
3-word window	92.0	81.4
4-word window	91.6	79.8
5-word window	91.0	78.7
Governor, prep	80.7	78.9
Prep, object	***94.2***	56.9
Governor, prep, object	94.0	***84.8***

Table 2: Accuracies (%) for Different Context Types and Sizes

The results show that the approach using both governor and object is the most accurate one. Of the fixed-window-size approaches, three words to either side works best. This does not necessarily reflect a general property of that window size, but can be explained by the fact that most governors and objects occur within this window size.[7] This distance can vary from corpus to corpus, so window size would have to be determined individually for each task. The difference between using governor and preposition versus preposition and object between coarse and fine-grained classification might reflect the annotation process: while Litkowski and Hargraves (2007) selected examples based on a search for governors[8], most annotators in the PTB may have based their decision of the PP label on the object that occurs in it. We conclude that syntactically related words present a better context for classification than fixed window sizes.

5.3 Features

Having established the context we want to use, we now turn to the details of extracting the feature words from that context.[9] Using higher-level features instead of lexical ones helps accounting for sparse training data (given an infinite amount of data, we would not need to take any higher-level

[7]Based on such statistics, O'Hara and Wiebe (2003) actually set their window size to 5.

[8]Personal communication.

[9]As one reviewer pointed out, these two dimensions are highly interrelated and influence each other. To examine the effects, we keep one dimension constant while varying the other.

features into account, since every case would be covered). Compare O'Hara and Wiebe (2009).

Following the prepocessing, we use a set of rules to select the feature words, and then generate feature values from them using a variety of feature-generating functions.[10] The word-selection rules are listed below.

Word-Selection Rules

- Governor from the MALT parse
- Object from the MALT parse
- Heuristically determined object of the preposition
- First verb to the left of the preposition
- First verb/noun/adjective to the left of the preposition
- Union of (First verb to the left, First verb/noun/adjective to the left)
- First word to the left

The feature-generating functions, many of which utilize WordNet (Fellbaum, 1998), are listed below. To conserve space, curly braces are used to represent multiple functions in a single line. The name of each feature is the combination of the word-selection rule and the output from the feature-generating function.

WordNet-based Features

- {Hypernyms, Synonyms} for {1st, all} sense(s) of the word
- All terms in the definitions ('glosses') of the word
- Lexicographer file names for the word
- Lists of all link types (e.g., meronym links) associated with the word
- Part-of-speech indicators for the existence of NN/VB/JJ/RB entries for the word
- All sentence frames for the word
- All {part, member, substance}-of holonyms for the word
- All sentence frames for the word

Other Features

- Indicator that the word-finding rule found a word
- Capitalization indicator
- {Lemma, surface form} of the word
- Part-of-speech tag for the word
- General POS tag for the word (e.g. NNS → NN, VBZ → VB)
- The {first, last} {two, three} letters of each word
- Indicators for suffix types (e.g., de-adjectival, de-nominal [non]agentive, de-verbal [non]agentive)
- Indicators for a wide variety of other affixes including those related to degree, number, order, etc. (e.g., *ultra-, poly-, post-*)
- Roget's Thesaurus divisions for the word

To establish the impact of each feature word on the outcome, we use leave-one-out and only-one evaluation.[11] The results can be found in Table 3. A word that does not perform well as the only attribute may still be important in conjunction with others. Conversely, leaving out a word may not hurt performance, despite being a good single attribute.

	coarse		fine	
Word	**LOO**	**Only**	**LOO**	**Only**
MALT governor	92.1	80.1	84.3	78.9
MALT object	93.4	94.2	84.9	56.3
Heuristics VB to left	92.0	77.9	85.0	62.1
Heur. NN/VB/ADJ to left	92.1	78.7	84.3	78.5
Heur. Governor Union	92.1	78.4	84.5	81.0
Heuristics word to left	92.0	78.8	84.4	77.2
Heuristics object	91.9	93.0	84.9	56.8
none	91.8	–	84.8	–

Table 3: Accuracies (%) for Leave-One-Out (LOO) and Only-One Word-Extraction-Rule Evaluation. *none* includes all words and serves for comparison. Important words reduce accuracy for LOO, but rank high when used as only rule.

Independent of the extraction method (MALT parser or POS-based heuristics), the governor is the most informative word. Combining several heuristics to locate the governor is the best single feature for fine-grained classification. The rule looking only for a governing verb fails to account

[10] Some words may be selected by multiple word-selection rules. For example, the governor of the preposition may be identified by the *Governor from MALT parse* rule, *first noun/verb/adjective to left*, and the *first word to the left* rule.

[11] Since the feature words are not independent of one another, neither of the two measures is decisive on its own.

	fine		coarse	
Prep	**Total**	**Acc**	**Total**	**Acc**
aboard	–	–	6	100.0
about	364	94.0	5	80.0
above	23	69.6	78	65.4
across	151	96.7	87	79.3
after	53	79.2	841	92.5
against	92	92.4	16	43.8
along	173	96.0	45	71.1
alongside	–	–	5	80.0
amid	–	–	58	70.7
among	50	80.0	358	93.9
amongst	–	–	1	0.0
around	155	69.0	107	86.0
as	84	100.0	232	84.5
astride	–	–	2	50.0
at	367	86.4	3078	92.0
atop	–	–	5	100.0
because	–	–	420	91.7
before	20	90.0	384	83.3
behind	68	77.9	65	87.7
below	–	–	94	71.3
beneath	28	78.6	11	72.7
beside	29	100.0	4	100.0
besides	–	–	1	0.0
between	102	94.1	98	84.7
beyond	–	–	45	64.4
by	248	88.3	1341	87.5
down	153	81.7	16	56.2
during	39	87.2	547	92.1
except	–	–	1	0.0
for	478	82.4	1455	84.5
from	578	85.5	1712	90.5
in	688	77.0	15706	95.0
inside	38	73.7	24	91.7
into	297	86.2	415	80.0
like	125	90.4	53	47.2
near	–	–	74	93.2
nearest	–	–	1	0.0
next	–	–	7	71.4
of	1478	87.9	71	64.8
off	76	84.2	28	75.0
on	441	81.4	2287	90.8
onto	58	91.4	15	53.3
out	–	–	90	68.9
outside	–	–	62	90.3
over	98	79.6	417	89.4
past	–	–	6	83.3
per	–	–	3	100.0
round	82	65.9	–	–
since	–	–	449	94.4
than	–	–	2	0.0
through	208	48.1	364	69.0
throughout	–	–	62	93.5
till	–	–	3	100.0
to	572	89.7	3166	97.5
toward	–	–	55	65.5
towards	102	97.1	2	100.0
under	–	–	604	91.4
underneath	–	–	2	50.0
until	–	–	208	94.2
up	–	–	20	75.0
upon	–	–	23	73.9
via	–	–	22	40.9
whether	–	–	1	100.0
while	–	–	3	33.3
with	578	84.4	272	69.5
within	–	–	213	96.2
without	–	–	69	63.8
Overall	8096	84.8	**35917**	**91.8**

Table 4: Accuracies (%) for Coarse and Fine-Grained PSD, Using MALT and Heuristics. Sorted by preposition.

for noun governors, which consequently leads to a slight improvement when left out.

Curiously, the word directly to the left is a better single feature than the object (for fine-grained classification). Leaving either of them out increases accuracy, which implies that their information can be covered by other words.

	Most Frequent Sense			O'Hara/Wiebe 2009			10-fold CV		
Class	**prec**	**rec**	**f1**	**prec**	**rec**	**f1**	**prec**	**rec**	**f1**
LOC	71.8	97.4	82.6	90.8	93.2	92.0	94.7	96.4	***95.6***
TMP	77.5	39.4	52.3	84.5	85.2	84.8	94.6	94.6	***94.6***
DIR	91.6	94.2	92.8	95.6	96.5	***96.1***	94.6	94.5	94.5
MNR	69.9	43.2	53.4	82.6	55.8	66.1	83.3	75.0	***78.9***
PRP	78.2	48.8	60.1	79.3	70.1	74.4	90.6	83.8	***87.1***
EXT	0.0	0.0	0.0	81.7	84.6	82.9	87.5	82.1	***84.7***
BNF	0.0	0.0	0.0	–	–	–	75.0	34.1	***46.9***

Table 5: Precision, Recall and F1 Results (%) for Coarse-Grained Classification. Comparison to O'Hara and Wiebe (2009). Classes ordered by frequency

5.4 Comparison with Related Work

To situate our experimental results within the body of work on PSD, we compare them to both a most-frequent-sense baseline and existing work for both granularities (see Table 6). The results use a syntactically selective context of preposition, governor, object, and word to the left as determined by combined extraction information (POS tagging and parsing).

	coarse	**fine**
Baseline	75.8	39.6
Related Work	89.3*	78.3**
Our system	93.9	84.8

Table 6: Accuracies (%) for Different Classifications. Comparison with O'Hara and Wiebe (2009)*, and Tratz and Hovy (2009)**.

Our system easily exceeds the baseline for both coarse and fine-grained PSD (see Table 6). Comparison with related work shows that we achieve an improvement of 6.5% over Tratz and Hovy (2009), which is significant at $p < .0001$, and of 4.5% over O'Hara and Wiebe (2009), which is significant at $p < .0001$.

A detailed overview over all prepositions for frequencies and accuracies of both coarse and fine-grained PSD can be found in Table 4.

In addition to overall accuracy, O'Hara and Wiebe (2009) also measure precision, recall and F-measure for the different classes. They omitted BNF because it is so infrequent. Due to different training data and models, the two systems are not strictly comparable, yet they provide a sense of the general task difficulty. See Table 5. We note that both systems perform better than the most-frequent-sense baseline. DIR is reliably classified using the baseline, while EXT and BNF are never selected for any preposition. Our method adds considerably to the scores for most classes. The low score for BNF is mainly due to the low number of instances in the data, which is why it was excluded by O'Hara and Wiebe (2009).

6 Conclusion

To get maximal accuracy in disambiguating prepositions—and also other word classes—one needs to consider context, features, and granularity. We presented an evaluation of these parameters for preposition sense disambiguation (PSD).

We find that selective context is better than fixed window size. Within the context for prepositions, the governor (head of the NP or VP governing the preposition), the object of the preposition (i.e., head of the NP to the right), and the word directly to the left of the preposition have the highest influence.[12] This corroborates the linguistic intuition that close mutual constraints hold between the elements of the PP. Each word syntactically and semantically restricts the choice of the other elements. Combining different extraction methods (POS-based heuristics and dependency parsing) works better than either one in isolation, though high accuracy can be achieved just using heuristics. The impact of context and features varies somewhat for different granularities.

[12] These will likely differ for other word classes.

Not surprisingly, we see higher scores for coarser granularity than for the more fine-grained one.

We measured success in accuracy, precision, recall, and F-measure, and compared our results to a most-frequent-sense baseline and existing work. We were able to improve over state-of-the-art systems in both coarse and fine-grained PSD, achieving accuracies of 91.8% and 84.8% respectively.

Acknowledgements

The authors would like to thank Steve DeNeefe, Victoria Fossum, and Zornitsa Kozareva for comments and suggestions. StephenTratz is supported by a National Defense Science and Engineering fellowship.

References

Baker, C.F., C.J. Fillmore, and J.B. Lowe. 1998. The Berkeley FrameNet Project. In *Proceedings of the 17th international conference on Computational linguistics-Volume 1*, pages 86–90. Association for Computational Linguistics Morristown, NJ, USA.

Berger, A.L., V.J. Della Pietra, and S.A. Della Pietra. 1996. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71.

Chan, Y.S., H.T. Ng, and D. Chiang. 2007. Word sense disambiguation improves statistical machine translation. In *Annual Meeting – Association For Computational Linguistics*, volume 45, pages 33–40.

Chiang, D., K. Knight, and W. Wang. 2009. 11,001 new features for statistical machine translation. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 218–226, Boulder, Colorado, June. Association for Computational Linguistics.

Fellbaum, C. 1998. *WordNet: an electronic lexical database*. MIT Press USA.

Hwang, J. D., R. D. Nielsen, and M. Palmer. 2010. Towards a domain independent semantics: Enhancing semantic representation with construction grammar. In *Proceedings of the NAACL HLT Workshop on Extracting and Using Constructions in Computational Linguistics*, pages 1–8, Los Angeles, California, June. Association for Computational Linguistics.

Litkowski, K. and O. Hargraves. 2005. The preposition project. *ACL-SIGSEM Workshop on "The Linguistic Dimensions of Prepositions and Their Use in Computational Linguistic Formalisms and Applications"*, pages 171–179.

Litkowski, K. and O. Hargraves. 2007. SemEval-2007 Task 06: Word-Sense Disambiguation of Prepositions. In *Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007)*, Prague, Czech Republic.

Marcus, M.P., M.A. Marcinkiewicz, and B. Santorini. 1993. Building a large annotated corpus of English: the Penn TreeBank. *Computational Linguistics*, 19(2):313–330.

McCallum, A.K. 2002. MALLET: A Machine Learning for Language Toolkit. 2002. http://mallet. cs. umass. edu.

Nivre, J., J. Hall, J. Nilsson, A. Chanev, G. Eryigit, S. Kübler, S. Marinov, and E. Marsi. 2007. MaltParser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering*, 13(02):95–135.

O'Hara, T. and J. Wiebe. 2003. Preposition semantic classification via Penn Treebank and FrameNet. In *Proceedings of CoNLL*, pages 79–86.

O'Hara, T. and J. Wiebe. 2009. Exploiting semantic role resources for preposition disambiguation. *Computational Linguistics*, 35(2):151–184.

Rudzicz, F. and S. A. Mokhov. 2003. Towards a heuristic categorization of prepositional phrases in english with wordnet. Technical report, Cornell University, arxiv1.library.cornell.edu/abs/1002.1095-?context=cs.

Shen, L., G. Satta, and A. Joshi. 2007. Guided learning for bidirectional sequence classification. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, volume 45, pages 760–767.

Tratz, S. and D. Hovy. 2009. Disambiguation of preposition sense using linguistically motivated features. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Student Research Workshop and Doctoral Consortium*, pages 96–100, Boulder, Colorado, June. Association for Computational Linguistics.

Yarowsky, D. and R. Florian. 2002. Evaluating sense disambiguation across diverse parameter spaces. *Natural Language Engineering*, 8(4):293–310.

Ye, P. and T. Baldwin. 2007. MELB-YB: Preposition Sense Disambiguation Using Rich Semantic Features. In *Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007)*, Prague, Czech Republic.

Learning to Annotate Scientific Publications

Minlie Huang
State Key Laboratory of Intelligent Technology and Systems,
Dept. Computer Science and Technology, Tsinghua University
aihuang@tsinghua.edu.cn

Zhiyong Lu
National Center for Biotechnology Information (NCBI),
U.S. National Library of Medicine, National Institutes of Health
luzh@ncbi.nlm.nih.gov

Abstract

Annotating scientific publications with keywords and phrases is of great importance to searching, indexing, and cataloging such documents. Unlike previous studies that focused on user-centric annotation, this paper presents our investigation of various annotation characteristics on service-centric annotation. Using a large number of publicly available annotated scientific publications, we characterized and compared the two different types of annotation processes. Furthermore, we developed an automatic approach of annotating scientific publications based on a machine learning algorithm and a set of novel features. When compared to other methods, our approach shows significantly improved performance. Experimental data sets and evaluation results are publicly available at the supplementary website[1].

1 Introduction

With the rapid development of the Internet, the online document archive is increasing quickly with a growing speed. Such a large volume and the rapid growth pose great challenges for document searching, indexing, and cataloging. To facilitate these processes, many concepts have been proposed, such as Semantic Web (Berners-Lee et al., 2001), Ontologies (Gruber, 1993), Open Directory Projects like Dmoz[2], folksonomies (Hotho et al., 2006), and social tagging systems like Flickr and CiteULike. Annotating documents or web-pages using Ontologies and Open Directories are often limited to a manually controlled vocabulary (developed by service providers) and a small number of expert annotators, which we call ***service-centric annotation***. By contrast, social tagging systems in which registered users can freely use arbitrary words to tag images, documents or web-pages, belong to ***user-centric annotation***. Although many advantages have been reported in user-centric annotation, low-quality and undesired annotations are always observed due to uncontrolled user behaviors (Xu et al., 2006; Sigurbjörnsson and Zwol, 2008). Moreover, the vocabulary involved in user-centric annotation is arbitrary, unlimited, and rapid-growing in nature, causing more difficulties in tag-based searching and browsing (Bao et al., 2007; Li et al., 2007).

Service-centric annotation is of importance for managing online documents, particularly in serving high-quality repositories of scientific literature. For example, in biomedicine, Gene Ontology (Ashburner et al., 2000) annotation has been for a decade an influential research topic of unifying reliable biological knowledge from the vast amount of biomedical literature. Document annotation can also greatly help service providers such as ACM/IEEE portals to provide better user experience of search. Much work has been devoted to digital document annotation, such as ontology-based (Corcho, 2006) and semantic-oriented (Eriksson, 2007).

This paper focuses on *service-centric annotation*. Our task is to assign an input document a list of entries. The entries are pre-defined by a controlled vocabulary. Due to the data availability, we study the documents and vocabulary in the

[1] http://www.ncbi.nlm.nih.gov/CBBresearch/Lu/indexing
[2] http://www.dmoz.org/

Coling 2008: Poster Volume, pages 463–471,
Beijing, August 2010

biomedical domain. We first analyze human annotation behaviors in two millions previously annotated documents. When compared to user-centric annotation, we found that the two annotation processes have major differences and that they also share some common grounds. Next, we propose to annotate new articles with a learning method based on the assumption that documents similar in content share similar annotations. To this end, we utilize a logistic regression algorithm with a set of novel features. We evaluate our approach with extensive experiments and compare it to the state of the art. The contributions of this work are two-fold: First, we present an in-depth analysis on annotation behaviors between service-centric and user-centric annotation. Second, we develop an automatic method for annotating scientific publications with significant improvements over other systems.

The remainder of the paper is organized as follows: We present several definitions in Section 2 and the analysis of annotation behaviors in Section 3. In Section 4, we presented the logistic regression algorithm for annotation. Benchmarking results are shown in Section 5. We surveyed related work in Section 6 and summarized our work in Section 7.

2 Definitions

A controlled vocabulary: *V*, a set of pre-specified entries for describing certain topics. Entries in the vocabulary are organized in a hierarchical structure. This vocabulary can be modified under human supervision.

Vocabulary Entry: an entry in a controlled vocabulary is defined as a triplet: *VE = (MT, synonyms, NodeLabels)*. *MT* is a *major term* describing the entry, and *NodeLabel*s are a list of node labels in the hierarchical tree. An entry is identified by its *MT*, and a *MT* may have multiple node labels as a *MT* may be mapped to several nodes of a hierarchical tree.

Entry Binary Relation: $ISA(VE_i, VE_j)$ means entry VE_j is a child of entry VE_i, and $SIB(VE_i, VE_j)$ meaning that VE_j is a sibling of entry VE_i. A set of relations determine the structure of a hierarchy.

Entry Depth: the depth of an entry relative to the root node in the hierarchy. The root node has a depth of 1 and the immediate children of a root node has a depth of 2, and so on. A major term may be mapped to several locations in the hierarchy, thus we have minimal, maximal, and average depths for each *MT*.

Given the above definitions, a controlled vocabulary is defined as $\{<VE_i, ISA(VE_i,VE_j), SIB(VE_i,VE_j)>|any\ i, j\ \}$. The annotation task is stated as follows: given a document *D*, predicting a list of entries *VEs* that are appropriate for annotating the document. In our framework, we approach the task as a ranking problem, as detailed in Section 4.

3 Analyzing Service-centric Annotation Behavior

Analyzing annotation behaviors can greatly facilitate assessing annotation quality, reliability, and consistency. There has been some work on analyzing social tagging behaviors in user-centric annotation systems (Sigurbjörnsson and Zwol, 2008; Suchanek et al., 2008). However, to the best of our knowledge, there is no such analysis on service-centric annotation. In social tagging systems, no specific skills are required for participating; thus users can tag the resources with arbitrary words (the words may even be totally irrelevant to the content, such as "*todo*"). By contrast, in service-centric annotation, the annotators must be trained, and they must comply with a set of strict guidelines to assure the consistent annotation quality. Therefore, it is valuable to study the differences between the two annotation processes.

3.1 PubMed Document Collection

To investigate annotation behaviors, we downloaded 2 million documents from PubMed[3], one of the largest search portals for biomedical articles. These articles were published from Jan. 1, 2000 to Dec. 31, 2008. All these documents have been manually annotated by National Library Medicine (NLM) human curators. The controlled vocabulary used in this system is the Medical Subject Headings (MeSH®)[4], a thesaurus describing various biomedical topics such as diseases, chemicals and drugs, and organisms. There are 25,588 entries in the vocabulary in 2010, and there are updates annually. By comparison, the vocabulary used in user-centric annotation is re-

[3] http://www.ncbi.nlm.nih.gov/pubmed/
[4] http://www.nlm.nih.gov/mesh/

markably larger (usually more than 1 million tags) and more dynamic (may be updated every day).

3.2 Annotation Characteristics

First, we examine the distribution of the number of annotated entries in the document collection. For each number of annotated entries, we counted the number of documents with respect to different numbers of annotations. The number of annotations per document among these 2 million documents varies from 1 (with 176,383 documents) to 97 (with one document only). The average number of annotations per document is 10.10, and the standard deviation is 5.95.

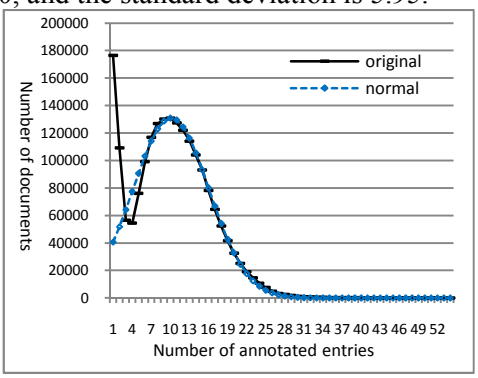


Figure 1. The original distribution and simulated normal distribution. Each data point denotes the number of documents (y-axis) that has the corresponding number of entries (x-axis).

As illustrated in Figure 1, when there are more than 4 annotations, the distribution fits a normal distribution. Comparing with user-centric annotation, there are three notable observations: a), the maximal number of annotations per document (97) is much smaller (in social tagging systems the number amounts to over 10^4) due to much less annotators involved in service-centric annotation than users in user-centric annotation; b), the number of annotations assigned to documents conforms to a normal distribution, which has not yet been reported in user-centric annotation; c), similar to user-centric annotation, the number of documents that have only one annotation accounts for a large proportion.

Second, we investigate whether the Zipf law (Zipf, 1949) holds in service-centric annotation. To this end, we ranked all the entries according to the frequency of being annotated to documents. We plotted the curve in logarithm scale, as illustrated in Figure 2. The curve can be simulated by a linear function in logarithm scale if ignoring the tail which corresponds to very infrequently used entries. To further justify this finding, we ranked all the documents according to the number of assigned annotations and plotted the curve in logarithm scale, as shown in Figure 3. Similar phenomenon is observed. In conclusion, the Zipf law also holds in service-centric annotation, just as reported in user-centric annotation (Sigurbjörnsson and Zwol, 2008).

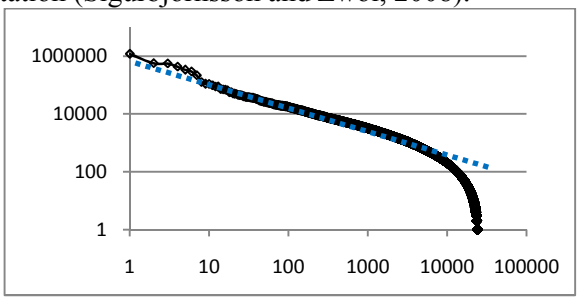


Figure 2. The distribution of annotated entry frequency. X-axis is the rank of entries (ranking by the annotation frequency), and y-axis is the frequency of an entry being used in annotation.

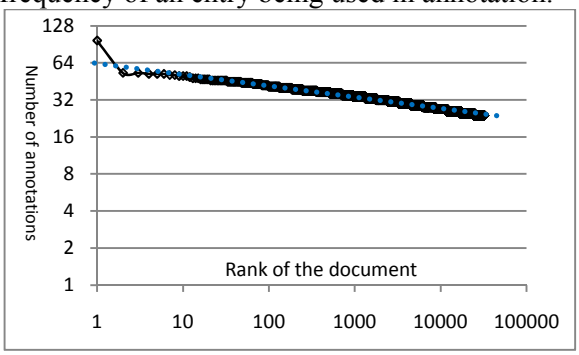


Figure 3. The distribution of the number of annotated entries. X-axis is the rank of a document (in $\log_{10}$ scale), and y-axis is the number of annotations assigned to documents (in $\log_2$ scale).

Furthermore, as mentioned in Section 2, the vocabulary corresponds to a hierarchy tree once a set of binary relations were defined. Thus we can easily obtain the minimal, maximal, and average depth of an entry. The larger depth an entry has, the more specific meaning it has.

Therefore, we investigate whether service-centric annotation is performed at very specific level (with larger depth) or general level (with smaller depth). We define prior depth and annotation depth for this study, as follows:

$$\text{PriorDepth} = \sum_{VE \in V} \frac{Dep(VE)}{|V|} \qquad (1)$$

$$\text{AnnoDepth} = \sum_{VE \in V} \Pr(VE) * Dep(VE) \quad (2)$$

$$\Pr(VE) = \frac{f(VE)}{\sum_{VE \in V} f(VE)} \quad (3)$$

where *Dep(VE)* is the minimal, maximal, or average depth of an entry, *f(VE)* is the usage frequency of *VE* in annotation, and |V| is the number of entries in the vocabulary. The two formulas are actually the mathematical expectations of the hierarchy's depth under two distributions respectively: a uniform distribution (1/|V|) and the annotation distribution (formula (3)). As shown in Table 1, the two expectations are close. This means the annotation has not been biased to either general or specific level, which suggests that the annotation quality is sound.

Dep(VE)	PriorDepth	AnnoDepth
MAX	4.88	4.56
MIN	4.25	4.02
AVG	4.56	4.29

Table 1. Annotation depth comparison.

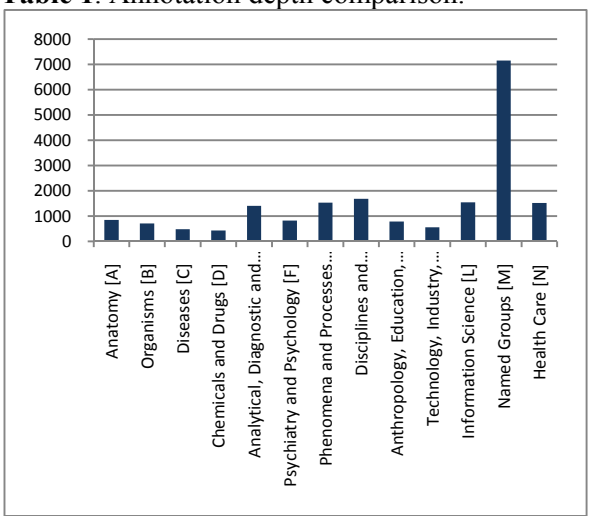


Figure 4. The imbalance frequency (y-axis) of annotated categories (x-axis).

3.3 Annotation Categorization Imbalance

We investigate here whether service-centric annotation is biased to particular categories in the hierarchy. We define a category as the label of root nodes in the hierarchy. In our vocabulary, there are 11 categories that have at least one annotation. The complete list of these categories is available at the website[5]. Three newly created categories have no annotations in the document collection. The total number of annotations within a category was divided by the number of entries in that category, as different categories may have quite different numbers of entries. If an entry is mapped to multiple locations, its annotations will be counted to corresponding categories repeatedly.

From Figure 4, we can see that there is imbalance with respect to the annotations in different categories. Category "*diseases*" has 473.5 annotations per entry (totally 4408 entries in this category). Category "*chemicals and drugs*" has 423.0 annotations per entry (with 8815 entries in total). Due to the fact that diseases and chemicals and drugs are hot scientific topics, these categories are largely under-annotated. The most frequently annotated category is: "*named groups*" (7144.4 annotations per entry), with 199 entries in total. The issue of imbalanced categorization may be due to that the topics of the document collection are of imbalance; and that the vocabulary was updated annually, so that the latest entries were used less frequently. As shown in (Sigurbjörnsson and Zwol, 2008), this imbalance issue was also observed in user-centric annotation, such as in Flickr Tagging.

4 Learning to Annotate

As shown in Section 3, there are much fewer annotations per document in service-centric annotation than in user-centric annotations. Service-centric annotation is of high quality, and is limited to a controlled vocabulary. However, manual annotation is time-consuming and labor intensive, particularly when seeking high quality. Indeed, our analysis shows that on average it takes over 90 days for a PubMed citation to be manually annotated with MeSH terms. Thus we propose **to** annotate articles automatically. Specifically, we approach this task as a ranking problem: First, we retrieve *k*-nearest neighboring (*KNN*) documents for an input document using a retrieval model (Lin and Wilbur, 2007). Second, we obtain an initial list of annotated entries from those retrieved neighboring documents. Third, we rank those entries using a logistic regression model. Finally, the top *N* ranked entries are suggested as the annotations for the target document.

4.1 Logistic Regression

We propose a probabilistic framework of directly estimating the probability that an entry can be used to annotate a document. Given a document

[5] http://www.nlm.nih.gov/mesh/2010/mesh_browser/MeSHtree.Z.html

D and an entry *VE*, we compute the probability *Pr(R(VE)|D)* directly using a logistic regression algorithm. *R(VE)* is a binary random variable indicating whether *VE* should be assigned as an annotation of the document. According to this probability, we can rank the entries obtained from neighboring documents. Much work used Logistic Regression as classification: *Pr(R=1|D)* $>\Delta$ where Δ is a threshold, but it is difficult to specify an appropriate value for the threshold in this work, as detailed in Section 5.5.

We applied the logistic regression model to this task. Logistic regression has been successfully employed in many applications including multiple ranking list merging (Si and Callan, 2005) and answer validation for question answering (Ko et al., 2007). The model gives the following probability:

$$\Pr(R(VE)\mid D)=\exp(b+\sum_{i=1}^{m}w_i*x_i)\Big/\left(1+\exp(b+\sum_{i=1}^{m}w_i*x_i)\right) \quad (4)$$

where $x=(x_1, x_2, \ldots, x_m)$ is the feature vector for *VE* and *m* is the number of features.

For an input document *D*, we can obtain an initial list of entries $\{VE_1, VE_2, \ldots, VE_n\}$ from its neighboring documents. Each entry is then represented as a feature vector as $x=(x_1, x_2, \ldots, x_m)$. Given a collection of *N* documents that have been annotated manually, each document will have a corresponding entry list, $\{VE_1, VE_2, \ldots, VE_n\}$, and each VE_i has gold-standard label $y_i=1$ if VE_i was used to annotate *D*, or $y_i=0$ otherwise. Note that the number of entries of label 0 is much larger than that of label 1 for each document. This may bias the learning algorithm. We will discuss this in Section 5.5. Given such data, the parameters can be estimated using the following formula:

$$\vec{w}^*, b^* = \arg\max_{\vec{w},b}\sum_{j=1}^{N}\sum_{i=1}^{L_j}\left(\log\Pr(R(VE_i)\mid D_j)\right) \quad (5)$$

where L_j is the number of entries to be ranked for D_j, and *N* is the total number of training documents. We can use the Quasi-Newton algorithm for parameter estimation (Minka, 2003). In this paper, we used the WEKA[6] package to implement this model.

4.2 Features

We developed various novel features to build connections between an entry and the document text. When computing these features, both the entry's text (major terms, synonyms) and the document text (title and abstract) are tokenized and stemmed. To compute these features, we collected a set of 13,999 documents (each has title, abstract, and annotations) from PubMed.

Prior probability feature. We compute the appearance probability of a major term (*MT*), estimated on the 2 million documents. This prior probability reflects the prior quality of an entry.

Unigram overlap with the title. We count the number of unigrams overlapping between the *MT* of an entry and the title, dividing by the total number of unigrams in the *MT*.

Bigram overlap with the document. We first concatenate the title and abstract, then count the number of bigram overlaps between the *MT* and the concatenated string, dividing by the total number of bigrams in the *MT*.

Multinomial distribution feature. This feature assumes that the words in a major term appear in the document text with a multinomial distribution, as follows:

$$\Pr(MT\mid Text)=\mid MT\mid !*\prod_{w\in MT}\frac{\Pr(w\mid Text)^{\#(w,MT)}}{\#(w,MT)!} \quad (6)$$

$$\Pr(w\mid Text)=(1-\lambda)\frac{\#(w,Text)}{\sum_{w_i}\#(w_i,Text)}+\lambda\Pr_c(w) \quad (7)$$

where:

#(w,MT) - The number of times that *w* appears in *MT*; Similarly for *#(w,Text)*;

|MT| - The number of single words in *MT*;

Text - Either the title or abstract, thus we have two features of this type: *Pr(MT|Title)* and *Pr(MT|Abstract)*;

$Pr_c(w)$ - The probability of word *w* occurring in a background corpus. This is obtained from a unigram language model that was estimated on the 13,999 articles;

λ – A smoothing parameter that was empirically set to be 0.2.

Query-likelihood features. The major term of an entry is viewed as a query, and this class of features computes likelihood scores between the query (as Q) and the article *D* (either the title or the abstract). We used the very classic okapi model (Robertson et al, 1994), as follows:

$$Okapi(Q,D)=\sum_{q\in Q}\frac{tf(q,D)*\log\left(\frac{N-df(q)+0.5}{df(q)+0.5}\right)}{0.5+1.5*\left(\frac{\mid D\mid}{avg(\mid D\mid)}\right)+tf(q,D)} \quad (8)$$

[6] http://www.cs.waikato.ac.nz/ml/weka/ .

where:
tf(q,D) - The count of *q* occurring in document *D*;
|D| - The total word counts in document *D*;
df(q) - The number of documents containing word *q*;
avg(|D|) - The average length of documents in the collection;
N - The total number of documents (13,999).

We have two features: *okapi(MT, Title)* and *okapi(MT, Abstract)*. In other words, the title and abstract are processed separately. The advantage of using such query-likelihood scores is that they give a probability other than a binary judgment of whether a major term should be annotated to the article, as only indirect evidence exists for annotating a vocabulary entry to an article in most cases.

Neighborhood features. The first feature represents the number of neighboring documents that include the entry to be annotated for a document. The second feature, instead of counting documents, sums document similarity scores. The two features are formulated as follows, respectively:

$$freq(MT \mid D) = \left|\{D_i \mid MT \in D_i, D_i \in \Omega_k\}\right| \quad (9)$$

$$sim(MT \mid D) = \sum_{MT \in D_i;\, D_i \in \Omega_k} sim(D, D_i) \quad (10)$$

where Ω_k is the *k*-nearest neighbors for an input document *D* and $sim(D_i,D_j)$ is the similarity score between a target document and its neighboring document, given by the retrieval model.

Synonym Features. Each vocabulary entry has synonyms. We designed two binary features: one judges whether there exists a synonym that can be exactly matched to the article text (title and abstract); and the other measures whether there exists a synonym whose unigram words have all been observed in the article text.

5 Experiment

5.1 Datasets

To justify the effectiveness of our method, we collected two datasets. We randomly selected a set of 200 documents from PubMed to train the logistic regression model (named Small200). For testing, we used a benchmark dataset, NLM2007, which has been previously used in benchmarking biomedical document annotation[7] (Aronson et al., 2004; Vasuki and Cohen, 2009; Trieschnigg et al., 2009). The two datasets have no overlap with the aforementioned 13,999 documents. Each document in these two sets has only title and abstract (i.e., no full text). The statistics listed in Table 2 show that the two datasets are alike in terms of annotations. Note that we also evaluate our method on a larger dataset of 1000 documents, but due to the length limit, the results are not presented in this paper.

Dataset	Documents	Total annotations	Average annotations
Small200	200	2,736	13.7
NLM2007	200	2,737	13.7

Table 2. Statistics of the two datasets.

5.2 Evaluation Metrics

We use *precision*, *recall*, *F-score*, and *mean average precision* (MAP) to evaluate the ranking results. As can be seen from Section 3.2, the number of annotations per document is about 10. Thus we evaluated the performance with top 10 and top 15 items.

5.3 Comparison to Other Approaches

We compare our approach to three methods on the benchmark dataset - NLM2007. The first system is NLM's MTI system (Aronson et al., 2004). This is a knowledge-rich method that employs NLP techniques, biomedical thesauruses, and a *KNN* module. It also utilizes handcrafted filtering rules for refinement. The second and third methods rank entries according to Formula (9) and (10), respectively (Trieschnigg et al., 2009).

We trained our model on Small200. All feature values were normalized to [0,1] using the maximum values of each feature. The number of neighbors was set to be 20. Neighboring documents were retrieved from PubMed using the retrieval model described in (Lin and Wilbur, 2007). Existing document annotations were not used in retrieving similar documents as they should be treated as unavailable for new documents. As the average number of annotations per document is around 13 (see Table 2), we computed precision, recall, F-score, and MAP with top 10 and 15 entries, respectively.

Results in Table 3 demonstrate that our method outperforms all other methods. It has substantial improvements over MTI. To justify whether the improvement over using *neighbor-*

[7] http://ii.nlm.nih.gov/.

hood similarity is significant, we conducted the *Paired t-test* (Goulden, 1956). When comparing results of using *learning* vs. *neighborhood similarity* in Table 3, the p-value is 0.028 for top 10 and 0.001 for top 15 items. This shows that the improvement achieved by our approach is statistically significant (at significance level of 0.05).

	Methods	Pre.	Rec.	F.	MAP
Top 10	MTI	.468	.355	.404	.400
	Frequency	.635	.464	.536	.598
	Similarity	.643	.469	.542	.604
	Learning	**.657**	**.480**	**.555**	**.622**
Top 15	MTI	.404	.442	.422	.400
	Frequency	.512	.562	.536	.598
	Similarity	.524	.574	.548	.604
	Learning	**.539**	**.591**	**.563**	**.622**

Table 3. Comparative results on NLM2007.

5.4 Choosing Parameter *k*

We demonstrate here our search for the optimal number of neighboring documents in this task. As shown in Table 4, the more neighbors, the larger number of gold-standard annotations would be present in neighboring documents. With 20 neighbors a fairly high upper-bound recall (*UBR*) is observed (about 85% of gold-standard annotations of a target document were present in its 20 neighbors' annotations), and the average number of entries (*Avg_VE*) to be ranked is about 100.

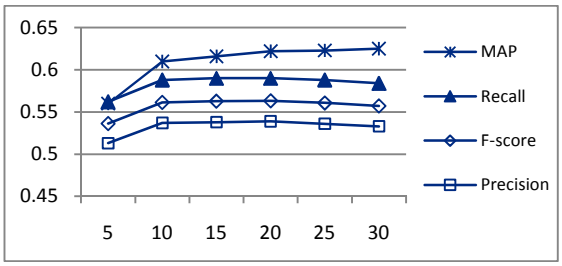


Figure 5. The performance (y-axis) varies with the number of neighbors (x-axis).

Measure	The number of neighboring documents					
	5	10	15	20	25	30
UBR	.704	.793	.832	.856	.871	.882
Avg_VE	38.8	64.1	83.6	102.2	119.7	136.4

Table 4. The upper-bound recall (*UBR*) and average number of entries (*Avg_VE*) with different number of neighboring documents.

To investigate whether the number of neighboring documents affects performance, we experimented with different numbers of neighboring documents. We trained a model on Small200, and tested it on NLM2007. The curves in Figure 5 show that the performance becomes very close when choosing no less than 10 neighbors. This infers that reliable performance can be obtained. The best performance (F-score of 0.563) is obtained with 20 neighbors. Thus, the parameter *k* is set to be 20.

5.5 Data Imbalance Issue

As mentioned in Section 4.1, there is a data imbalance issue in our task. For each document, we obtained an initial list of entries from 20 neighboring documents. The average number of gold-standard annotations is about 13, while the average number of entries to be ranked is around 100 (see Table 4). Thus the number of entries of label 0 (negative examples) is much larger than that of label 1 (positive examples). We did not apply any filtering strategy because the gold-standard annotations are not proportional to their occurring frequency in the neighboring documents. In fact, as shown in Figure 6, the majority of gold-standard annotations appear in only few documents among 20 neighbors. For example, there are about 250 gold-standard annotations appearing in only one of 20 neighboring documents and 964 appearing in less than 6 neighboring documents. Therefore, applying any filtering strategy based on their occurrence in neighboring documents may be harmful to the performance.

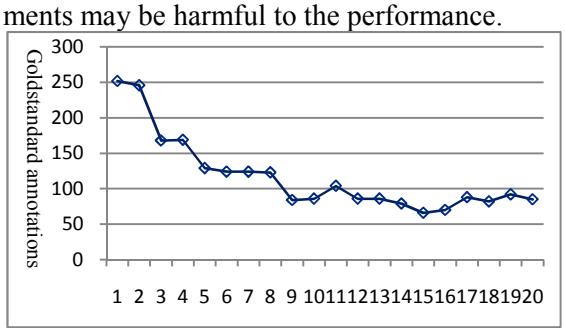


Figure 6. The distribution of annotations. X-axis is the number of neighboring documents in which gold-standard annotations are found.

5.6 Feature Analysis

To investigate the impact of different groups of features, we performed a feature ablation study. The features were divided into four groups. For each round of this study, we remove one group of features from the entire feature set, re-train the model on Small200, and then test the performance on NLM2007 with top 15 entries. We divided the features into four independent groups:

prior probability features, neighborhood features, synonym features, and other features (including unigram/bigram feature, query likelihood feature, etc., see Section 4.2). Results in Table 5 show that neighborhood features are dominant: removing such features leads to a remarkable decrease in performance. On the other hand, using only neighborhood features (the last row) yields significant worse results than using all features. This means that combining all features together indeed contributes to the optimal performance.

Feature Set	Pre.	Rec.	F.	MAP
All features	**.539**	**.591**	**.563**	**.622**
- Prior probability	.538	.590	.563	.622
- Neighborhood features	.419*	.459*	.438*	.467*
- Synonym features	.532	.583	.556	.611
- Other features	.529	.580	.553	.621
Only neighborhood features	.523*	.573*	.547*	.603*

Table 5. Feature analysis. Those marked by stars are significantly worse than the best results.

5.7 Discussions

All methods that rely on neighboring documents have performance ceilings. Specifically, for the NLM2007 dataset, the upper bound recall is around 85.6% with 20 neighboring documents, as shown in Table 5. Due to the same reason, this genre of methods is also limited to recommend entries that are recently added to the controlled vocabulary as such entries may have not been annotated to any document yet. This phenomenon has been demonstrated in the annotation behavior analysis: those latest entries have substantially fewer annotations than older ones.

6 Related Work

Our work is closely related to ontology-based or semantic-oriented document annotation (Corcho, 2006; Eriksson, 2007). This work is also related to *KNN*-based tag suggestion or recommendation systems (Mishne, 2006).

The task here is similar to keyword extraction (Nguyen and Kan, 2007; Jiang et al., 2009), but there is a major difference: keywords are always occurring in the document, while when an entry of a controlled vocabulary was annotated to a document, it may not appear in text at all.

As for the task tackled in this paper, i.e., annotating biomedical publications, three genres of approaches have been proposed: (1) *k-Nearest Neighbor* model: selecting annotations from neighboring documents, ranking and filtering those annotations (Vasuki and Cohen, 2009; Trieschnigg et al., 2009). (2) Classification model: learning the association between the document text and an entry (Ruch, 2006). (3) Based on knowledge resources: using domain thesauruses and NLP techniques to match an entry with concepts in the document text (Aronson, 2001; Aronson et al., 2004). (4) LDA-based topic model: (Mörchen et al., 2008).

7 Conclusion

This paper presents a novel study on service-centric annotation. Based on the analysis results of 2 million annotated scientific publications, we conclude that service-centric annotation exhibits the following unique characteristics: a) the number of annotation per document is significant smaller, but it conforms to a normal distribution; and b) entries of different granularity (general vs. specific) are used appropriately by the trained annotators. Service-centric and user-centric annotations have in common that the Zipf law holds and categorization imbalance exists.

Based on these observations, we introduced a logistic regression approach to annotate publications, with novel features. Significant improvements over other systems were obtained on a benchmark dataset. Although our features are tailored for this task in biomedicine, this approach may be generalized for similar tasks in other domains.

Acknowledgements

This work was supported by the Intramural Research Program of the NIH, National Library of Medicine. The first author was also supported by the Chinese Natural Science Foundation under grant No. 60803075 and the grant from the International Development Research Center, Ottawa, Canada IRCI.

References

Alan R. Aronson. Effective mapping of biomedical text to the UMLS Metathesaurus: the metamap program. In Proc AMIA Symp 2001. p. 17-21.

Alan Aronson, Alan R. Aronson, James Mork, James G. Mork, Clifford Gay, Clifford W. Gay, Susanne Humphrey, Susanne M. Humphrey, Willie Rogers, Willie J. Rogers. The NLM Indexing Initiative's

Medical Text Indexer. Stud Health Technol Inform. 2004;107(Pt 1):268-72.

Michael Ashburner, Catherine A. Ball, Judith A. Blake, David Botstein, Heather Butler, et al. Gene Ontology: tool for the unification of biology. Nat Genet. 2000 May; 25(1):25-9.

Shenghua Bao, Xiaoyuan Wu, Ben Fei, Guirong Xue, Zhong Su, and Yong Yu. Optimizing Web Search Using Social Annotations. WWW 2007, May 8–12, 2007, Banff, Alberta, Canada. Pp 501-510.

Tim Berners-Lee, James Hendler and Ora Lassila. The Semantic Web. Scientific American Magazine. (May 17, 2001).

Oscar Corcho. Ontology based document annotation: trends and open research problems. International Journal of Metadata, Semantics and Ontologies, Volume 1, Issue 1, Pages: 47-57, 2006.

Henrik Eriksson. An Annotation Tool for Semantic Documents. In Proceedings of the 4th European conference on The Semantic Web: Research and Applications, pages 759-768, 2007. Innsbruck, Austria.

Cyril Harold Goulden. Methods of Statistical Analysis, 2nd ed. New York: Wiley, pp. 50-55, 1956.

Thomas R. Gruber (1993). A Translation Approach to Portable Ontology Specifications. Knowledge Acquisition, 5(2), 1993, pp. 199-220.

Andreas Hotho, Robert Jaschke, Christoph Schmitz, Gerd Stumme. Information Retrieval in Folksonomies: Search and Ranking. In "The Semantic Web: Research and Applications", Vol. 4011 (2006), pp. 411-426.

Xin Jiang, Yunhua Hu, Hang Li. A Ranking Approach to Keyphrase Extraction. SIGIR'09, July 19–23, 2009, Boston, Massachusetts, USA.

Jeongwoo Ko, Luo Si, Eric Nyberg. A Probabilistic Framework for Answer Selection in Question Answering. Proceedings of NAACL HLT 2007, pages 524–531, Rochester, NY, April 2007.

Rui Li, Shenghua Bao, Ben Fei, Zhong Su, and Yong Yu. Towards Effective Browsing of Large Scale Social Annotations. In WWW '07: Proceedings of the 16th international conference on World Wide Web, 2007.

Jimmy Lin and W. John Wilbur. PubMed related articles: a probabilistic topic-based model for content similarity. BMC Bioinformatics 8: (2007).

Thomas P. Minka. A Comparison of Numerical Optimizers for Logistic Regression. 2003. Unpublished draft.

Gilad Mishne. AutoTag: A Collaborative Approach to Automated Tag Assignment for Weblog Posts. WWW 2006, May 22–26, 2006, Edinburgh, Scotland. pages 953–954.

Fabian Mörchen, Mathäus Dejori, Dmitriy Fradkin, Julien Etienne, Bernd Wachmann, Markus Bundschus. Anticipating annotations and emerging trends in biomedical literature. In KDD '08: pp. 954-962.

Thuy Dung Nguyen and Min-Yen Kan. Keyphrase Extraction in Scientific Publications. In Proc. of International Conference on Asian Digital Libraries (ICADL '07), pages 317-326.

Stephen E. Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu, and Mike Gatford. Okapi at TREC-3. In Proceedings of the Third Text REtrieval Conference (TREC 1994). Gaithersburg, USA, November 1994.

Patrick Ruch. Automatic assignment of biomedical categories: toward a generic approach. Bioinformatics. 2006 Mar 15;22(6):658-64.

Luo Si and Jamie Callan. 2005 CLEF2005: Multilingual retrieval by combining multiple multilingual ranked lists. In Proceedings of Cross-Language Evaluation Forum.

Börkur Sigurbjörnsson and Roelof van Zwol. Flickr Tag Recommendation based on Collective Knowledge. WWW 2008, April 21–25, 2008, Beijing, China. Pp. 327-336.

Fabian M. Suchanek, Milan Vojnovi´c, Dinan Gunawardena. Social Tags: Meaning and Suggestions. CIKM'08, October 26–30, 2008, Napa Valley, California, USA.

Dolf Trieschnigg, Piotr Pezik, Vivian Lee, Franciska de Jong, Wessel Kraaij, Dietrich Rebholz-Schuhmann. MeSH Up: effective MeSH text classification for improved document retrieval. Bioinformatics, Vol. 25 no. 11 2009, pages 1412–1418.

Vidya Vasuki and Trevor Cohen. Reflective Random Indexing for Semiautomatic Indexing of the Biomedical Literature. AMIA 2009, San Francisco, Nov. 14-18, 2009.

Zhichen Xu, Yun Fu, Jianchang Mao, and Difu Su. Towards the Semantic Web: Collaborative Tag Suggestions. In WWW2006: Proceedings of the Collaborative Web Tagging Workshop (2006).

George K. Zipf. (1949) Human Behavior and the Principle of Least-Effort. Addison-Wesley.

Mining Large-scale Comparable Corpora from Chinese-English News Collections

Degen Huang[1] **Lian Zhao**[2] **Lishuang Li**[3] **Haitao Yu**[4]
Department of Computer Science and Technology
Dalian University of Technology
[1]huangdg@dlut.edu.cn [3]lils@dlut.edu.cn
[2]zhaolian@mail.dlut.edu.cn [4]gengshenspirit@163.com

Abstract

In this paper, we explore a CLIR-based approach to construct large-scale Chinese-English comparable corpora, which is valuable for translation knowledge mining. The initial source and target document sets are crawled from news website and standardized uniformly. Keywords are extracted from the source document firstly, and then the extracted keywords are translated and combined as query words through certain criteria to retrieve against the index created using target document set. Meanwhile, the mapping correlations between source and target documents are developed according to the value of similarity calculated by the retrieval tool. Two methods are evaluated to filter the comparable document pairs so as to ensure the quality of the comparable corpora. Experimental results indicate that our approach is effective on the construction of Chinese-English comparable corpora.

1 Introduction

Parallel corpora are key resource for statistical machine translation, in which machine learning techniques are used to learn translation knowledge. Sufficient data is necessary for the data-driven approaches to estimate the model parameters reliably. However, as Munteanu (2006) stated, beyond a few resource-rich language pairs such as English-Chinese or English-French and a small number of contexts like parliamentary debates or legal texts, parallel corpora remain a scarce resource, despite the proposition of automated methods to collect parallel corpora from the Web. Researches on comparable corpora are motivated by the scarcity of parallel corpora. Compared with parallel corpora, comparable corpora are more abundant, up-to-date and accessible.

Comparable corpora are defined as pairs of monolingual corpora selected according to the same set of criteria, but in different languages or language varieties. When creating comparable corpora, the key process is to align the source document with relevant target documents. Early work by Braschler and Scäuble (1998) employed content descriptors and publication dates to align German and Italian news stories. Resnik (1999) mined comparable corpora on the assumption that the pages which are comparable of each other share a similar structure (headers, paragraphs, etc.) when text is presented in many languages in the Web. Tao and Zhai (2005) acquired comparable bilingual text corpora based on the observation that terms that are translations of each other or share the same topic tend to co-occur in the comparable corpora at the same/similar time periods. Recently, Talvensaari et al. (2007) introduced a CLIR-based approach to align two document collections with different languages. All the target documents were indexed with Lemur. Then appropriate keywords were extracted from the source language documents and translated into the target language as query words to retrieve similar target documents.

As we know, the problems may vary with the language of documents when using CLIR-based approach to construct comparable corpora, such as keyword extraction, out-of-vocabulary keyword translation and so on. This paper is a further endeavor to CLIR-based approach for com-

This work was supported by Microsoft Research Asia.

Coling 2008: Poster Volume, pages 472–480,
Beijing, August 2010

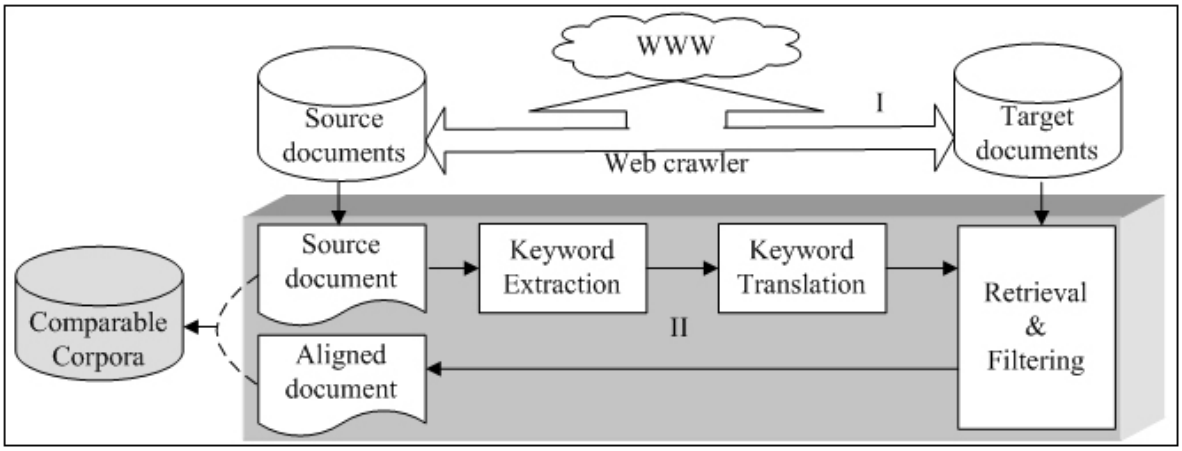


Figure 1. The general architecture of comparable corpora construction

parable corpora construction. We focus on the construction of Chinese-English comparable corpora, explore and address the issues during the construction. Experimental results show that our method is better through a rough comparison with Talvensaari et al. (2007) and it also outperforms our reconstruction of Tao and Zhai (2005) in respect to the quality of comparable corpora.

This paper is organized as follows. In section 2, the general architecture of our system is described, and each module is illuminated in detail. Section 3 reports and analyzes the experimental results followed by conclusions in section 4.

2 System Architecture

Figure 1 shows the general architecture of our comparable corpora construction system. It consists of two components: component I and component II. Component I is mainly composed by a web crawler, which is used to harvest source and target documents from selected web sites. We can get the final source and target document sets through content extraction and noise filtering. The core of the system is component II, which aligns a source document with target documents having comparable contents. It implements on the two document sets generated by component I. Component II is composed of three modules: keyword extraction, keyword translation, and retrieval & filtering. The methods for three modules are detailed respectively.

2.1 Keyword Extraction

A keyword is described as a meaningful and significant expression containing one or more words. Appropriate keywords briefly describe the theme of a document. In this paper, keywords are viewed as basic units of search indexes in order to retrieve closely related documents. Generally, phrases can capture the main idea of a document more effectively, inasmuch as they have more information than single words (an independent linguistic unit after word segmentation for Chinese).

Existing approaches for keyword extraction could be distinguished into two main categories: supervised or unsupervised methods. Supervised machine learning algorithms were widely used in keyword extraction such as Naïve Bayes (Frank et al., 1999; Witten et al., 1999), SVM (Zhang et al., 2006), CRF (Zhang et al., 2008), etc. These approaches had excellent stability. However, it was difficult for us to construct a big-enough golden annotated corpus to train a good classifier, especially for news web pages. Unsupervised methods hinged on evaluating various features to select keywords, such as word frequency (Luhn, 1957), word co-occurrence (Matsuo and Ishizuka, 2004), and TF*IDF (Li et al., 2007). The inherent problem in these methods was that most of their work came in the judgment whether a candidate was a keyword or not, but they had not paid sufficient attention to the identification of phrase candidates. Wan and Xiao (2008) proposed a method for keyphrase extraction from single document. However, it simply combined the adjacent candidate words to a multi-word phrase.

Based on the above observation, our approach for keyword extraction focuses more on the construction of phrasal candidates. It is mainly based on MWE (Multi-Word Expression) extraction together with relevant word ranking method.

MWE is a special lexical unit including compound terms, idioms and collocations, etc. The process of keyword extraction in this paper mainly depends on the following stages.

Stage 1: The generation of phrasal candidates

(1) The extraction of MWEs from the preprocessed document

Document preprocessing is a procedure of morphological analysis including segmentation and part of speech tagging for Chinese. The method based on the marginal probabilities detailed in (Luo and Huang, 2009) is adopted in this part.

We extract MWEs using LocalMaxs selection algorithm together with a relevance measure calculation method (FSCP) proposed by Silva et al. (1999). Suffix arrays and related structures in (Aires et al., 2008) are used to compute the FSCP value so as to raise efficiency. And the initial collection of MWEs named *G* for the document is generated after filtered by stopword list.

(2) The acquisition of new MWEs through the modification for segmentation

As a matter of fact, the results of segmentation for the document usually have some errors especially for out-of-vocabulary (OOV) words which are segmented to single Chinese characters in most cases. Inaccurate segmentation leads to some faults for keyword extraction. As stated in (Liu et al., 2007), OOV words can be identified by the method of MWEs extraction mentioned above. Therefore, we modify the segmentation like this: any MWE in *G* is merged to one word if it only consists of single Chinese characters and its frequency > *freq*. The changes before and after merging are shown in Table 1. Because the method of MWE extraction is based on statistical techniques, so low frequency of MWE will result in poor performance. But large value for *freq* means that very few MEWs can satisfy the frequency restriction. In our experiments, we set *freq*=2. The extraction process is called again to identify MWEs from the document with modified segmentation. Consequently, new collection of MWEs is acquired.

Additionally, some simple rules are defined according to language features to filter MWEs. In this paper, our method is tailored to extract keywords from news web pages which contain some special symmetric marks like "「, 」". The words in a specially marked area are usually important to the document. So we extract words within each paired marks and view them as a MWE on the condition that it contains two or more than two words. All of the MWEs are viewed as phrasal candidates and filtered by stopword list.

Stage 2: The generation of single words candidates

Our method also generates single word candidates with the account that both phrase and single word can be served as a keyword. The process of single word selection is independent of MWE extraction. The candidate words are restricted to nouns, verbs, strings (like WTO) and merged words as discussed in the previous stage. But the word will be removed if it only appears once in the document or is contained in the stopword list.

Stage 3: Keyword selection based on candidates ranking

As for MWE candidates, we calculate the weight for them using Formula 1 which refers to the formula used to sort NP phrases in (Bracewell et al., 2008). But the weight of *len* is reduced.

$$Weight(MWE) = \log(\sqrt{len} + f_{MWE} + \frac{1}{len} \times \sum_{i=1}^{len} tf(w_i)) \quad (1)$$

Where *len* is the length of MWE (in number of words); f_{MWE} is the frequency of the MWE within in the document; $tf(w_i)$ is the frequency of word w_i. The following rules are used to rank MWEs:

MWE	Segmentation before merging	Segmentation after merging	Pos before merging	Pos after merging
布 卡	鸟/ 人/ 布/ 卡/ 为/ 脚伤/ 所/ 苦/	鸟/ 人/ 布卡/ 为/ 脚伤/ 所/ 苦/	鸟/n 人/n 布/n 卡/n 为/vl 脚伤/n 所/us 苦/a	鸟/n 人/n 布卡/oov 为/vl 脚伤/n 所/us 苦/a
琼 丝	琼/ 丝/ 五金/ 梦/	琼丝/ 五金/ 梦/	琼/jb 丝/n 五金/b 梦/n	琼丝/oov 五金/b 梦/n

Table 1. Changes before and after merging

(a) more frequent MWEs are ranked higher; (b) MWEs with larger weight are ranked higher. In order to avoid redundancy, we remove the redundant MWEs with lower rank.

Single word candidates are ranked as follows: (a) the single word *w* with larger TF*IDF value is ranked higher; (b) the pos score for *w* in descending order is: named entity, merged words, nouns, strings, verbs. In the end, top-*a* MWEs and top-*b* single words are chosen to form the keyword set of the document.

Stage 4: Parameters evaluation and experimental results

The max number of keywords extracted from each document is limited to ten (a+b=10) and we run our approach on the dataset which include one hundred Chinese documents from the corpus of NTCIR-5 since they are also news articles. For evaluation of the results, the keywords extracted by our method are compared with the manually extracted keywords (at most ten keywords are assigned to each document). The F-measure is used as evaluation metric. It is defined like this: $F=(P+R)/2$; $P=num_{match}/num_{system}$; $R=num_{match}/num_{manual}$. Where num_{match} is the count of keywords extracted by our method matching with manually extracted keywords; num_{system} is the count of keywords extracted by our method; num_{manual} is the count of keywords assigned by human.

Figure 2 shows the performance curves for our extraction method. In this figure, *a* ranges from 0 to 10 while *b* is 10 to 0. It performs best when *a* = 4 and *b* = 6. So the two values are adopted in this paper.

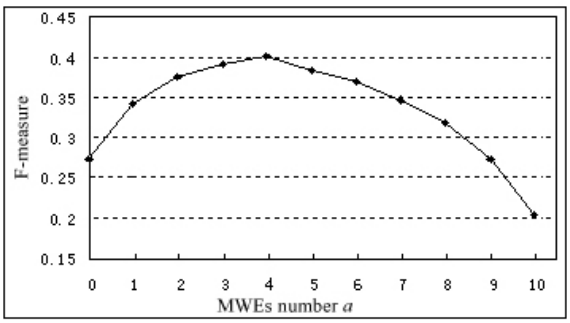


Figure 2. F-measure varies with the value of *a*

We test our approach on another dataset which also contains one hundred documents. In the experiments, the max number of keywords is set to ten. Table 2 shows the results of keyword extraction under three different conditions respectively. (A) Only extracts single words as keywords while just MWEs with (B). (C) The method presented in this paper which makes a proper combination of MWEs and single words.

	P	R	F
A (single words)	24.2%	28.5%	26.4%
B (MWEs)	18.1%	23.0%	20.6%
C (A+B)	34.2%	43.6%	38.9%

Table 2. Keyword extraction results

2.2 Keyword Translation

As for keyword translation, there are three main approaches: translation based on dictionary, parallel corpora and machine translation. Dictionary based approach is adopted in our system by taking the acquisition of translation resource into account.

Word Sense Disambiguation (WSD) and OOV problem are the main difficulties in CLIR (Cross Language Information Retrieval) task. A typical bilingual dictionary will provide a set of alternative translations for a given keyword, so how to choose the optimal translation is called Word Sense Disambiguation. Actually some keywords can not be found and translated due to the coverage limitation of a bilingual dictionary, which is called OOV problem.

In this paper, the keyword is given up if its size of translations gained from the bilingual dictionary is larger than two for the convenience of WSD. Additionally, both of the translations are treated as synonyms and equal weight is assigned to them when retrieval.

To address the OOV problem, researchers proposed methods using snippets returned by a search engine. For example, Wang et al. (2004) introduced a statistics-based approach called SCPCD to mine translations from the returned snippets. Different from (Wang et al., 2004), Zhou et al. (2007) used a pattern-based approach to analyze the mixed-languages snippets.

Leveraging on previous work, we analyze the co-occurrence mode of the OOV term and the corresponding translation in the returned snippets. Table 3 shows the typical co-occurrence modes collected during experiments, where the English words in bold are the corresponding translations of the underlined Chinese OOV terms. From Table 3, we can see the translations in number 1, 2 and 3 are included in the symmetric symbols, like bracket, quotation marks. However, the

Serial number	Segments extracted from the returned snippets
1	…原版英语论坛书名：廊桥遗梦《**The Bridges of Madison County**》作者：美国…
2	…英文影评：廊桥遗梦（**The Bridges of Madison County**）-52 影评网...
3	…具有布什特色的“牛仔外交”（**cowboy diplomacy**）反而被“现实主义”取代…
4	...用于数据挖掘的贝叶斯网络 **Bayesian Network** for Data Mining-作者：慕...
5	…以《布什牛仔外交终结》（The End of **Cowboy Diplomacy**）为题作封面故事…
6	…廊桥遗梦隐藏摘要. **The Bridges of Madison County**. Forrest Gump 阿甘正传...

Table 3. Chinese OOV and the corresponding translation in returned snippets

translations in number 4, 5, and 6 are embedded in the partial sentence while there are noise English words. In order to get the correct translation, the partial sentence needs to be segmented. By above analysis, we integrate the SCPCD method and the pattern-based method so as to extract more correct translations. The SCPCD method can be used to determine the boundaries for OOVs like number 4, 5, and 6; while pattern-based method makes use of the symmetric symbols like number 1, 2 and 3. Table 4 shows the experimental results for OOV translation methods. The average top-n inclusion rate is adopted as a metric. For a set of test OOV terms, its top-n inclusion rate is defined as the percentage of the OOVs whose translations can be found in the first n extracted translations.

	Pattern	SCPCD	Pattern + SCPCD
Top-1	40.0%	49.2%	68.1%
Top-3	41.5%	55.4%	70.2%

Table 4. The performance comparison of different OOV translation methods

The test dataset used is the Chinese topic terms in CLIR task of NTCIR-5. The search engine is Google. The bilingual dictionary used by us is LDC_CE_DICT 2.0. And we only adapt the pattern with symmetric symbols, which has the highest precision proposed by Cao et al. (2007).

2.3 Retrieval and Filtering

The process of retrieval is to construct the alignment relationship between source and target document pairs. It is a core module in our system since the quality of comparable corpora is greatly influenced by alignment level which depends on the relevance between document pairs. Our intention here is to retrieve high relevant target documents for the source documents. Open-source toolkit Indri is introduced to assist the retrieval process. Indri is a part of the Lemur project[1]. On the basis of Lemur, it combines inference networks with language modeling. And it's widely adopted by institution for scientific research since it is effective, flexible, usable and powerful. So it is employed by us to retrieve related documents. A query for each source document is formed by the translated keywords with Indri query language and then run against the target collection.

The essential of alignment is to compare the similarity between source and target document pairs. In order to reduce the workload of comparing, Pooling method is applied to assist the comparing process. We choose the top r documents returned by Indri retrieval system to build the related document pool. And g ($g<=r$) documents in the pool are selected to form the alignment document pairs together with the source document. In our experiments, we set r=10 and g=1.

In the process of alignment, three features are used to filter the alignment pairs for the sake of pruning the low relevant pairs. The first is publication date contained in documents. The second is similarity calculated by Indri between the query and the target document when retrieval. The last is KSD (Keyword similarity between document pairs) which is defined by our system. In this paper, we propose two methods to filter the alignment pairs by using various features.

(1) DSF filtering

This method depends on two features: date and similarity. At first, we give a priority to the target documents that have the closest date to the source document during the top-r documents searching. A date-window size d is defined to measure the date difference. We set d=1 in this paper. That is to say, the target documents with

[1] Lemur toolkit is developed by Carnegie Mellon University and University of Massachusetts. The open source code is available at http://www.lemurproject.org.

exactly the same date as the source document, and one day earlier or later are considered to be closest. Then, we select *g* documents with larger similarity from the related document pool. Finally, we rank all of the alignment pairs with the score of similarity and set a similarity threshold *s* to filter further. It should be noted that there are $n \cdot g$ alignment pairs, where *n* is the number of source documents having non-empty related document pool.

(2) DSKF filtering

This method utilizes all of the features: date, similarity and KSD. As for KSD, it integrates two factors. One is NTK, namely the number of translated keywords appeared in the target document, since the target document is more similar to the source document as increasing of NTK. The other is FIS, namely frequency information score. Inspired by paper (Tao and Zhai, 2005), we use the score of FIS to measure the correlations between the keywords in source document and translated keywords in target document which represent the matching for source and target document pair. We define d_s as the source document, d_t as the target document, *ks* as the set of keywords extracted from d_s, *kts* as the set of translated keywords. Formula 2 is used to compute the score of FIS:

$$Score_{FIS} = \sum_{i=1}^{ktsLen} (BM25(x_i, d_s) \cdot IDF(x_i) \cdot BM25(y_i, d_t) \cdot IDF(y_i) / norm(Dif(x_i, y_i))) \quad (2)$$

Where, *ktsLen* is the size of *kts*, y_i is an element in *kts*, x_i is the element in *ks* while y_i is the translation of x_i. Moreover, *BM25*(*w*, *d*) is the normalized frequency of word *w* in document *d*. It has been considered as one of the most effective matching functions for retrieval. *IDF* stands for Inverse Document Frequency which is also commonly used in information retrieval. *Dif*(*x, y*) is defined as the difference between *BM25*(*x*, d_s) and *BM25*(*y*, d_t). Formula 2 penalizes large difference due to the conditions like this: any keyword in source document appears many times while its translation appears rarely in target document. The process of its normalization is run by Formula 3 which makes the score less sensitive to the absolute value:

$$norm(score) = \begin{cases} 1, & score < 1 \\ \sqrt{score}, & else \end{cases} \quad (3)$$

Furthermore, the final KSD score is got by simply adding the normalized scores of NTK and FIS which are dealt with Formula 3. Actually, the two filtering methods differ principally in the last step. DSKF sorts all of the alignment pairs according to the KSD score while it is similarity in DSF. We also set a KSD threshold *k* for DSKF method to filter further. The values for *s* and *k* will be investigated in the following experiments.

3 Experiments

In this section, we first introduce how to acquire the source and target document sets. Then our system is tested on the two sets. The experimental results are reported and analyzed finally.

3.1 Experiment Setup

To test the effectiveness of the proposed system, large-scale of Chinese and English news web pages are crawled respectively from XinHuaNet and used as the document resource. The reasons for choosing news pages are:

(1) Many websites, like portal website, news agency, government and so on, provide large-scale news reports. At the same time, a large proportion of the reports can be crawled politely, so document acquisition is relatively easy.

(2) The news pages include various contents, such as politics, economy, sports, so the corpora made up of news pages can avoid the limitations of domain-specific corpora.

All the news pages are processed uniformly. The core content of each web page crawled is extracted and several tags describing the headline and publication date are added. Meanwhile, the original contents are kept with no change. Table 5 shows the basic information of document sets.

Year	Number of source documents	Number of target documents
2003	23747	3390
2004	25660	2943
2005	47333	11578
2006	28572	25320
2007	25036	25247
2008	14021	24292
2009	7476	10887
Total	171845	103657

Table 5. The composition of source document set and target document set

3.2 Results and Discussion

The quality of comparable corpora highly de-

pends on the alignment level between source and target document pairs. Braschler and Scäuble (1998) used five levels of relevance to assess the alignments as follows:

(1) Same story. The two documents deal with the same event.

(2) Related story. The two documents deal with the same event or topic from a slightly different viewpoint. Alternatively, the other document may concern the same event or topic, but the topic is only a part of a broader story or the article is comprised of multiple stories.

(3) Shared aspect. The documents deal with related events. They may share locations or persons.

(4) Common terminology. The events or topics are not directly related, but the documents share a considerable amount of terminology.

(5) Unrelated. The similarities between the documents are slight or nonexistent.

We randomly select 500 source documents published in 2009 as the test dataset. Experiments with different parameters are constructed based on this dataset. The quality of each alignment pair is manually assessed using the five-level relevance as discussed above. What should to be pointed out is that parameter s and k are not absolute values, but percentile rank level in our work. For instance, k = 10 means that we only choose the alignment pairs whose KSD score rank in top ten percent among all of the results.

Table 6 shows the results filtered by DSF method with different values of s ($s_1 < s_2 < s_3 < s_4$). Table 7 shows the results filtered by DSKF method with various values of k ($k_1 < k_2 < k_3 < k_4$). In order to evaluate the results conveniently, two standards are established: (a) the number of high relevant pairs created, which is the count of document pairs in Level 1 and 2; (b) the quality of the whole alignments, that is to say the percentage of alignment pairs with Level 1 and 2. Seen from Table 6 and 7, DSKF is better than DSF by considering the two standards. Compared with DSF, more high relevant pairs are left filtered by DSKF when they have the same total number of pairs. In other words, the DSKF method is more powerful to make high relevant pairs in higher rank so as to reduce alignment pairs which are rarely relevant. Therefore, DSKF is adopted in our system. Taking the first criterion into account, we give up the parameter k_1, k_2. Parameter k_4 is not the best considering the second criterion. Ultimately, k_3 is chosen as the final value for k. At this point, the number of alignment pairs in Level 1 and 2 is close to the maximum. Meanwhile, the percentage of high alignments reaches 68.5%.

Among the surveyed related work, Talvensaari et al. (2007) created Swedish-English comparable corpora based on CLIR techniques and its framework of construction is similar to ours. However, the two systems are different in the following aspects:

Level	s_1 = 10		s_2 = 30		s_3 =50		s_4 =70	
	Number	%	Number	%	Number	%	Number	%
Leve1 1	23	46.9%	54	36.5%	83	33.5%	96	27.7%
Level 2	18	36.7%	43	29.1%	62	25.0%	81	23.3%
Level 3	4	8.2%	21	14.2%	40	16.1%	57	16.4%
Level 4	4	8.2%	19	12.8%	41	16.5%	60	17.3%
Level 5	0	0.0%	11	7.4%	22	8.9%	53	15.3%
Total	49	100%	148	100%	248	100%	347	100%

Table 6. The distribution results filtered by DSF with different s parameters

Level	k_1 = 10		k_2 = 30		k_3 = 50		k_4 =70	
	Number	%	Number	%	Number	%	Number	%
Level 1	33	67.3%	78	52.7%	93	37.5%	98	28.2%
Level 2	15	30.6%	52	35.1%	77	31.0%	89	25.6%
Level 3	1	2.0%	9	6.1%	37	14.9%	62	17.9%
Level 4	0	0.0%	9	6.1%	34	13.7%	60	17.3%
Level 5	0	0.0%	0	0.0%	7	2.8%	38	11.0%
Total	49	100%	148	100%	248	100%	347	100%

Table 7. The distribution results filtered by DSKF with different k parameters

(1) The language is different. We focus on building comparable corpora of Chinese-English while they were Swedish-English.

(2) A series of sub problems are different due to language difference. As for keyword extraction, we propose a method to select both key phrases and single words, while they used RATF (Relative Average Term Frequency) method. For OOV problem, we combine the SCPCD method with the pattern-based method to extract OOV translations from snippets returned by a search engine. However, the classified s-gram matching technique was utilized by Talvensaari et al. (2007) to translate OOV words.

(3) Talvensaari et al. (2007) filtered their alignment pairs mainly depending on date and similarity, while we introduce new feature KSD to extend the original feature set.

Talvensaari et al. (2007) also randomly chose 500 source documents and assessed the quality of alignments using the same five-level relevance.

In addition to this, we implement the method of Tao and Zhai (2005) which is a purely statistical-based and language independent approach. The source and target documents published in 2009 are employed to test the method. The same sample as our system including 500 Chinese documents is chosen to make a further comparison with our work. We align each source document with one target document through the BM25Corr model in (Tao and Zhai, 2005). The alignment pairs are ranked according to mapping scores calculated by the BM25Corr model. And we select the top N (N = 248) alignment pairs for the benefit of comparison.

Table 8 shows the distribution results for the three systems. As illustrated in Table 8, we can roughly conclude that our approach creates more alignment pairs with the same number of source documents when compared with Talvensaari et al. (2007). Meanwhile, the percentage of high relevant document pairs is larger.

Likewise, our system outperforms BM25Corr in that it aligns more high relevant documents pairs when they use the same sample of test corpora and create the same total number of pairs. Obviously, the quality of comparable corpora gained by our system is better than BM25Corr.

All the experimental results and analysis mentioned above indicate that our method is effective to create alignment pairs. Up to now, both the source and target documents published in 2007-2009 years are used to build comparable corpora through our proposed system. It includes 23102 alignment pairs after filtered by DSKF.

Level	Talvensaari et al. (2007)		Our System (DSKF filtering)		BM25Corr (Top N = 248)	
	Number	%	Number	%	Number	%
Level 1	21	21.6%	93	37.5%	1	0.4%
Level 2	20	20.6%	77	31.0%	2	0.8%
Level 3	33	34.0%	37	14.9%	3	1.2%
Level 4	19	19.6%	34	13.7%	5	2.0%
Level 5	4	4.1%	7	2.8%	237	95.6%
Total	97	100%	248	100%	248	100%

Table 8. The distribution results for Talvensaari et al. (2007), Our System, and BM25Corr

4 Conclusions

In this paper, we propose a CLIR-based approach to create large-scale Chinese-English comparable corpora. Firstly, we harvest the original source and target document sets from news website using open-source crawler. Then the core content of each document is extracted through discriminating noise contents. Next, we delve into the approaches of problems such as keyword extraction and OOV translation followed by the process of retrieval to develop mapping correlations between source and target documents. Finally, three features as publication date, similarity score and KSD value are used to filter the aligned document pairs. Experimental results show that our approach is effective to mine Chinese-English document pairs with comparable contents. In the future, we will optimize the approach for every module in the construction of comparable corpora for the sake of improving the performance of the whole system. What's more, it will be worth consideration to mine mappings between terms which can be served as a feature for the process of developing mappings between document pairs in turn.

References

Aires, José, Gabriel Lopes, and Joaquim Ferreira Silva. 2008. Efficient Multi-word Expressions Extractor Using Suffix Arrays and Related Structures. In *Proceeding of the 2nd ACM workshop on Improving non english web searching*, pp. 1-8.

Bracewell, David B., Fuji Ren, and Shingo Kuroiwa. 2008. Mining News Sites to Create Special Domain News Collections. *International Journal of Computational Intelligence*, 4(1): 56-63.

Braschler, Martin, and Peter Scäuble. 1998. Multilingual Information Retrieval Based on Document Alignment Techniques. In *Proceedings of the 2nd European Conference on Research and Advanced Technology for Digital Libraries*, pp. 183-197.

Cao Guihong, Jianfeng Gao, and Jianyun Nie. A System to Mine Large-Scale Bilingual Dictionaries from Monolingual Web pages. 2007. In *Proceedings of Machine Translation Summit XI*, pp. 57-64.

Frank, Eibe, Gordon W. Paynter, and Ian H. Witten. 1999. Domain-Specific Keyphrase Extraction. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence*, pp. 668-673.

Li Juanzi, Qi'na Fan, and Kuo Zhang. 2007. Keyword Extraction Based on tf/idf for Chinese News Document. *Wuhan University Journal of Natural Sciences*, 12(5): 917-921.

Liu Tao, Bingquan Liu, Xiaolong Wang, and Minghui Li. 2007. The Effectiveness Study of Local Maximum Feature for Chinese Unknown Word Identification. *Journal of Chinese Language and Computing*, 17(1): 15-26.

Luhn, Hans Peter. 1957. A Statistical Approach to Mechanized Encoding and Searching of Literary Information. *IBM Journal of Research and Development*, 1(4): 309-317.

Luo Yanyan, and Degen Huang. 2009. Chinese Word Segmentation Based on the Marginal Probabilities Generated by CRFs. *Journal of Chinese Information Processing*, 23(5): 3-8.

Matsuo, Yutaka and Mitsuru Ishizuka. 2004. Keyword Extraction from a Single Document Using Word Co-occurrence Statistical Information. *International Journal on Artificial Intelligence Tools*, 13(1): 157-169.

Munteanu, Dragos Stefan. 2006. Exploiting Comparable Corpora. *Doctoral Thesis*. UMI Order No.3257825. University of Southern California.

Resnik, Philip. 1999. Mining the web for bilingual text. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pp. 527-534.

Silva, Joaquim Ferreira, Gaël Dias, Sylvie Guilloré, and José Gabriel Pereira Lopes. 1999. Using LocalMaxs Algorithm for the Extraction of Contiguous and Non-contiguous Multiword Lexical Units. In *Proceedings of the 9th Portuguese Conference on Artificial Intelligence*, pp. 113-132.

Talvensaari, Tuomas, Jorma Laurikkala, Kalervo Järvelin, Martti Juhola and Heikki Keskustalo. 2007. Creating and Exploiting a Comparable Corpus in Cross-Language Information Retrieval. *ACM Transactions on Information Systems*, 25(1):1-21.

Tao Tao, and Chengxiang Zhai. 2005. Mining Comparable Bilingual Text Corpora for Cross-Language Information Integration. In *Proceedings of the 11th ACM SIGKDD international conference on Knowledge discovery in data mining*, pp. 691-696.

Wan Xiaojun, and Jianguo Xiao. 2008. CollabRank: Towards a Collaborative Approach to Single-Document Keyphrase Extraction. In *Proceeding of the 22nd International Conference on Computational Linguistics*, pp. 969-976.

Wang Jenq Haur, Jie Wen Teng, Pu Jen Cheng, Wen Hsiang Lu, and Lee Feng Chien. 2004. Translating Unknown Cross-Lingual Queries in Digital Libraries using a Web-based Approach. In *Proceedings of the 4th ACM/IEEE-CS joint Conference on Digital Libraries*, pp. 108-116.

Witten, Ian H., Gordon W. Paynter, Eibe Frank, Carl Gutwin, and Craig G. Nevill-Manning. 1999. KEA: Practical automatic keyphrase extraction. In *Proceedings of the 4th ACM Conference on Digital Libraries*, pp. 254-255.

Zhang Chengzhi, Huilin Wang, Yao Liu, Dan Wu, Yi Liao, and Bo Wang. 2008. Automatic Keyword Extraction from Documents Using Conditional Random Fields. *Journal of Computational Information Systems*, 4(3): 1169-1180.

Zhang Kuo, Hui Xu, Jie Tang, and Juanzi Li. 2006. Keyword Extraction Using Support Vector Machines. In *Proceedings of the 7th International Conference on Web-Age Information Management*, pp. 85-96.

Zhou Dong, Mark Truran, Tim Brailsford, and Helen Ashman. 2007. NTCIR-6 Experiments Using Pattern Matched Translation Extraction. In *Proceedings of 6th NTCIR Workshop Meeting*, pp. 145-151.

Bilingual lexicon extraction from comparable corpora using in-domain terms

Azniah Ismail
Department of Computer Science
University of York
azniah@cs.york.ac.uk

Suresh Manandhar
Department of Computer Science
University of York
suresh@cs.york.ac.uk

Abstract

Many existing methods for bilingual lexicon learning from comparable corpora are based on similarity of context vectors. These methods suffer from noisy vectors that greatly affect their accuracy. We introduce a method for filtering this noise allowing highly accurate learning of bilingual lexicons. Our method is based on the notion of *in-domain terms* which can be thought of as the most important contextually relevant words. We provide a method for identifying such terms. Our evaluation shows that the proposed method can learn highly accurate bilingual lexicons without using orthographic features or a large initial seed dictionary. In addition, we also introduce a method for measuring the similarity between two words in different languages without requiring any initial dictionary.

1 Introduction

In bilingual lexicon extraction, the context-based approach introduced by Rapp (1995) is widely used (Fung, 1995; Diab and Finch, 2000; among others). The focus has been on learning from comparable corpora since the late 1990s (Rapp, 1999; Koehn and Knight, 2002; among others). However, so far, the accuracy of bilingual lexicon extraction using comparable corpora is quite poor especially when orthographic features are not used. Moreover, when orthographic features are not used, a large initial seed dictionary is essential in order to acquire higher accuracy lexicon (Koehn and Knight, 2002). This means that current methods are not suitable when the language pairs are not closely related or when a large initial seed dictionary is unavailable.

When learning from comparable corpora, a large initial seed dictionary does not necessarily guarantee higher accuracy since the source and target texts are poorly correlated. Thus, inducing highly accurate bilingual lexicon from comparable corpora has so far been an open problem.

In this paper, we present a method that is able to improve the accuracy significantly without requiring a large initial bilingual dictionary. Our approach is based on utilising *highly associated terms* in the context vector of a source word. For example, the source word *powers* is highly associated with the context word *delegation*. We note that, firstly, both share context terms such as *parliament* and *affairs*. And, secondly, the translation equivalents of *powers* and *delegation* in the target language are not only highly associated but they also share context terms that are the translation equivalents of *parliament* and *affairs* (see Figure 1).

2 Related work

Most of the early work in bilingual lexicon extraction employ an initial seed dictionary. A large bilingual lexicon with 10k to 20k entries is necessary (Fung, 1995; Rapp, 1999).

Koehn and Knight (2002) introduce techniques for constructing the initial seed dictionary automatically. Their method is based on using identical spelling features. The accuracy of such initial bilingual lexicon is almost 90.0 percent and can be increased by restricting the word length (Koehn and Knight, 2002). Koehn and Knight found approximately 1000 identical words in their German

Coling 2008: Poster Volume, pages 481–489,
Beijing, August 2010

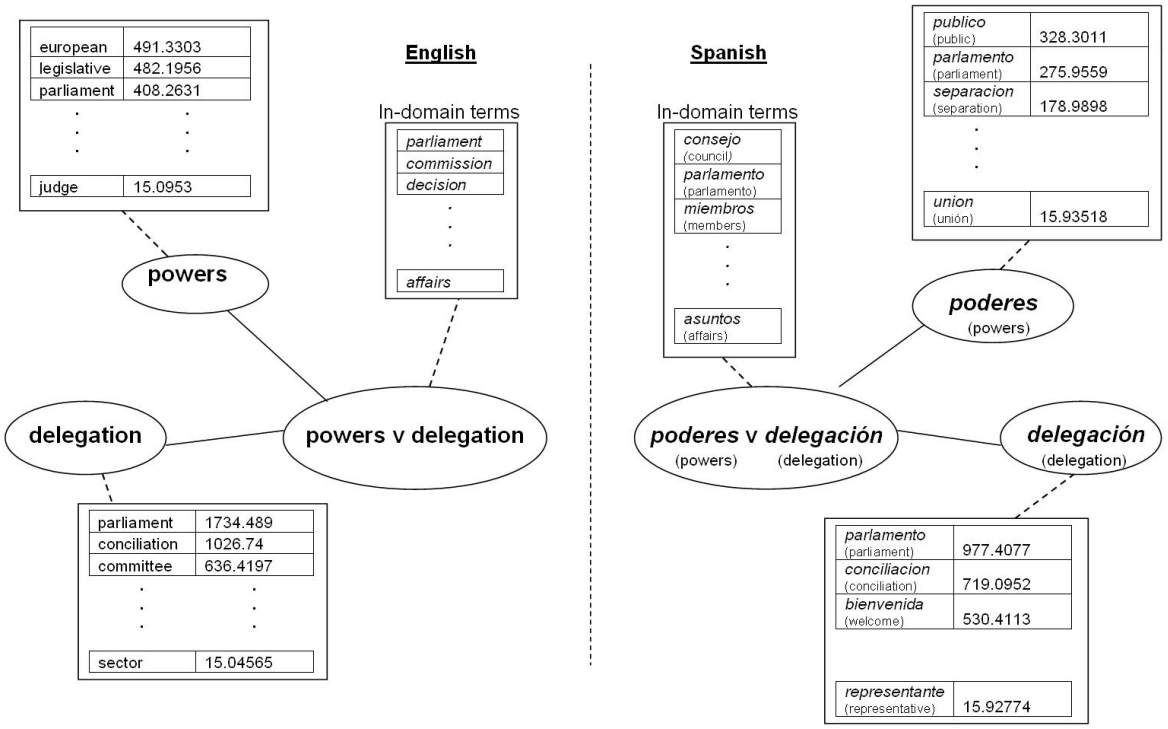


Figure 1: An example of in-domain terms that co-occur in English and Spanish. The source word is *powers* and the target word is *poderes*. The word *delegation* and *delegacion* are the highly associated words with the source word and the target word respectively. Their in-domain terms, as shown in the middle, can be used to map the source word in context of word *delegation* to its corresponding target word in context of *delegacion*.

and English monolingual corpora. They expanded the lexicon with the standard context-based approach and achieved about 25.0 percent accuracy (Koehn and Knight, 2002).

Similar techniques were used in Haghighi et al. (2008) who employ *dimension reduction* in the extraction method. They recorded 58.0 percent as their best F_1 score for the context vector approach on non-parallel comparable corpora containing *Wikipedia* articles. However, their method scores less on comparable corpora containing distinct sentences derived from the *Europarl English-Spanish* corpus.

3 Learning in-domain terms

In the standard context vector approach, we associate each source word and target word with their context vectors. The source and target context vectors are then compared using the initial seed dictionary and a similarity measure. Learning from comparable corpora is particularly problematic due to data sparsity, as important context terms may not occur in the training corpora while some may occur but with low frequency and can be missed. Some limitations may also be due to the size of the initial seed dictionary being small.

The initial seed dictionary can also contribute irrelevant or less relevant features that can mislead the similarity measure especially when the number of dimensions is large. The approach we adopt attempts to overcome this problem.

In Figure 1, for the source word *powers*, *delegation* is the highly associated word. Both *powers* and *delegation* share common contextual terms such as *parliament* and *affairs*. Now the translation equivalent of *delegation* is *delegacion*. For the potential translation equivalent *poderes*, we see that the common contextual terms shared by *powers* and *poderes* are terms *parlamento* (*parliament*) and *asuntos* (*affairs*).

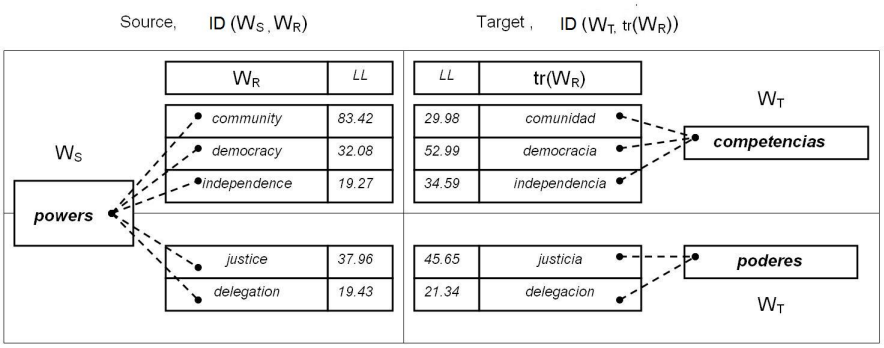


Figure 2: An example of English-Spanish lexicon learnt for the source word *powers*. On the top, the system suggested *competencias* and rejected *poderes* when *powers* is associated with *community*, *democracy* or *independence*. The word *poderes* is suggested when *powers* is associated with *justice* or *delegation*.

We observe that these common contextual terms are simultaneously the *first-order* and *second-order* context terms of the target word. They are the *shared* context terms of the target word and its highly associated context term. We define these terms as *in-domain terms*. These in-domain terms can be used to map words to their corresponding translations. The highly associated context terms can be thought of as sense discriminators that differentiate the different uses of the target word. In Figure 2, we show how *delegation* helps in selecting between the "control or influence" sense of *powers* while rejecting the "ability or skill" sense.

In this paper, our focus is not on sense disambiguation and we follow current evaluation methods for bilingual lexicon extraction. However, it is clear that our method can be adapted for building sense disambiguated bilingual lexicons.

3.1 Identifying highly associated words

To identify the context terms $CT(W_S)$ of a source word W_S, as in (Rapp, 1999), we use *log-likelihood ratio* (LL) Dunning (1993). We choose all words with $LL > t_1$ where t_1 is a threshold.

The *highly associated words* then are the top k highest ranked context terms. In our experiments, we only choose the top 100 highest ranked context terms as our highly associated terms.

In order to compute the log-likelihood ratio of target word a to co-occur with context word b, we create a contingency table. The contingency table contains the observed values taken from a given corpus. An example of the contingency table is shown in Table 1.

C[i,j]	***community***	***¬ community***		
powers	124	1831	**1955**	*C(powers)*
¬ powers	11779	460218	**471997**	*C(¬ powers)*
	11903	**462049**		
	C(community)	*C(¬ community)*		

Here $C[i, j]$ denotes the count of the number of sentences in which i co-occurs with j.
Total corpus size: N = 473952 in the above

Table 1: Contingency table for observed values of target word *powers* and context word *community*.

The LL value of a target word a and context word b is given by:

$$LL(a, b) =$$

$$\sum_{i \in \{a, \neg a\}, j \in \{b, \neg b\}} 2C(i, j) \log \frac{C(i, j)N}{C(i)\, C(j)}$$

3.1.1 Identifying in-domain terms

In our work, to find the translation equivalent of a source word W_S, we do not use the context terms $CT(W_S)$. Instead, we use the *in-domain terms* $IDT(W_S, W_R)$. For each highly associated term

W_R, we get different in-domain terms. Furthermore, $IDT(W_S, W_R)$ is a subset of $CT(W_S)$.

The in-domain terms of W_S given the context terms W_R is given by:

$$ID(W_S, W_R) = CT(W_S) \cap CT(W_R)$$

Programme and *public* are some of the examples of in-domain terms of *powers* given *community* as the highly associated term.

3.1.2 Finding translations pairs

Note that $ID(W_S, W_R)$ is an in-domain term vector in the source language. Let W_T be a potential translation equivalent for W_S. Let, $tr(W_R)$ be a translation equivalent for W_R. Let $ID(W_T, tr(W_R))$ be an in-domain term vector in the target language.

We use $tr(W_S|W_R)$ to denote the translation proposed for W_S given the highly associated term W_R. We compute $tr(W_S|W_R)$ using:

$$tr(W_S|W_R) = \underset{W_T}{\operatorname{argmax}}\, sim(ID(W_S, W_R), ID(W_T, tr(W_R)))$$

Our method learns translation pairs that are conditioned on highly associated words (W_R). Table 2 provides a sample of English-Spanish lexicon learnt for the word *power* with different W_R.

English		Spanish		Sim
W_S	W_R	$tr(W_R)$	W_T	
powers	community	comunidad	**competencias**	**0.9876**
			poderes	0.9744
			independiente	0.9501
	democracy	democracia	**competencias**	**0.9948**
			poderes	0.9915
			independiente	0.9483
	independence	independencia	**competencias**	**0.9939**
			poderes	0.9745
			independiente	0.9633
	justice	justicia	**poderes**	**0.9922**
			competencias	0.3450
			independiente	0.9296
	delegation	delegacion	**poderes**	**0.9568**
			competencias	0.9266
			independiente	0.8408

Table 2: A sample of translation equivalents learnt for *powers*.

In the next section, we introduce a similarity measure that operates on the context vectors in the source language and the target language without requiring a seed dictionary.

4 Rank-binning similarity measure

Most existing methods for computing similarity cannot be directly employed for measuring the similarity between in-domain term context vectors since each context vector is in a different language. A bilingual dictionary can be assumed but that greatly diminishes the practicality of the method.

We address this by making an assumption. We assume that the relative distributions of in-domain context terms of translation equivalent pairs are roughly comparable in the source language and in the target language. For example, consider the log-likelihood values of the in-domain terms for the translation pair *agreement-acuerdo* (conditioned on the highly associated term *association-associacion*) given in Figure 3. We note that the distribution of in-domain terms are comparable although not identical. Thus, the distribution can be used as a clue to derive translation pairs but we need a method to compute similarity of the vector of in-domain terms.

Rank-binning or rank histograms are usually used as a diagnostic tool to evaluate the spread of an ensemble rather than as a verification method. Wong (2009) use the method of rank-binning to roughly examine performance of a system on learning lightweight ontologies. We apply the rank-binning procedure for measuring the similarity of word pairs.

Pre-processing step:

1. Let W_S be a source language word and $x_1, x_2, ..., x_n$ be the set of n context terms ranked in descending log-likelihood values of W_S (see Table 3).

2. We transform the rank values of context terms x_k into the range [0,1] using:

$$z_k = \frac{rank(x_k) - 1}{n - 1}$$

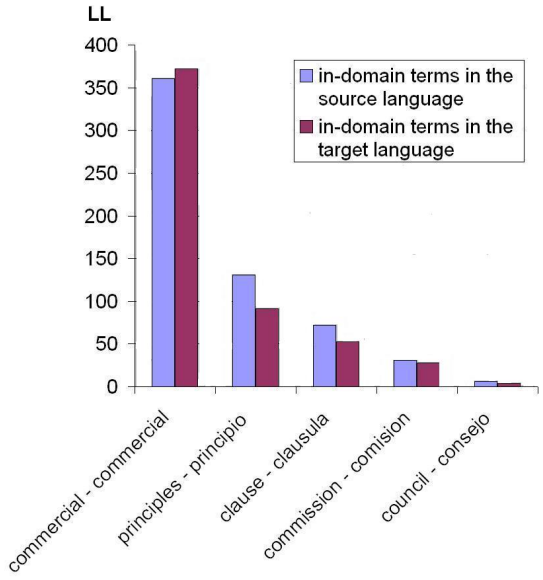


Figure 3: Similar distribution of in-domain terms for *agreement* with *association* and *acuerdo* with *asociacion*.

Binning procedure

We divide the interval $[0, 1]$ into g bins[1] of equal length. Let $b_1, \ldots, b_g$ denote the g bins. Then we map the in-domain terms vector $ID(W_S, W_R)$ into the binned vector $b_1, \ldots, b_g$. For each $x_k \in ID(W_S, W_R)$, this mapping is done by using the corresponding z_k from the pre-processing step. For each bin, we count the number of different in-domain terms that are mapped into this bin. Thus, if the range of the first bin b_1 is [0, 0.009] then *european, legislative, parliament* are mapped into b_1 i.e. $b_1 = 3$. The bins are normalised by dividing with $\mid ID(W_S, W_R) \mid$.

Rank binning similarity

We use Euclidean distance to compute similarity between bins. Given, bins $P = p_1, \ldots, p_g$ and $Q = q_1, \ldots, q_g$, the Euclidean distance is given by:

$$dist(P, Q) = \sqrt{\sum_{i=1}^{g} (p_i, q_i)^2}$$

[1]We used the following formula to estimate the number of bins:

$$g = 1 + 3.3 * log(\mid ID(W_S, W_R) \mid)$$

$CT(\textbf{\textit{powers}})$			
Context term	LL	$rank$	z_k
european	491.33	1	0.00000
legislative	482.19	2	0.00406
parliament	408.26	3	0.00813
:	:	:	:
:	:	:	:
:	:	:	:
public	16.96	245	0.99186
programme	15.40	246	0.99593
representatives	15.32	247	1.00000
$n = 247$			

Table 3: Some examples of transformed values of each term in $CT(powers)$.

In the next section, we describe the setup including the data, the lexicon and the evaluation used in our experiments.

5 Experimental setup

5.1 Data

For comparable text, we derive English and Spanish distinct sentences from the Europarl parallel corpora. We split the corpora into three parts according to year. We used about 500k sentences for each language in the experiments. This approach is further explained in Ismail and Manandhar (2009) and is similar to Koehn and Knight (2001) and Haghighi et al. (2008).

5.2 Pre-processing

For corpus pre-processing, we use sentence boundary detection and tokenization on the raw text before we clean the tags and filter stop words. We sort and rank words in the text according to their frequencies. For each of these words, we compute their context term log-likelihood values.

5.3 Lexicon

In the experiment, a bilingual lexicon is required for evaluation. We extract our evaluation lexicon from the `Word Reference`[2] free online dictionary. This extracted bilingual lexicon has low coverage.

[2]`http://wordreference.com`

5.4 Evaluation

In the experiments, we considered the task of building a bilingual English-Spanish lexicon between the 2000 high frequency source and target words, where we required each individual word to have at least a hundred highly associated context terms that are not part of the initial seed dictionary. Different highly associated W_R terms for a given W_T might derive similar (W_S, W_T) pairs. In this case, we only considered one of the (W_S, W_T) pairs. In future work, we would like to keep these for word sense discrimination purposes. Note that we only considered proposed translation pairs whose similarity values are above a threshold t_2.

We used the F_1 measure to evaluate the proposed lexicon against the evaluation lexicon. If either W_S or W_T in the proposed translation pairs is not in the evaluation lexicon, we considered the translation pairs as unknown, although the proposed translation pairs are correct. $Recall$ is defined as the proportion of the proposed lexicon divided by the size of the lexicon and $precision$ is given by the number of correct translation pairs at a certain recall value.

6 Experiments

In this section, we look into how the in-domain context vectors affect system performance. We also examine the potential of rank-binning similarity measure.

6.1 From standard context vector to in-domain context vector

Most research in bilingual lexicon extraction so far has employed the standard context vector approach. In order to explore the potential of the in-domain context vectors, we compare the systems that use in-domain approach against systems that use the standard approach. We also employ different sets of seed lexicon in each system to be used in the similarity measure:

- Lex_{700}: contains 700 cognate pairs from a few Learning Spanish Cognate websites[3].

- Lex_{100}: contains 100 bilingual entries of the most frequent words in the source corpus that have translation equivalents in the extracted evaluation lexicon. We select the top one hundred words in the source corpus, so that their translation equivalents is within the first 2000 high frequency words in the target corpus.

- Lex_{160}: contains words with similar spelling that occur in both corpora. We used 160 word pairs with an edit distance value less than 2, where each word is longer than 4 characters.

Models using the standard approach are denoted according to the size of the particular lexicon used in their context similarity measure, i.e. *CV-100* for using Lex_{100}, *CV-160* for using Lex_{160} and *CV-700* for using Lex_{700}. We use *IDT* to denote our model. We use lexicon sizes to distinguish the different variants, e.g. *IDT-CV100* for using Lex_{100}, *IDT-CV160* for using Lex_{160} and *IDT-CV700* for using Lex_{700}.

With *CV-700*, the system achieved 52.6 percent of the best F_1 score. Using the same seed dictionary, the best F_1 score has increased about 20 percent points with *IDT-CV700* recorded 73.1 percent. *IDT-CV100* recorded about 15.0 percent higher best F_1 score than *CV-100* with 80.9 and 66.4 percent respectively. Using an automatically derived seed dictionary, *IDT-CV160* yielded 70.0 percent of best F_1 score while *CV-160* achieved 62.4 percent. Results in Table 4 shows various precisions p_x at recall values *x*.

Model	$P_{0.10}$	$P_{0.25}$	$P_{0.33}$	$P_{0.50}$	$BestF_1score$
CV-700	58.3	61.2	64.8	55.2	52.6
CV-100	52.0	53.0	47.2	44.8	66.4
CV-160	68.5	56.8	48.8	48.8	62.4
IDT-CV700	83.3	90.2	82.0	66.7	73.1
IDT-CV100	80.0	75.8	66.7	69.4	80.9
IDT-CV160	90.0	80.6	73.9	69.2	70.0

Table 4: Performance of different models.

[3]such as `http://www.colorincolorado.org` and `http://www.language-learning-advisor.com`

6.2 Similarity measure using rank-binning

We use *RB* to denote our model based on the rank-binning approach. Running *RB* means that no seed dictionary is involved in the similarity measure. We also ran the similarity measure in the *IDT* (*IDT-RB160*) by employing the derived Lex_{160} for the in-domain steps.

We ran several tests using *IDT-RB160* with different numbers of bins. The results are illustrated in Figure 4. The *IDT-RB160* yielded 63.7 percent of best F_1 score with 4 bins. However, the F_1 score starts to drop from 61.1 to 53.0 percent with 6 and 8 bins respectively. With 3 and 2 bins the *IDT-RB160* yielded 63.7 and 62.0 percent of best F_1 score respectively. Using 1 bin is not be possible as all values fall under one bin. Thus, the rank-binning similarity measure for the rest of the experiments where *RB* is mentioned, refers to a 4 bins setting.

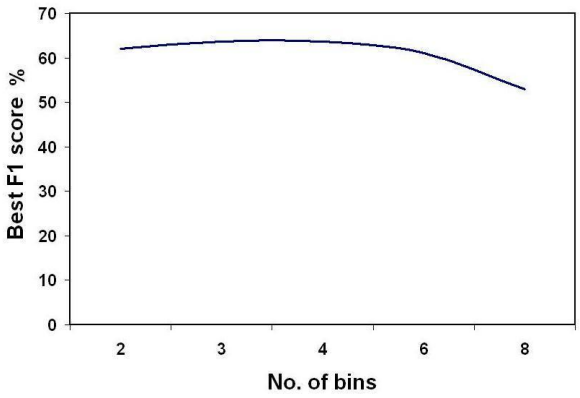


Figure 4: Performance of *IDT-RB160* with different numbers of bins.

While systems using the standard context similarity measure yielded scores higher than 50.0 percent of best F_1, the *RB* achieved only 39.2 percent. However, *RB* does not employ an initial dictionary and does not use orthographic features. As mentioned above, the system scored higher when the similarity measure was used in the *IDT* (i.e. *IDT-RB160*). Note that Lex_{160} is derived automatically so the approach can also be considered as unsupervised. The system performance is slightly lower compared to the conventional *CV-160*. However, *IDT-CV160* outperforms both of the systems (see Figure 5).

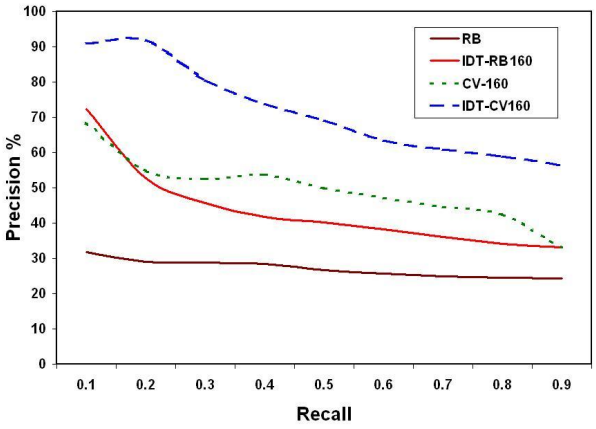


Figure 5: Performance of different unsupervised models.

Overall, systems that exploit in-domain terms yielded higher F_1 scores compared to the conventional context vector approach.

6.3 Comparison with CCA

Previous work in extracting bilingual lexicons from comparable corpora generally employ the conventional context vector approach. Haghighi et al. (2008) focused on applying *canonical correlation analysis* (*CCA*), a dimension reduction technique, to improve the method. They were using smaller comparable corpora, taken from the first 50k sentences of English Europarl and the second 50k of Spanish Europarl, and different initial seed dictionary. Hence, we tested *CCA* in our experimental setup. In *CV-700* setting, using *CCA* yields 57.5 percent of the best F_1 score compared to 73.1 percent of the best F_1 score with *IDT* that we reported in Section 6.2.

7 Discussion

7.1 Potential of in-domain terms

Our experiments clearly demonstrate that the use of in-domain terms achieves higher F_1 scores compared to conventional methods. It also shows that our method improves upon earlier reported dimension reduction methods. From our observation, the number of incorrect translation pairs

were further reduced when the context terms were filtered. Recall that the in-domain terms in the target language were actually the shared context terms of the target word and its highly associated context terms. Nevertheless, this approach actually depends on the initial bilingual lexicon in order to translate those highly associated context terms into the source language. Table 5 shows some examples of most confidence translation pairs proposed by the *IDT-CV100*.

English	Spanish	Sim score	Correct?
principle	principio	0.9999	Yes
government	estado	0.9999	No
government	gobierno	0.9999	Yes
resources	recursos	0.9999	Yes
difficult	dificil	0.9999	Yes
sector	competencia	0.9998	No
sector	sector	0.9998	Yes
programme	programa	0.9998	Yes
programme	comunidad	0.9998	No
agreement	acuerdo	0.9998	Yes

Table 5: Some examples of most confident translation pairs proposed by *IDT-CV100* ranked by similarity scores.

7.2 Seed dictionary variation

The initial seed dictionary plays a major role in extracting bilingual lexicon from comparable corpora. There are a few different ways for us to derive a seed dictionary. Recall that Lex_{700} and Lex_{100}, that are used in the experiments, are derived using different methods. The F_1 scores of the system using Lex_{100} were much higher compared to the system using Lex_{700}. Thus, extending Lex_{100} with additional high frequency words may provide higher accuracy.

One important reason is that all bilingual entries in Lex_{100} occur frequently in the corpora. Although the size of Lex_{700} is larger, it is not surprising that most of the words never occur in the corpora, such as *volleyball* and *romantic*. However, using Lex_{160} is more interesting since it is derived automatically from the corpora, though one should realize that the relationship between the language pair used in the respective monolingual corpora, English and Spanish, may have largely affect the results. Thus, for other systems involving unrelated language pairs, the rank-binning similarity measure might be a good option.

7.3 Word sense discrimination ability

As mentioned in Section 5.4, each source word may have more than one highly associated context term, W_R. Different W_R may suggest different target words for the same source word. For example, given the source word *powers* and the highly associated word *community*, *competencias* is proposed as the best translation equivalent. On the other hand, for same source word *powers*, when the highly associated word is *delegation*, the target word *poderes* is suggested.

8 Conclusion

We have developed a method to improve the F_1 score in extracting bilingual lexicon from comparable corpora by exploiting in-domain terms. This method also performs well without using an initial seed dictionary. More interestingly, our work reveals the potential of building word sense disambiguated lexicons.

References

Aria Haghighi, Percy Liang, Taylor Berg-Kirkpatrick and Dan Klein. 2008. Learning bilingual lexicons from monolingual corpora. In *ACL 2008*, Columbus, Ohio.

Azniah Ismail and Suresh Manandhar. 2009. Utilizing contextually relevant terms in bilingual lexicon extraction In *Workshop on Unsupervised and Minimally Supervised Learning of Lexical Semantics*, Boulder, Colorado.

Mona Diab and Steve Finch. 2000. A statistical word-level translation model for comparable corpora. In *Proceedings of the Conference on Content-based multimedia information access (RIAO).*

Pascale Fung. 1995. Compiling bilingual lexicon entries from a non-parallel English-Chinese corpus. In *Proceedings of the 3rd Annual Workshop on Very Large Corpora*, Boston, Massachusetts, 173-183.

Philipp Koehn and Kevin Knight. 2001. Knowledge sources for word-level translation models. In *Proceedings of the Conference on empirical method in natural language processing (EMNLP).*

Philipp Koehn and Kevin Knight. 2002. Learning a translation lexicon from monolingual corpora. In *Proceedings of the ACL 2002*, Philadelphia, USA, 9-16.

Reinhard Rapp. 1995. Identifying word translations in non-parallel texts. In *Proceedings of the ACL 33*, 320-322.

Reinhard Rapp. 1999. Automatic identification of word translations from unrelated English and German corpora. In *Proceedings of the ACL 37*, 519-526.

Ted Dunning. 1993. Accurate methods for the statistics of surprise and coincidence. *Computational Linguistic*, volume 19(1), 61-74.

Wilson Yiksen Wong. 2009. *Learning lightweight ontologies from text across different domains using the web as background knowledge*. Ph.D. Thesis. University of Western Australia

A framework for representing lexical resources

Fabrice Issac
LDI
Université Paris 13

Abstract

Our goal is to propose a description model for the lexicon. We describe a software framework for representing the lexicon and its variations called Proteus. Various examples show the different possibilities offered by this tool. We conclude with a demonstration of the use of lexical resources in complex, real examples.

1 Introduction

Natural language processing relies as well on methods, algorithms, or formal models as on linguistic resources. Processing textual data involves a classic sequence of steps : it begins with the normalisation of data and ends with their lexical, syntactic and semantic analysis. Lexical resources are the first to be used and must be of excellent quality.

Traditionally, a lexical resource is represented as a list of inflected forms that are projected on a text. However, this type of resource can not take into account linguistic phenomena, as each unit of information is independent. This results in a number of problems regarding the improvement or review of the resource. On the other hand some languages such as Arabic, because of the potential large lexicon, lends itself less easily to this kind of manipulation.

Our goal is to propose a model for the description of the lexicon. After presenting the existing theory and software tools, we introduce a software framework called Proteus, capable of representing the lexicon and its variations. The different possibilities offered by this tool will be illustrated through various examples. We conclude with a demonstration of the use of lexical resources in different languages.

2 Context

Whatever the writing system of a language (logographic, syllabic or alphabetic), it seems that the word is a central concept. Nonetheless, the very definition of a word is subject to variation depending on the language studied. For some Asian languages such as Mandarin or Vietnamese, the notion of word delimiter does not exist ; for others, such as French or English, the space is a good indicator. Likewise, for some languages, prepositions are included in words, while for others they form a separate unit.

Languages can be classified according to their morphological mechanisms and their complexity ; for instance, the morphological systems of French or English are relatively simple compared to that of Arabic or Greek.

There are two main branches of morphology, inflectional or grammatical morphology and lexical morphology. The first one deals with context-related variations, as the rules of agreement in gender and number or the conjugation of verbs. The second one concerns word formation, generally involving the association of a lexeme to prefixes or suffixes.

3 Tagging

Text tagging consists in adding one or more information units to a group of characters : the token. This association is firstly performed in a context-free way, that is to say considering only the token, and secondly by increasing the context size : the tagging process is subsequently repeated in or-

Coling 2008: Poster Volume, pages 490–497,
Beijing, August 2010

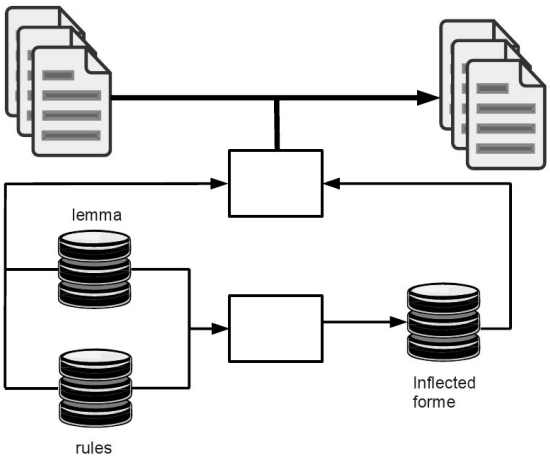


Figure 1: tagging schema

der to merge multiple tokens. Token merging applies to polylexical units, syntactic or para-textual structures.

We distinguish two main types of resources that can be projected on raw texts in order to enrich them. The first of these resources is a set of inflected forms associated with a number of information units (in the example below, the lemma and the morphosyntactic annotation of each form) :

```
abyssal   abyssal  A--ms
abysses   abysse   N-mp
```

The projection of this type of resources in textual corpora is quite simple. After identifying a token, the program only needs to check if the token is included in the resource and add the information units associated with it.

The second type of resources contains a set of rules and a set of canonical forms (usually the lemma but not necessarily). These sets are used jointly, to produce all the inflected forms or to analyse the tokens. Analysis consists in determining, for a given inflected form, which rule was used, on which canonical form, in order to generate it. Then the information to be associated with the inflected form is related to the rule found.

Diagram 1 presents the place of different resources in the tagging process.

4 Tools and resources

Several concepts are related to the use of lexical resources ; here we provide some examples of tools, theoretical as well as computational.

- resources in the form of a frozen list: Morphalou (Romary et al., 2004), Morfetik (Buvet et al., 2007), Lexique3 (New, 2006) ;
- lexical representation formalisms: DATR (Evans and Gazdar, 1996) ;
- inflections' parsers: Flemm (Namer, 2000) ;
- complete software platforms: Nooj (Silberztein, 2005), Unitex (Paumier, 2002) ;
- lexicon acquisition: lefff (Sagot et al., 2006).

4.1 Frozen resources

If this kind of resource is directly used in the tagging process, it raises many maintenance issues. Moreover, in the case of languages with rich morphology, the number of elements becomes too large. These lists are most often the result of inflection engines that use canonical forms and inflection rules to generate the inflected forms.

4.2 Hierarchical lexical representation formalisms

The goal of this type of formalisms is to represent the most general information at the top-level of a hierarchy. There is an inheritance mechanism to transmit, specify, and if necessary, delete information along the structure. It is possible to group under one tag a set of morphological phenomena and their exceptions. Multiple inheritance allows a node to inherit from several different hierarchies.

4.3 Inflections' parsers

They propose a morphological analysis for a given inflected form : they try to apply the derivation rules backwards and test whether the result obtained corresponds to an attested canonical form. The use of canonical forms is optional ; it provides an analysis for lexical neologisms but can cause incorrect results (*Hundred* is not the past participle of *hundre*).

4.4 Software platforms

Unitex / Intex / NooJ are complete environments for performing linguistic analysis on documents. They are able to project dictionaries on texts and to query them. They offer a set of tools for managing dictionaries, both lexical and inflectional. NooJ is the successor of Intex ; among the new features, is the redesign of the architecture of the dictionaries. It proposes handling simple and compound words in a single way. The method is a mix of manipulation of characters and use of morphological grammars.

The inflexion mechanism is based on classic character handling operators as well as on word manipulation operators. Here is the list of some operators:
<B> delete last character
<D> duplicate last character
<L> go left
<R> go right
<N> go to the end of next word form
<P> go to the end of previous word form
<S> delete next character

5 Representation and structuring of inflections: the Proteus model

We introduce a framework capable to represent and structure inflections efficiently, not only in terms of resource creation, but in terms of linguistic consistency too. At the inflection level we propose a simple multilingual mechanism for simple and compound forms. It is used both to generate simple forms and to analyse them. Regarding the lexicon, the model allows for clusters.

We distinguish three levels:

- the inflection level: determine how to produce a derived form from a base form ; the atomic processing unit here is the character (*i.e. local transformation*).
- the conjugation level: determine how to organise family rules effectively in order to avoid redundancy ; the atomic processing unit here is the transformation rule.
- the word level: once the derived form is produced, determine which operation is required to validate the form against non-morphological rules ; the processing unit here is the token (*i.e.* global transformation).

The model was developed to meet the following objectives:

1. A verbatim description of a language does not allow for the analysis of unknown words even if their inflection is regular. We must therefore develop a mechanism that we can use for both analysis and generation. Then we will be able to analyse not only known words but also neologisms.
2. In a lexical database, where, for French, the number of elements reaches one million, the presence of an error is always possible, even inevitable. We must therefore consider an effective maintenance procedure : a dictionary of lemmas linked to a dictionary of inflections and not a read-only resource containing all inflected forms.
3. The concept of word is so complex that we cannot limit a resource to simple words. The model must integrate the management of both simple and compound words. The only limit we set is syntax: the management of idioms, even if it is fundamental, requires the implementation of other tools.
4. The concept of inflection varies depending on the language. We must build a system capable of dealing with all types of affixation (prefixation, suffixation or infixation). The treatment of Arabic is from this point of view a good indicator since it uses all three types of affixation.
5. An inflection rule, applied to a canonical form, is never completely autonomous ; it is part of a group. For instance, we group together all the inflections of a verb type, for all tenses.
6. The transformation is not limited to morphological changes. For instance, phonological phenomena can occur too. More generally there are treatments that cannot be modelled on simple rules.

7. The proposed model is based on a set of simple tools, it is able to easily integrate third-party applications and allows use of dictionaries built in another environment.

5.1 Inflection description

Let f be the inflexion function and f^{-1} its inverse function, then

$$f(canonicalform, code) = inflectedform$$

$$f^{-1}(inflectedform, code) = canonicalform$$

By simplifying the model to the extreme, we use of a rule that generates an inflected form from its lemma. The form is represented as a list of caracters : (i) it shifts characters from the list to a stack or vice versa (ii) and deletes or inserts characters in the list. By default, the operations apply to characters placed at the end of the list of characters or, depending on the operator, at the top of the stack. Most operators allow for the application of the inverse function for the analysis of an inflection. Due to the operator based construction, the function has the following property:

let c_1, c_2 be valid code and x a character string, then

$$f(f(x, c_1), c_2) = f(x, c_1 \bullet c_2)$$

We now present the different operators. They must be sufficiently numerous, to offer the necessary expressive power to represent any kind of inflection, but small enough, not to make the task of creating the rule too difficult.

P (Push) : move a character from the list to the stack

D (Dump) : moves the character of the stack to the list

E (Erase) : deletes a character from the list

/x/ : adds the character x at the end of the list

To simplify code writing, it is possible to indicate the number of repetitions of an operator. Here is an example of code that generates an inflected form from its lemma.

Step	Mot	Pile	Code restant
1	céder		`3PE/è/3D/ais/`
2	cé	der	`E/è/3D/ais/`
3	c	der	`/è/3D/ais/`
4	cè	der	`3D/ais/`
5	cèder		`/ais/`
6	cèderais		

Steps:

1→2 : stack of three characters

2→3 : deleting of a character

3→4 : adding the character *è*

4→5 : dumping three characters from the stack

5→6 : addition of three characters *ais*

This code can inflect french verbs like *céder* (as *révéler*, *espérer*, ...). However, this kind of code does not allow reversing the operation, *i.e.* find the lemma from the inflection : the E (erase) operator, unlike other operators, is not reversible. Therefore we add another operator to erase this function or remove the characters to delete.

`\x\` : erase given character from the list (this rule cannot be applied if the character is not present)

The code of the previous example becomes `3P\é\/è/3D/ais/`.

Since consonant duplication is a common phenomenon, we introduce a specific operator :

C (Clone) : duplicates the last character of the list.

The code `C/ing/` generates the present participle of words such as *run*, *sit* or *stop*

The management of prefixes requires the addition of operators:

`]` (fill stack) : transfers all the characters from the list to the stack

`[` (empty stack) : transfers all the characters from the stack to the list

Operator] can prepare an addition at the beginning of a word since all characters are put in the stack. We are now able to describe the inflexion of the form: *move* → *unmovable*. The transformation "remove the character 'e' at the end of a word, add 'un' at the beginning and 'able' end of a word" is coded `\e\]/un/[/able/`. The same code can analyse verb constructions that end with the character *e*.

Processing compound words requires the addition, or rather the transformation, of an operator. The difficulty here is to distinguish the different components of an expression with respect to one or more separators (traditionally in French space, hyphen or apostrophe).

P|x| (Push): moves the character of the list to the stack to meet the character x

Changing the stacking operator allows us to access directly an element of an expression or a compound word. Please note that access to different elements of an expression is achieved by stacking and unstacking successively. The code `2P|-|/s/[` allows to form the plural of expressions such as *brother-in-law* : only the third word from the end is pluralized (*brothers-in-law*). To preserve the analysis function of the model, it would be necessary to add, symmetrically, a conditional popping operator (*e.g.* `D|x|`). However, compound words analysis is far more complex, and such an operator could not bring the solution.

5.2 Management of inflexion

We have defined an XML DTD to manage the inflections expressed in code.

```
<flex id="n-y-p" type="final">
    <name>Np</name>
    <info>Noun plural with
          a terminal y</info>
    <code>\y\/ies/</code>
<\flex>
```

The above definition associates n-y-p identifier with the code `\y\/ies/`. A typical inflexion is characterised by:

- an identifier (attribute `id`) is used by the description language ;
- a status (optional attribute `type`) ;
- a name (optional element `<name>`) which corresponds to the tag associated to the inflected form ;
- information (optional element `<info>`) about the inflection ;
- a Proteus inflection code (element `<code>`).

However it is often necessary to combine several transformations : masculine/feminine and singular/plural for nouns and adjectives, persons and tenses for verbs. Take for example the conjugation of a French verb in the first group in the present tense. The prototypical inflection may be given as follows:

```
<flex id="v1ip" type="term">
    <name>Vp</name>
    <info>verbes
        indicatif présent</info>
    <flex id="p1ns">
        <name>1s</name>
        <code>/e/</code>
    </flex>
    <flex id="p2ns">
        <name>2s</name>
        <code>/es/</code>
    </flex>
    <flex id="p3ns">
        <name>3s</name>
        <code>/e/</code>
    </flex>
    <flex id="p1np">
        <name>1p</name>
        <code>/ons/</code>
    </flex>
    <flex id="p2np">
        <name>2p</name>
        <code>/ez/</code>
    </flex>
    <flex id="p3np">
        <name>3p</name>
        <code>/ent/</code>
    </flex>
</flex>
```

In this structure we regroup all the inflections of a given tense. Each inflection is : associated to its own identifier, prefixed with the main identifier, separated with a point, and associated to a name which is also a concatenation. Note that it is the identifier that must be unique and not the name. This mechanism allows for the expression of variants in a paradigm (see below). The previous definition states that, for the first group of the present tense, French verbs require suffixes at the end of the canonical form. Note that this is

a generic definition that can take into account exceptions, and can be apllied to any tense or mood.

identifier	name	code
v1ip.p1ns	Vip1s	/e/
v1ip.p2ns	Vip2s	/es/
v1ip.p3ns	Vip3s	/e/
v1ip.p1np	Vip1p	/ons/
v1ip.p2np	Vip2p	/ez/
v1ip.p3np	Vip3p	/ent/

It is also possible to group inflections with a new element (<op> with the attribute type).

```
<flex id="vig1-1" type="nonterm">
    <name></name>
    <info>first group
      indicative</info>
    <op type="add">
        <item value="v1ip"/>
        <item value="v1ii"/>
        <item value="v1ips"/>
        <item value="v1ifs"/>
    </op>
</flex>
```

To the previous definitions we need to modify the code in order to add a prefix operation: *remove the 'er' at the end of the lemma*. So we added the possibility of code concatenation to a previously defined group. In the example bellow the pos attribute determines if the code to be added is a prefix (p) or a suffix (s). The value attribute indicates the identifier of the structure upon which the operation is applied.

```
<flex id="v1" type="final">
    <name></name>
    <info>"er" verb</info>
    <op type="conc" value="vig1-1">
        <item pos="p">\re\</item>
    </op>
</flex>
```

In some cases, modification has to be performed on a particular inflection. This is done via the application of a *mask* which operates on a group of inflections and changes, possibly selectively, codes of inflexion. A mask is a set of rules applied on code. A regular expression on the identifier (ervalue attribute) performs the selection. We use Proteus code to modify Proteus code. This *mise en abyme* seems inconsistent, since Proteus has been designed to apply on a language element. But it seemed inappropriate to introduce a new syntax.

The definition below allows to add the letter *e* to a form in order to maintain its pronunciation [ʒ].

```
<mask id="m-ge">
 <info>add e after a g</info>
  <item ervalue="v1ip\.p1np">
        ]5D/e/[</item>
  <item ervalue="v1ii\.p([123]ns|3np)">
        ]5D/e/[</item>
  <item ervalue="v1if\.p([12]n[ps]|3np)">
        ]5D/e/[</item>
</mask>
```

The previous definition transforms code as \er\/ons/ in \er\/eons/, \re\/ais/ in \re\/eais/, ... The mask is used in combination with the attribute mask in a inflection definition, as in the conc attribute.

```
  <flex id="v1" type="final">
      <name></name>
      <info>verbes en "er"</info>
      <op type="mask" value="vig1-1">
          <item value="m-ge"/>
      </op>
  </flex>
```

You can build a complex inflection by using a base and applying masks successively. The inflection of the French verb *neiger* (to snow) can be expressed using two masks. First a mask to take into account the pronunciation of the [ʒ] and a second one, the weather verb mask, which is only used in the third singular person.

5.3 Applications

The examples bellow show the different capabilities of the model.

5.3.1 Neologisms in French

The formation of French inflections should not create significant problems. For the most common languages, simple forms are less than 1 million. Therefore, most systems use this set of inflected forms and supply modules to guess unknown words, only when they arise. This experiment is used to validate the model and to analyse unknown forms.

The example below shows how we analyse an unknown form.

```
anticonservationnistes =(/s/)=>
anticonservationniste =(]/anti/[)=>
conservationniste =(/niste/)=>
conservation =(\e\/ation/)=>
conserve
```

The algorithm tries to apply a code and reiterates the process on the result until we obtain an

attested form. The set of rules provides a potential analysis of the unknown word. Note that the rules used allow to determine the part of speech. In the example, the analysed word can be a plural noun or an adjective.

5.3.2 Arabic verbs

Arabic is a Semitic language ; it uses a semitic root to derive all the words used. For example from the root كتب (which refers to the writing) it is possible to produce verb (write), noun (desk) or adjective (written).

With these lemmas it is possible to agglutinate prefixes and suffixes. The rules are very regular in morphology but also very productive. We build all inflexion from the semitic root. So we have the schema: root → radical → inflected form. We then define inflexion for prefix/suffix (identifier `pass1term`), a mask for the radical (identifier `pass1radical`) and a definition which combine both.

```
<mask id="pass1radical">
 <info>add radical past</info>
 <item ervalue=".">
   ]/2P\/´\/D\/´\/D/[+
 </item>
</mask>
```

```
<flex id="pass1" type="nonterm">
  <name>Vis</name>
  <info>passe</info>
  <op type="mask" value="pass1term">
    <item value="pass1radical"/>
  </op>
</flex>
```

The problem encounter with arab text is the possible non use of all vowels. In fact they are rarely used, generally in pedagogical or religious text. This mean that the context is fundamental to interpret a text, a vowel is added only to remove an ambiguity. However we decide to describe language fully vowelled and to manage this specificity in an earlier stage.

The objective is to provide a resource used during an lexical analysis. This can be done in two ways[1]:

[1]We used here a transliteration version of arab writing to be more clear.

5.3.3 Old French

We are developping (author reference) an Old French resource, as exhaustive as possible. One difficulty is to consider the various alternatives, dialectal or chronological. This *proto-morphological* problem complicates the development of the dictionary nomenclature. We solved this problem by introducing an arbitrary "language Phantom" and by adding one level to the composition of the nomenclature, in the form of a label named hyperlemma. All derivations are from this entity using Proteus rules. All variants are generated from this item by application of successive masks.

The example below shows the successive masks applied on the inflection rules to account for the variations of the imperfect tense. Each mask is named *modifxxx* and corresponds to the modification of the Proteus rule for each century *xxx*.

```
<flex id="vg1i-5" type="final">
      <name>Vii</name>
      <info>first group
            imparfait</info>
    <op type="mask" value="v1ii">
        <item value="modifXI"/>
        <item value="modifXII"/>
        <item value="modifXIII"/>
        <item value="modifXIIIa"/>
        <item value="vrber"/>
    </op>
</flex>
```

6 Implementation

This framework is not only a theoretical tool ; it is designed to be implemented in a tagging software as an autonomous module. Based on abstract descriptions (Proteus code and XML language), it allows the resource creator to focus on linguistic aspects. It is simple enough to be easily expressed in any computational language.

The platform described here is developed in Python, which allows a very compact coding and can be used for both generation and analysis.

7 Conclusion

Our work is part of a set of tools and resources dedicated to the analysis of natural language. We have presented a model for the representation of inflections coupled with a language to structure the transformation rules. Compound words are

handled in the same way as simple words. The proposed model also allows simple word identification in both analysis and resource generation functionality. We have presented three examples of the use of the model, each introducing a specificity: French, Old French and Arabic. In the near future we expect to begin work on the Korean, Polish and Greek.

The best way to improve the framework is to create real, *i.e.* exhaustive linguistic resources. The development of the framework can be considered from several ways.

The Proteus code and the XML language description need stability. In our opinion, addition of operations to take into account some language specificities would complicate the model without adding any significant improvement. These modifications will take place during the third stage, the word level, where post-treatments are applied. For instance, the tonic accent in Greek can move along the last three syllables of a word and affects the use of the diaeresis mark in diphthongs.

As far as the analysis functionnality is concerned, we are considering to develop specific heuristics for each language in order to guide the choice of rules.

References

Buvet, Pierre-André, Emmanuel Cartier, Fabrice Issac, and Salah Mejri. 2007. Dictionnaires électroniques et étiquetage syntactico-sémantique. In Hathout, Nabil and Philippe Muller, editors, *Actes des 14e journées sur le Traitement Automatique des Langues Naturelles*, pages 239–248, Toulouse. IRIT Press.

Evans, Roger and Gerald Gazdar. 1996. Datr: A language for lexical knowledge representation. *Computational Linguistics*, 22(2):167–216.

Namer, F. 2000. Flemm : Un analyseur flexionnel du français à base de règles. *Revue Traitement Automatique des Langues*, 41(2).

New, Boris. 2006. Lexique 3 : Une nouvelle base de données lexicales. In Mertens, P., C. Fairon, A. Dister, and P. Watrin, editors, *Verbum ex machina. Actes de la 13e conférence sur le Traitement automatique des langues naturelles*, Cahiers du Cental 2,2, Louvain-la-Neuve. Presses universitaires de Louvain.

Paumier, Sébastien, 2002. *Manuel d'utilisation du logiciel Unitex*. Université de Marne-la-Vallée.

Romary, Laurent, Susanne Salmon-Alt, and Gil Francopoulo. 2004. Standards going concrete : from lmf to morphalou. In Zock, Michael, editor, *COLING 2004 Enhancing and using electronic dictionaries*, pages 22–28, Geneva, Switzerland, August 29th. COLING.

Sagot, Benoît, Lionel Clément, Éric Villemonte de la Clergerie, and Pierre Boullier. 2006. The lefff 2 syntactic lexicon for french: architecture, acquisition, use. In *LREC'06*, Gênes.

Silberztein, Max. 2005. NooJ's dictionaries. In Vetulani, Zygmunt, editor, *LTC'05*, pages 291–295, Poznań, Poland, April.

Language-Specific Sentiment Analysis in Morphologically Rich Languages

Hayeon Jang
Dept. of Linguistics
Seoul National University
hyan05@snu.ac.kr

Hyopil Shin
Dept. of Linguistics
Seoul National University
hpshin@snu.ac.kr

Abstract

In this paper, we propose language-specific methods of sentiment analysis in morphologically rich languages. In contrast of previous works confined to statistical methods, we make use of various linguistic features effectively. In particular, we make chunk structures by using the dependence relations of morpheme sequences to restrain semantic scope of influence of opinionated terms. In conclusion, our linguistic structural methods using chunking improve the results of sentiment analysis in Korean news corpus. This approach will aid sentiment analysis of other morphologically rich languages like Japanese and Turkish.

1 Introduction

The Internet is a global forum where citizens of the world gather to express their opinions. Online services exist for users to share their personal thoughts while the use of blogs and Twitter substitutes for private diaries. For this reason, sentiment analysis which automatically extracts and analyzes the subjectivities and sentiments (or polarities) in written texts has recently been receiving attention in the field of NLP.

Sentiment analysis of English employs various statistical and linguistic methods referencing such linguistic resources as The Berkeley Parser and SentiWordNet. In the case of Korean, however, most previous works have been confined to statistical methods which focus either on the frequency of words or relevance of co-occurring words only. This is because it is hard to find proper resources due to the nature of Korean, exhibiting such features as rich functional morphemes, a relatively free word-order and frequent deletion of primary elements of sentences like the subject and object. The major drawbacks of statistical-based approaches are the facts that the 'real' meaning of the expressions which we feel when we read them cannot be reflected in the analysis, and that complex statistical measuring methods are computationally taxing.

In this paper, in order to overcome previous shortcomings, while making use of Korean case studies we propose a new approach for morphologically rich languages that makes effective use of linguistic information such as the semantic classes of words, semantic scope of negation terms like *not, no,* and the functional meaning of modal affixes. Especially, this approach makes chunk structures by using dependency relation of morpheme sequences to limit the semantic scope of influence of opinionated terms. This chunking method is simpler and more efficient than total syntactic parsing. In addition, we utilize subjectivity clues and contextual shifters whose effectiveness is established in previous references.

The contents of this paper are as follows: firstly, we review previous works related to our approaches. We follow up by introducing the framework and main processes of our approach are introduced. Finally, we describe our experiments and show how a linguistic approach is feasible in sentiment analysis of Korean as a morphologically rich language.

2 Related Work

Sentiment analysis research has been performed to distinguish the authors' polarity (sentiment orientation) on certain topics from document-level (Turney, 2002: Pang et al., 2002; Dave et al., 2003) to sentence-level (Hu and Liu, 2004;

Coling 2008: Poster Volume, pages 498–506,
Beijing, August 2010

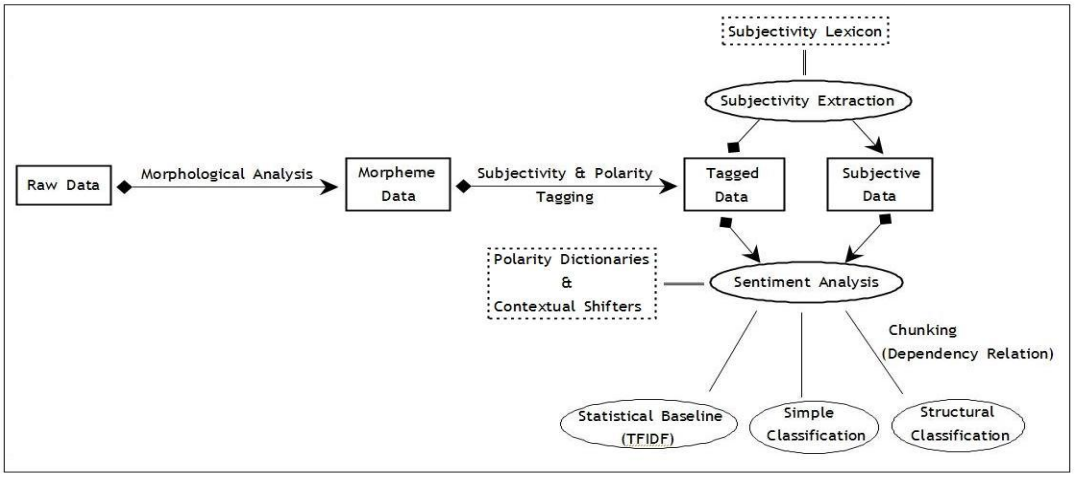


Figure 1. Sentiment Analysis Framework

Kim and Hovy, 2004). We will focus on sentence-level sentiment classification with our presupposition that the polarity of sentences in a single document can be diversified due to the inclusion of various subtopics.

Recently, much research has focused on subjectivity[1] extraction that divides objective facts from subjective opinions in data. Pang and Riloff (2005) and Yu and Hatzivassiloglou (2003) trained sentence-level subjectivity classifiers and proved that performing sentiment analysis targeting selected subjective sentences only gets higher results. We adopt a method of Wiebe and Riloff (2005)'s methods which classifies sentences containing more than two lexical items associated with subjectivity and compare the result of the experiments on full and extracted subjective corpora.

The core of the proposed new approach is the use of structural information in morphologically rich languages in the process of sentiment analysis. Choi et al. (2005) and Mao and Lebanon(2006) are representative of the structured sentiment analysis approach which takes advantage of Conditional Random Fields (CRF) to determine sentiment flow. McDonald et al. (2007) also dealt with sentiment analysis, via the global joint-structural approach. Furthermore, since there are a lot of good parsers for English data, Meena and Prabhakr (2007) and Liu and Seneff (2009) utilized sentiment structure information by such parsers such as Berkeley Parser.

In the case of Korean, much research applies dependency grammars for reducing the complexity of sentences to match the characteristics of Korean (Kim and Lee, 2005; Nam et al., 2008) but this still causes problems which prohibit wide use. Therefore we suggest a new morphological chunking method that binds semantically related concatenations of morphemes. This helps to define boundaries of semantic scopes of opinionated terms and is faster, simpler and more efficient on sentiment analysis than a general full parser.

Our approach focuses on the role of contextual shifters as well. In this paper, the term 'contextual shifter' covers both negation shifters and flow shifters: the former refers to the terms which can change semantic orientation of other terms from positive to negative and vise versa, the latter the terms which can control sentiment flow in sentences, for example, in English *not, nobody* (negation shifters), *however, but* (flow shifters). Kennedy and Inkpen (2006) did sentiment analysis of movie and product reviews by utilizing the contextual shifter information. Miyoshi and Nakagami (2007) also used this method to see the advancement of the result on sentimental analysis of electric product reviews in Japanese. In this work, we make use of the functions of each shifter to properly modify the value of the terms in the sentences and limit the number of the features which have to be observed in the analysis process to increase efficiency.

[1] The term 'subjectivity' is equivalent to Quick et al. (1985)'s private state which was defined as the words and phrases expressing individual mental and emotional states.

3 Sentiment Analysis Framework

The process of sentiment analysis in this paper is described in Figure 1. In this section, we explain each step of the process in detail.

3.1 Morphological Analysis

Korean is an agglutinative language where roots and affixes which have their own functional meaning combine to form complete words. Consequently, sufficient morphological analysis is very important to catch the precise and deep meaning of such expressions. If a certain sentence is misunderstood by wrong morphological analysis, there will be a strong possibility that opinionated terms in the sentence cannot be correctly analyzed.

We used the KTS [2] which is open-source probability based Korean morphological analyzer. Although the probabilistic rules established in KTS are elaborate, the main source of inaccuracy is rooted in the inadequacy of the lexicon. After categorizing all listed words in the sentence, the remaining words are mostly classified as general nouns. In this case, the terms which should play a role as important features in the process of sentiment analysis will be probably misunderstood.

(1) 너무 진부한 내용
nemu cinpuha-n nayyong
too stale-AD[3] content
'too stale contents'

(2) 너무/a 진부/ncs 하/xpa
nemu/a[4] cinpu/ncs ha/xpa
ㄴ/exm 내용/nc
n/exm nayyong/nc

(3) 너/npp 무진/nc 부/nc
ne/npp mucin/nc pu/nc
한/nc 내용/nc
han/nc nayyong/nc
'you Mujin(place name) wealth resentment contents'

For example, if sentence (1) which has to be analyzed as in (2) is incorrectly analyzed as in (3). This fault result ignores original spacing and randomly conjoins syllables in order to find the lexical items included in the dictionary because of the lack of lexicon. As the result, we cannot grasp the intended sentiment cinbu 'stale' in respect to the object nayyong 'contents' in the sentence. In order to solve such problems, we expanded the lexicon of KTS by adding 53,800 lexical items which are included in the Sejong[5] dictionary.

3.2 Subjectivity and Polarity Tagging

News corpora have no marks representing polarity of sentences as exist in the grading systems found in movie review corpora. In addition news data contain relatively more objective sentences which corpora tend to refer to as facts, as compared with reviews. Therefore in the case of news corpora there is a need to process the annotation of subjectivity and polarity tags for each sentence manually.

In our work, two native Korean annotators manually attached polarity labels to each sentence. Sentences are classified as subjective when they contain opinions pertaining to a certain object. Even if the opinion is not expressed on the surface using direct sentiment terms, the sentences are classified as subjective when the annotator can feel the subjectivity through the tone of voice. In the case of sentences containing common sense polarity value words such as donation, murder, etc, terms do not work as the judgment criterion, rather the annotator's judgment about the main theme of the sentence is applied. Only when the sentences are classified as subjective, the polarity tags are attached. The agreement rate of the two annotators in the manual annotation of polarity is 71%.

[2] http://kldp.net/projects/kts/

[3] Abbrebiates: AD(adnominal suffix), NM(nominative particle), IN(instrumental particle), SC(subordinative conjuctive suffix), CP(conjunctive particle), PST(past tense suffix), DC(declarative final suffix), RE(retrospective suffix), CN(conjectural suffix), PR(pronoun), PP(propositive suffix), AC(auxiliary conjunctive suffix), GE (genitive particle)

[4] POS tags of KTS: a(adverb), ncs(stative common noun), xpa(adjective-derived suffix), exm(adnominal suffix),nc (common noun), npp(personal pronoun)

[5] The 21st century Sejong Project is one of the Korean information policies run by the Ministry of Culture and Tourism of Korea. The project was named after King Sejong the Great who invented Hangeul. (http://www.sejong.or.kr/)

Label	Number of items	Lexical items
Positive	2,285 (1838 nouns, 133 verbs , 314 adjectives)	*Coh/pa* 'good', k*elcak/nc* 'masterpiece', *chincel/ncs* 'kind'
Negative	2,964 (2300 nouns, 359 verbs , 305 adjective)	*Nappu/pa* 'bad', *ssuleki/nc* 'trash', *koylophi/pv* 'harass'
Cynical	21 (adverbs)	*celday/a* 'Never', *kyeu/a* 'barely'
Intensifier	91 (80 adverbs, 10 nouns, 1 interjections)	*acu/a* 'very', *hancung/a* 'more', *tanyeonkho/a* 'decisively'
Conjectural	19 (13 final suffixes, 4 pre-final suffixes, 2 adnominal suffixes)	*keyss/efp* CN, *lthenteyo/ef* CN, *l/exm* CN
Obligative	6 (4 final suffixes, 2 auxiliary conjunctive suffixes)	*eya/ecx* 'must', *eyacyo/ef* PP
Quotative	5 (final suffixes)	*ntanunkun/ef* DC, *tayyo/ef* DC

Table 1. Polarity Dictionary

3.3 Subjectivity Extraction

The subjective lexicon used in subjectivity extraction contains 2,469 lexical items which includes 1,851 nouns, 201 verbs, 247 adjectives, 124 adverbs, 44 suffixes, and 2 conjunctive particles. The lemmas of Sejong dictionary are classified by a total of 581 semantic classes. Among them are 23 subjectivity-related semantic classes which include Abusive Language, External Mental State, Internal Mental State etc. Firstly, we have registered those lexical items –nouns, adjectives, verbs- under subjectivity-related semantic classes. Since they will be compared with morphologically analyzed data before subjectivity classification, all items were registered as tagged forms. Nouns took the biggest portion in the lexicon through this process, since adjectives and verbs which consist respectively of stative nouns (ncs) and active nouns (nca) plus derived suffixes (xpa, xpv) were all registered as nouns.

In Korean, sentiment can also be judged from particles and affixes having modal meaning.

(4) 정부가 무응답으로 대응한지 3 일이나 지났다.
*jengpwu-ka mwuungtap-ulo tayungha-nci 3il-**ina** cina-ss-ta.*
Government-NM no response-IN action-SC 3days-CP pass-PST-DC
'It already passed 3 days after government did not response'

(5) 그 배우가 안 나왔더라면 좋았을텐데.
*ku paywu-ka an-nao-ass-te-lamyen coh-ass-**ltheyntey***
the actor-NM not-star-PST-RE-if nice-PST-CN
'It were nice, if the actor would not have starred the main character'

(6) 그거 정말 맛있었겠다.
*ku-ke cengmal masiss-ess-**keyss**-ta*
that-PR really delicious-PST-CN-DC
'That must have been really delicious'

For example, conjunctive particle *-(i)na* in the sentence (4), final suffix *-ltheyntey* in (5), and pre-final suffix *-keyss* in (6) are very influential in judging the subjectivity of sentences. Therefore, we added those functional terms in the subjective lexicon.

We classified the sentences which contains more than two subjective items as subjective. When the sentence contained less than five morphemes, however, we manage to judge the sentence as subjective even when only one subjective item shows. The result of subjectivity extraction is confirmed by the widely used statistical method, TFIDF, in the following section.

3.4 Term Weighting

In our process of sentiment analysis, every term gets its own values by using polarity dictionaries and contextual shifters. In this section we introduce our polarity dictionary and contextual shif-

ters, and their lexical items. Also, the term-weighting methods of our approach is described.

Polarity dictionary: Table 1 shows our polarity dictionary used in sentiment classification. In the same way as a subjective lexicon, all lexical items are registered in the shape of a tagged morpheme. In addition, every item has labels with its own functional categories.

First, Positive and Negative refer to the basic polarity value of individual terms of sentences. The terms that are neither positive nor negative are classified as neutral. We registered nouns, adjectives and verbs included in Sejong dictionary's semantic class related with emotion or evaluation such as Positive Property Human, Negative Property Human, etc. After that, we selected the terms that are generally used to express polarity from other review corpora and added them to the dictionary. Since we deal with on-line texts, we also added acronyms, neologisms and new words which are frequently used to express opinion online.

Next we add various functional lexical items that are from other parts of speech to the polarity dictionary. Cynical items play a role of adding negative nuance to sentences. Intensifiers emphasize the meaning of following expressions. Conjectural, Obligative and Quotative items refer to something other than the author's opinion. Conjectural and Obligative means that the opinion included in the expressions is not actual but hypothetical. Quotative means that opinionated terms which are in same phrase express another person's opinions.

To determine the value of the terms, our approach uses a very simple measuring method. Every term initially gets +1 if Positive, -1 if Negative. All other words receive a value of 0. In the next step, the contexts of the sentences are examined and the values are modified. In the case of simple classification which does not go through the chunking process, we consider the distance of content words in Korean sentences which have various auxiliaries and affixes, and set a [-2, +2] window. In the case of structural classification, we take advantage of structures made by chunking. If Positives and Negatives are neighboring, we modify the values of the terms to reflect the fact that they influence each other. When Cynical items appear with Positives, we multiply by -1 to the value of Positives. When Cynicals appear with Negative items, we intensify the value of Negative by multiplying by 2. If Cynicals appear with neutral terms, we change the value of neutral terms to -1. The value of the terms which are affected by the Intensifier doubles, whereas the values of the terms which are in the scope of Conjectural, Obligative and Quotative items are reduced to half. In this way we control the importance of the terms in the sentence.

Contextual Shifters: contextual shifters in Korean consist of 13 negation shifters (adverbs such as *an/a* 'not'*, mos/a* 'cannot' and auxiliary verbs such as *anh/px* 'not'*, mal/px* 'stop') and 23 flow shifters (sentence-conjunctive adverbs such as *kulena/ajs, haciman/ajs* 'but, though', subordinative conjuctive suffixes *pnitaman/ecs, ntey/ecs* CN and conjunctive suffixes such as *eto/ecx* AC).

Since negation shifters play the role of shifting the polarity of the sentiment terms in our approach, we multiply them by -1. In the case of flow shifters, we limit the number of features to the terms after the shifter appears. We deemed it more important to understand an author's empathetic point, rather than to catch full sentiment flow in the sentences. Also such emphasized contents mostly exist after the flow shifters. Therefore we utilize this characteristic to reduce the work load and to prevent confusions which are caused by other minor sentiment terms.

(7) 음악도 좋고 영상도 좋았는데 스토리가 별로였다.
*umak-to coh-ko yengsang-to coh-ass-**nuntey** sutholi-ka pyello-yess-ta*
music-also good-CN image-also good-CN story-NM not so good-PST-DC
'music was good and image also good though, story is not so good,'

For example, in the sentence (7) *-nuntey* functions as a flow shifter. Dealing with the words after *–nuntey*, we can limit the object morphemes to 5 out of 14. Therefore, measuring load is significantly reduced, and furthermore, we can prevent the confusion from two positive terms *coh* 'good' before the flow shifter.

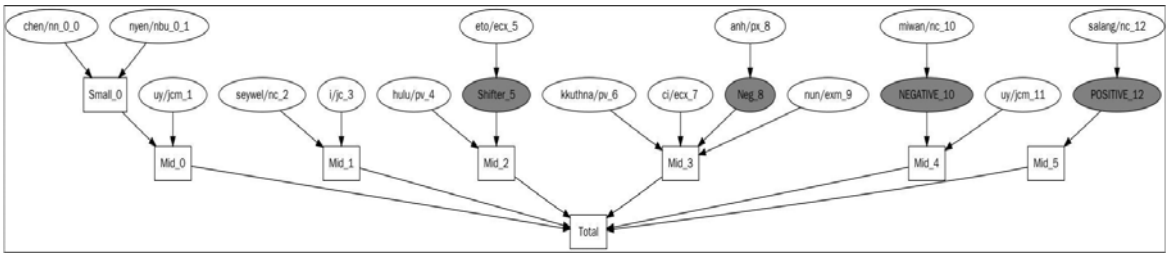


Figure 2. Chunking structure of the below sentence. (A short movie reviews)
천년의 세월이 흘러도 끝나지 않는 미완의 사랑
chen-nyen-uy seywel-i hulu-eto kkuthna-ci anh-nun miwan-uy salang
1000-year-GE time-NM flow-CN finish-CN not-AD incomplete-GE love
'an incomplete love that has not finished even after 1000 years'

3.5 Chunking using morphological dependency relation

In our approach, instead of complete syntactic parsing we use a chunking method based on the dependency relation of morpheme sequences in terms of the provision that it is important to limit the semantic influential scopes of main opinionated expressions.

Korean is a head-final language: in terms of dependency grammar, governors are always located after their dependents. We reflect upon this characteristic to form a relation if a certain morpheme acts as the governor of a previous morpheme. Chunks (small and mid nodes shown in figure 2.) are formed until an unrelated morpheme appears. The terms in a single chunk exert great semantic influence to control the value of each other. After determining the values of every morpheme in each chunk, this process is replicated at a higher level and finally the ultimate values of every term in the sentence are determined.

For example in Figure 2, the structure *[[chen+nyen]+uy] [seywel+i] [hulu+eto] [kkuthna+ci+anh+nun] [miwan+uy] [sarang]* is the result of the chunking process of the sentence *chen-nyen-uy seywel-i hulu-eto kkuthna-ci anh-nun miwan-uy salang* 1000-year-GE time-NM flow-CN finish-CN not-AD incomplete-GE love 'an incomplete love that has not finished even after 1000 years'. If we focus on the terms after the flow shifter *-eto*, the negation shifter *anh* 'not' in the first phrase only influences the verb *kkuthna-* 'finish' in the same chunk. This limitation of semantic scope of the negation shifter eliminates the possibility that it excessively modifies the values of other unrelated elements. Since the simple classification has a [-2, +2] window, *miwan* 'incomplete' is also influenced by *-anh*. Then the value of *miwan* becomes +1 which is classified as a positive term, and the whole expression *miwan-uy salang* 'an incomplete love' is misclassified as positive.

4 Experiment

4.1 Corpora

Since movie review data is commonly used for sentiment analysis, we primarily collected movie reviews. Following the comments of many previous works that it is hard to separate the sentences which mention the plot of movies from opinion sentences, especially short movie reviews which containing 1~2 sentences deliberately selected. The reason is that short reviews having limited space probably include opinions only. Movie review data of less than 20 characters was crawled from a representative movie site in Korea, Cine21[6]. It contains 185,405 reviews ranging from December 31, 2003 to December 28, 2009 (total 19.5MB).

Next, we collected 79,390 news articles from January 1, 2009 to April 7, 2010 (total 146.6MB) from the web site of the daily newspaper, The Hankyoreh[7]. The news data includes both objective and subjective sentences, and is categorized into 3 groups by the following characteristics: 71,612 general news articles, 3,743 opinionated news articles having subjective subtopics such as 'Yuna Kim, terrorism, etc.' and 3,432 editorial articles including columns and contributions. After randomly extracting 100 articles from each data group a Korean annotator attached subjectivity and polarity labels to each

[6] http://www.cine21.com/
[7] http://www.hani.co.kr/

Method	total		subjective	
	Accuracy (%)	F-measure[8] (%)	Accuracy	F-measure
TFIDF	87.67	93.431	90.02	94.748
NO chunking NO shifter	**87.676**	**93.432**	**90.034**	**94.757**
NO chunking YES shifters	**87.674**	**93.433**	90.018	94.745
YES chunking NO shifter	83.212	90.835	87.29	93.214
YES chunking YES shifters	83.212	90.835	87.29	93.214

Table 2. Sentiment analysis of short movie review corpora

Method	Data		Accuracy (%)	F-measure (%)
TFIDF	News articles	Total	63.032	28.532
		Subjective	82.00	89.919
	Subtopic News articles	Total	61.95	32.287
		Subjective	73.332	84.44
	Editorial articles	Total	57.53	73.04
		Subjective	87.23	93.18

Table 3. Subjectivity extraction of news corpora

sentence. The collection of sample sentences consists of 1,225 general news sentences, 1,185 subtopic news sentences and 2,592 sentences of editorial articles.

4.2 Experiment 1: Short Movie Reviews

Table 2 shows the result of a 5-fold cross variation experiment on the sentiment analysis of short movie review data using SVMlight. The numbers in bold face are the values being larger than the baseline, the results using TFIDF. A subjectivity extraction experiment was not carried out because of the presumption that all movie reviews used in this work are subjective. (There were a few reviews containing quotes from the movies or meaningless words only. Such cases, however, were ignored.) In the case of movie review data, selected subjective data is regarded as having stronger subjectivity.

When subjective data is compared with total data by the same experimental methods, there are consistent improvements in sentiment analysis for the subjective data. It is no surprise that the sentences that contain a more intense level of subjectivity can be easily classified as correct polarity.

In addition, contrary to our expectations, the application of the simple classification method (NO chunking) gets the higher results in comparison with the structural classification method (YES chunking) regardless of the use of contextual shifters. This phenomenon can be analyzed based on the limited length of reviews and the characteristics of online data. First, most sentences have a simple structure like the sequence of nouns or noun phrases due to restricted writing space. For this reason, the effect of chunking and contextual shifters on sentiment classification is insignificant. Second, the data includes various terms only seen on the Internet, vulgarisms and ungrammatical words. Furthermore, there are the problems of word spacing and spelling. Because of these drawbacks of online data, morphological analysis errors frequently occurred. The errors are further propagated to structures as a result of chunking. For this reason, when the chunking method is used, contextual shifters are ineffective at all as shown the results using the chunking method in Table 1.

4.3 Experiment 2: News articles

Subjectivity Extraction: The results of a 5-fold cross variation experiment of subjectivity extraction using SVMlight are described in Table 3. In this experiment, we use the commonly used statistical method TFIDF to compare total data with subjective data in the three groups in the subjectivity classification task. In conclusion, the chosen subjective data of all groups get higher results. Especially in the cases of news articles and subtopic news articles which are less subjective than editorial articles, F-measure value is greatly increased.

[8] F-measure = 2*precision*recall/(precision+recall)

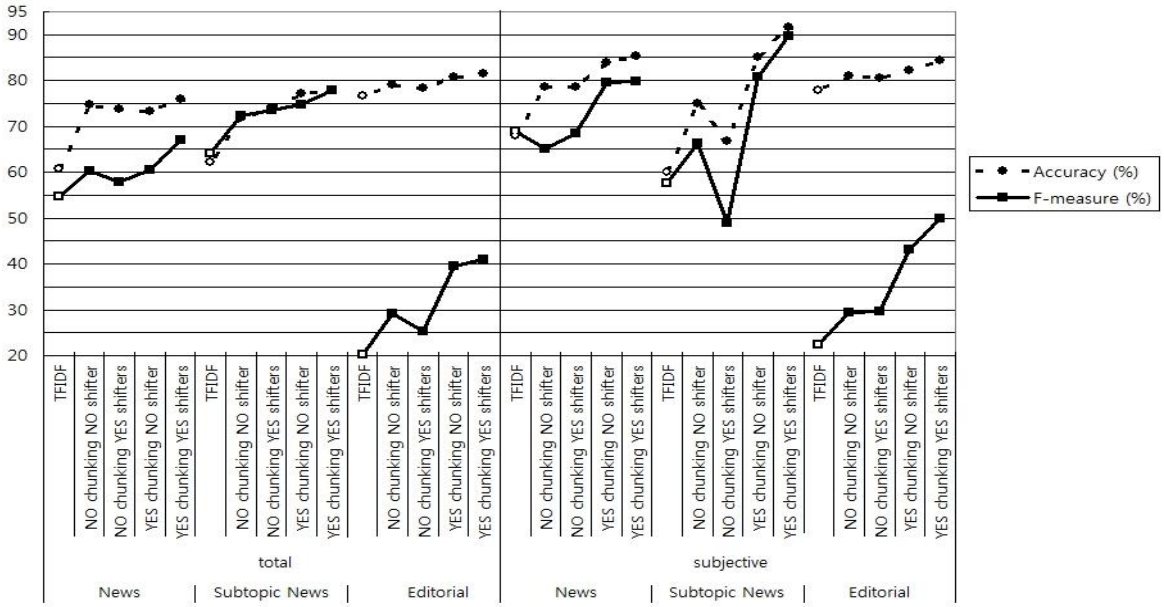


Figure 3. Sentiment analysis of news corpora

Sentiment Analysis: The results of sentiment analysis on the three groups of news data are summarized in Figure 3. The white points in Figure 3 are the values being larger than the baseline, the results using TFIDF.

First of all, all of our proposed classification methods get higher results than TFIDF, except in the case of F-measure of subjective News data. This shows that using language-specific features which inflect the target language's linguistic characteristics well, without complex mathematical measuring techniques, we could get better results than statistical methods in sentiment classification.

Secondly, similar to the result of movie review corpora, mostly subjective data shows greatly improved results in experimental methods overall. This means that our subjectivity extraction works successfully.

Finally, in contrast to the results of experiment 1, we get higher values of sentiment classification by using chunking and contextual shifters. This implies that the restriction on semantic scope of opinionated terms and the methods reducing features and properly modifying values of polarity terms by using contextual shifters also have merits in sentiment analysis of data such as news which has complex sentence structure like news. Furthermore, this tendency is noticeable particularly in the subjective data of all three groups. This confirms the effectiveness of utilizing linguistic methods in subjectivity extraction and sentiment analysis for news data which tries to maintain objectivity.

5 Discussion and Further Work

In this paper, we verified that simple measurements utilizing language-specific features can improve the results of sentiment analysis. Particularly the chunking method using morphological dependency relations and the lexicon which contains suffixes and particles having important functional meanings is expected to aid the sentiment analysis of other agglutinative languages such as Turkish and Japanese. In addition, this approach of sentiment analysis can be applied to various applications for extracting important information on the Internet to monitor a certain brand's reputations or to make social network for peoples who have similar opinions.

We have plans to confirm the results of this paper by experiments on corpora which are expanded in size and type in future work. We will also increase the number of lexical items of subjectivity lexicon and polarity dictionary. Furthermore, we will utilize other linguistic information such as synonym lists of Korean ontology and elaborate measuring methods using linguistic-specific features of morphologically rich languages effectively.

References

Choi, Y., C. Cardie, E. Riloff, and S. Patwardhan. 2005. Identifying sources of opinions with conditional random fields and extraction patterns. *In Proceedings of the HLT/EMNLP*.

Dave, K., S. Lawrence, and D. M. Pennock. 2003. Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. *In Proceedings of the WWW-2003*.

Hu, Minqing, and Bing Liu. 2004. Mining and summarizingcustomer reviews. *In Proceedings of the KDD*.

Kennedy, A., and D. Inkpen. 2006. Sentiment Classification of Movie and Product Reviews Using Contextual Valence Shifters. *Computational Intelligence*, 22(2):110–125.

Kim, Mi-Yong, and Jong-Hyeok Lee. 2005. Syntactic Analysis based on Subject-Clause Segmentation. *In Proceedings of the KCC 2005*, 32(9):936-947. In Korean.

Kim, S. M., and E. Hovy. 2004. Determining the sentiment of opinions. *In Preceeding of the COLING*.

Liu, Jingjing, and Stephanie Seneff. 2009. Review sentiment scoring via a parse-and-paraphrase paradigm. *In Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, 1(1).

Mao, Y., and G. Lebanon. 2006. Isotonic conditional random fields and local sentiment flow. *In Proceedings of the NIPS*.

McDonald, R., K. Hannan, T. Neylon, M. Wells, and J. Reynar. 2007. Structured Models for Fine-to-Coarse Sentiment Analysis. *In Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, 432—439.

Meena, Arun, and T. V. Prabhakar. 2007. Sentence Level Sentiment Analysis in the Presence of Conjuncts Using Linguistic Analysis. *Lecture Notes in Computer Science*, 573-580. Springer.

Nam, Sang-Hyub, Seung-Hoon Na, Yeha Lee, Yong-Hun Lee, Jungi Kim, and Jong-Hyeok Lee. 2008. Semi-Supeervised Learning for Sentiment Phrase Extraction by Combining Generative Model and Discriminative Model. *In Proceedings of the KCC(Korea Computer Congress) 2008*, 35(1):268-273. in Korean.

Pang, Bo, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up? Sentiment classification using machine learning techniques. *In Proceedings of the ACL-2002 conference on Empirical methods in natural language processing*, 10.

Pang, Bo, and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. *In Proceedings of the ACL-2004*.

Polanyi, Livia, and Annie Zaenen. 2004. Contextual valence shifters. *In Proceedings of the AAAI Symposium on Exploring Attitude and Affect in Text: Theories and Applications*.

Ptaszynski, Michal, Pawel Dybala, Wenhan Shi, Rafal Rzepka, and Kenji Araki. 2010. Contextual affect analysis: a system for verification of emotion appropriateness supported with Contextual Valence Shifters. *International Journal of Biometrics*, 2(2):134-154.

Quirk, R., S. Greenbaum, G. Leech, and J. Svartvik. 1985. *A Comprehensive Grammar of the English Language*. Longman, New York.

Riloff, Ellen, and Janyce Wiebe. 2003. Learning extraction patterns for subjective expressions. *In Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, 105-112.

Tetsuya, Miyoshi, and Nakagami Yu. 2007. Sentiment classification of customer reviews on electric products. *In Proceeding of the IEEE International Conference on Systems Man and Cybernetics*, 2028-2033.

Turney, P. D. 2002. Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. *In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL'02)*, 417-424.

Wiebe, Janyce, Theresa Wilson, Rebecca Bruce, Matthew Bell, and Melanie Martin. 2004. Learning subjective language. *Computational Linguistics*, 30(3).

Wiebe, Janyce, and E. Riloff. 2005. Creating subjective and objective sentence classifiers from unannotated texts. *In Proceedings of the CICLing 2005*, 486-497.

Wilson, Theresa, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis. *In Proceedings of the HLT/EMNLP*, 347-354.

Yu, H., and Hatzivassiloglou V. 2003. Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. *In Proceedings of EMNLP,* 32.

Challenges from Information Extraction to Information Fusion

Heng Ji
Computer Science Department
Queens College and Graduate Center
City University of New York
hengji@cs.qc.cuny.edu

Abstract

Information Extraction (IE) technology is facing new challenges of dealing with large-scale heterogeneous data sources from different documents, languages and modalities. Information fusion, a new emerging area derived from IE, aims to address these challenges. We specify the requirements and possible solutions to perform information fusion. The issues include redundancy removal, contradiction resolution and uncertainty reduction. We believe this is a critical step to advance IE to a higher level of performance and portability.

1 Introduction

Latest development of Information Extraction (IE) techniques has made it possible to extract 'facts' (entities, relations and events) from unstructured documents, and converting them into structured representations (e.g. databases). Once the collection grows beyond a certain size, an issue of critical importance is how a user can monitor a compact knowledge base or identify the interesting portions without having to (re) read large amounts of facts. In this situation users are often more concerned with the speed in which they obtain results, rather than obtaining the exact answers to their queries (Jagadish et al., 1999). The facts extracted from heterogeneous data sources (e.g. text, images, speech and videos) must then be integrated in a knowledge base, so that it can be queried in a uniform way. This provides unparalleled challenges and opportunities for improved decision making.

Data can be noisy, incorrect, or misleading. Unstructured data, mostly text, is difficult to interpret. In practice it is often the case that there are multiple sources which need to be extracted and compressed. In a large, diverse, and interconnected system, it is difficult to assure accuracy or even coherence among the data sources. In this environment, traditional IE would be of little value. Most current IE systems focus on processing a single document and language, and are customized for a single data modality. In addition, automatic IE systems are far from perfect and tend to produce errors.

Achieving really advances in IE requires that we take a broader view, one that looks outside a single source. We feel the time is now ripe to incorporate some information integration techniques in the database community (e.g. Seligman et al., 2010) to extend the IE paradigm to real-time information fusion and raise IE to a higher level of performance and portability. This requires us to work on a more challenging problem of *information fusion* - to remove redundancy, resolve contradictions and uncertainties by multiple information providers and design a general framework for the veracity analysis problem. The goal of this paper is to lay out the current status and potential challenges of information fusion, and suggest the following possible research avenues.

- **Cross-document:** We will discuss how to effectively aggregate facts across documents via entity and event coreference resolution.
- **Cross-lingual:** A shrinking fraction of the world's Web pages are written in English, and so the ability to access pages across a range of languages is becoming increasingly important for many applications. This need can be addressed in part by cross-lingual information fusion. We will discuss the chal-

Coling 2008: Poster Volume, pages 507–515,
Beijing, August 2010

lenges of extraction and translation respectively.

- **Cross-media:** Advances in speech and image processing make the application of IE possible on other data modalities, beyond traditional textual documents.

2 Cross-Document Information Fusion

Most current IE systems focus on processing one document at a time, and except for coreference resolution, operate one sentence at a time. The systems make only limited use of 'facts' already extracted in the current document. The output contains rich structures about entities, relations and events involving such entities. However, due to noise, uncertainty, volatility and unavailability of IE components, the collected facts may be incomplete, noisy and erroneous. Several recent studies have stressed the benefits of using information fusion across documents. These methods investigate quite different angles while follow a common research theme, namely to exploit global background knowledge.

2.1 Information Inference

Achieving really high performance (especially, recall) of IE requires deep semantic knowledge and large costly hand-labeled data. Many systems also exploited lexical gazetteers. However, such knowledge is relatively static (it is not updated during the extraction process), expensive to construct, and doesn't include any probabilistic information. Error analysis on relation extraction shows that a majority (about 78%) of errors occur on nominal mentions, and more than 90% missing errors occur due to the lack of enough patterns to capture the context between two entity mentions. For instance, to describe the "located" relation between a bomber and a bus, there are more than 50 different intervening strings (e.g. "killed many people on a", "'s attack on a", "blew apart a", "blew himself up on a", "drove his explosives-laden car into a", "had rigged the", "set off a bomb on a", etc.), but the ACE[1] training corpora only cover about 1/3 of these expressions.

Several recent studies have stressed the benefits of using information redundancy on estimating the correctness of the IE output (Downey et al., 2005), improving disease event extraction (Yangarber, 2006), Message Understanding Conference event extraction (Mann, 2007; Patwardhan and Riloff, 2009) and ACE event extraction (Ji and Grishman, 2008). This approach is based on the premise that many facts will be reported multiple times from different sources in different forms. This may occur both within the same document and within a cluster of topically related and successive documents. Therefore, by aggregating similar facts across documents and conducting statistical global inference by favoring interpretation consistency, enhanced extraction performance can be achieved with heterogeneous data than uniform data.

The underlying hypothesis of cross-document inference is that the salience of a fact should be calculated by taking into consideration both its confidence and the confidence of other facts connected to it, which is inspired by PageRank (Page et al., 1998) and LexRank (Erkan and Radev, 2004). For example, a vote by linked entities which are highly voted on by other entities is more valuable than a vote from unlinked entities. There are two major heuristics: (1) *an assertion that several information providers agree on is usually more trustable than that only one provider suggests*; and (2) *an information provider is* trustworthy *if it provides many pieces of true information, and a piece of information is likely to be true if it is provided by many trustworthy providers*. (Yin et al., 2008) used the above heuristics in a progressive, iterative enhancement process for information fusion.

The results from the previous work are promising, but the heuristic inferences are highly dependent on the order of applying rules, and the performance may have been limited by the thresholds which may overfit a small development corpus. One promising method might be using Markov Logic Networks (Richardson and Domingos, 2006), a statistical relational learning language, to model these global inference rules more declaratively. Markov Logic will make it possible to compactly specify probability distributions over the complex relational inferences. It can capture non-deterministic (soft) rules that tend to hold among facts but do not have to. Exploiting this approach will also provide greater flexibility to incorporate additional linguistic and world knowledge into inference.

[1] http://www.itl.nist.gov/iad/mig/tests/ace/

The information fused across documents can be represented as an information network (Ji, 2009) in which entities can be viewed as vertices on the graph and they can be connected by some type of static relationship (e.g. those attributes defined in NIST TAC-KBP task (McNamee and Dang, 2009)), or as a temporal chain linking dynamic events (e.g. Bethard and Martin, 2008; Chambers and Jurafsky, 2009; Ji et al., 2009a). The latter representation is more attractive because business or international affairs analysts often review many news reports to track people, companies, and government activities and trends. The query logs from the commercial search engines show that there is a fair number of news related queries (Mishne & de Rijke, 2006), suggesting that blog search users have an interest in the blogosphere response to news stories as they develop. For example, (Ji et al., 2009a) extracted centroid entities and then linked events centered around the same centroid entities on a time line.

Temporal ordering is a challenging task in particular because about half of the event mentions don't include explicit time arguments. The text order by itself is a poor predictor of chronological order (only 3% temporal correlation with the true order). Single-document IE technique can identify and normalize event time arguments from the texts, which results in a much better correlation score of 44% (Ji et al., 2009a). But this is still far from the ideal performance for real applications. In order to alleviate this bottleneck, a possible solution is to exploit global knowledge from the related documents and Wikipedia, and related events to recover and predict some implicit time arguments (Filatova and Hovy, 2001; Mani et al., 2003; Mann, 2007; Eidelman, 2008; Gupta and Ji, 2009).

2.2 Coreference Resolution

One of the key challenges for information fusion is cross-document entity coreference – precise clustering of mentions into correct entities. There are two principal challenges: the same entity can be referred to by more than one name string and the same name string can refer to more than one entity. The recent research has been mainly promoted in the web people search task (Artiles et al., 2007) such as (Balog et al., 2008), ACE2008 such as (Baron and Freedman, 2008) and NIST TAC KBP (McNamee and Dang, 2009) evaluations. Interestingly, the quality of information can often be improved by the fused fact network itself, which can be called as self-boosting of information fusion. For example, if two GPE entities are involved in a "conflict-attack" event, then they are unlikely to be connected by a "part-whole" relation; "Mahmoud Abbas" and "Abu Mazen" are likely to be coreferential if they get involved in the same "life-born" event. Some prior work (Ji et al., 2005; Jing et al., 2007) demonstrated the effectiveness of using semantic relations to improve entity coreference resolution; while (Downey et al., 2005; Sutton and McCallum, 2004; Finkel et al., 2005; Mann, 2007) experimented with information fusion of relations across multiple documents. The TextRunner system (Banko et al., 2007) can collapse and compress redundant facts extracted from multiple documents based on coreference resolution (Yates and Etzioni, 2009), semantic similarity computation and normalization.

Two relations are central for event fusion: *contradiction* – part of one event mention contradicts part of another, and *redundancy* – part of one event mention conveys the same content as (or is entailed by) part of another. Once these central relations are identified they will provide a basis for identifying more complex relations such as elaboration, presupposition or consequence. It is important to note that redundancy and contradiction among event mentions are *logical* relations that are not captured by traditional topic-based techniques for similarity detection (e.g. Brants and Stolle, 2002). Contradictions also arise from complex differences in the structure of assertions, discrepancies based on world-knowledge, and lexical contrasts. Ritter et al. (2009) described a contradiction detection method based on functional relations and pointed out that many contradictory fact pairs from the Web appear consistent, and that requires background knowledge to predict.

Assessing event coreference is essential: for texts to contradict, they must refer to the same event. Event coreference resolution is more challenging than entity coreference because each linking decision needs to be made based upon the overall similarity of the event trigger and multiple arguments. Hasler and Orasan (2009)

further found that in many cases even coreferential even arguments are not good indicators for event coreference.

Earlier work on event coreference resolution (e.g. Bagga and Baldwin, 1999) was limited to several MUC scenarios. Recent work (Chen et al., 2009) focus on much wider coverage of event types defined in ACE. The methods from the knowledge fusion community (e.g. Appriou et al., 2001; Gregoire, 2006) mostly focus on resolving conflicts rather than identifying them (i.e. inconsistency problem rather than ambiguity). These approaches allow the conflicts to be resolved in a straightforward way but they rely on the availability of meta-data (e.g., distribution of weights between attributes, probability assignment etc.). However, it is not always clear where to get this meta-data.

The event attributes such as Modality, Polarity, Genericity and Tense (Sauri et al., 2006) will play an important role in event coreference resolution because two event mentions cannot be coreferential if any of the attributes conflict with each other. Such attempts have been largely neglected in the prior research due to the low weights of attribute labeling in the ACE scoring metric. (Chen et al., 2009) demonstrated that simple automatic event attribute labeling can significantly improve event coreference resolution. In addition, some very recent work including (Nicolae and Nicolae, 2006; Ng, 2009; Chen et al., 2009) found that graph-cut based clustering can improve coreference resolution. The challenge lies in computing the affinity matrix.

3 Cross-Lingual Information Fusion

Cross-lingual comparable corpora are also prevalent now because almost all the influential events can be reported in multi-languages at the first time, but probably in different aspects. Therefore, linked fact networks can be constructed and lots of research tasks can benefit from such structures. Since the two networks are similar in structure but not homogeneous, we can do alignment and translation which may advance information fusion. Cross-lingual information fusion is concerned with technologies that fuse the information available in various languages and present the fused information in the user-preferred language. The following fundamental cross-lingual IE pipelines can be employed: (1) Translate source language texts into target language, and then run target language IE on the translated texts. (2) Run source language IE on the source language texts, and then use machine translation (MT) word alignments to translate (project) extracted information into target languages. Regardless of the different architectures, both pipelines are facing the following challenges from extraction and translation.

3.1 Extraction Challenges

Some recent fusion work focus on cross-lingual interaction and inference to improve both sides synchronously, beyond the parallel comparisons of cross-lingual IE pipelines in (e.g. Riloff et al., 2002). One of such examples is on cross-lingual co-training (e.g. Cao et al., 2003; Chen and Ji, 2009). In co-training (Blum and Mitchell, 1998), the uncertainty of a classifier is defined as the portion of instances on which it cannot make classification decisions. Exchanging tagged data in bootstrapping can help reduce the uncertainties of classifiers. The cross-lingual fusion process satisfies the co-training algorithm's assumptions about two views (in this case, two languages): (1) the two views are individually sufficient for classification (IE systems in both languages were learned from annotated corpora which are enough for reasonable extraction performance); (2) the two views are conditionally independent given the class (IE systems in different languages may use different features and resources).

(Cao et al., 2003) indicated that uncertainty reduction is an important factor for enhancing the performance of co-training. It's important to design new uncertainty measures for representing the degree of uncertainty correlation of the two classifiers in co-training. (Chen and Ji, 2009) proposed a new co-training framework using cross-lingual information projection. They demonstrated that this framework is particularly effective for a challenging IE task which is situated at the end of a pipeline and thus suffers from the errors propagated from upstream processing and has low-performance baseline.

3.2 Translation Challenges

Because the facts are aggregated from multiple languages, the translation errors will bring us great challenges. However, in order to extend

cross-lingual information fusion techniques to more language pairs, we can start from the much more scalable task of "information" translation (Etzioni et al., 2007). The additional processing may take the form of machine translation (MT) of extracted facts such as names and events. IE tasks performed notably worse on machine translated texts than on texts originally written in English, and error analysis indicated that a major cause was the low quality of name translation (Ji et al., 2009b). Traditional MT systems focus on the overall fluency and accuracy of the translation but fall short in their ability to translate certain informationally critical words. In particular, it appears that better entity name translation can substantially improve cross-lingual information fusion.

Some recent work (e.g. Klementiev and Roth, 2006; Ji, 2009) has exploited comparable corpora to enhance information translation. There are no document-level or sentence-level alignments across languages, but important facts such as names, relations and events in one language in such corpora tend to co-occur with their counterparts in the other. (Ji, 2009) used a bootstrapping approach to align the information networks from bilingual comparable corpora, and discover name translations and extract relations links simultaneously. The general idea is to start from a small seed set of common name pairs, and then rely on the link attributes to align their related names. Then the new name translations are added to the seed set for the next iteration. This bootstrapping procedure is repeated until no new translations are produced. This approach is based on graph traverses and doesn't need a name transliteration module to serve as baseline, or compute document-wise temporal distributions.

The novelty of using comparable corpora lies in constructing and mining multi-lingual information fusion framework which is capable of self-boosting. First, this approach can generate information translation pairs with high accuracy by using a small seed set. Second, the shortcomings of traditional approaches are due to their limited use of IE techniques, and this approach can effectively integrate extraction and translation based on reliable confidence estimation. Third, compared to bitexts this approach can take advantage of much less expensive comparable corpora. This approach can be extended to foster the research in other aspects for information fusion. For example, the aligned sub-graphs with names, relations and events can be used to reduce information redundancy; the outlier (misaligned) sub-graphs can be used to detect the novel or local information described in one language but not in the other after the fusion process. It does happen that the two persons have been explicitly reported as Father and Son relationship in one language, but in the other language, they are just reported as two common persons.

4 Cross-Media Information Fusion

The research challenges discussed so far concerned with textual data. Besides written texts, ever-increasing human generated data is available as speech recordings, microblogs, images and videos. We now discuss how to develop techniques for fusing a variety of media sources. State-of-the-art IE techniques have been developed primarily on newspaper articles and a few web texts, and it is not clear how systems would perform on other sources and how to integrate all available information.

4.1 Coreference Resolution

The main challenge is on designing a coherent information fusion framework that is able to exploit information across different parts of multimedia documents and link them via cross-media coreference resolution. The framework will handle multimedia information by considering not only the document's text and images data but also the layout structure which determines how a given text block is related to a particular image or video. For example, a Web news page about "Health Care Reform in America" is composed by text describing some event (e.g., Final Senate vote for the reform plans, Obama signs the reform agreement), images (e.g., images about various government involvements over decades) and videos (e.g. Obama's speech video about the decisions) containing additional information regarding the real extent of the event or providing evidence corroborating the text part.

Current state-of-the-art information fusion approaches can be divided into two groups: formal "top-down" methods from the generic knowledge fusion community and quantitative "bottom-up" techniques from the applied Semantic

Web community (Appriou et al., 2001; Gregoire, 2006). Both approaches have their limitations. It will be beneficial to combine both types of approaches so that the fusion decision can be made depending on the type of problem and the amount of domain information it possesses. Saggion et al. (2004) described a multimedia extraction approach to create composite index from multiple and multi-lingual sources. Magalhaes et al. (2008) described a semantic similarity metric based on key word vectors for multi-media fusion. Iria and Magalhaes (2009) exploited information across different parts of a multimedia document to improve document classification. It is important to go beyond key words and attempt representing the documents by the semantic facts identified by IE.

One possible solution is to exploit the linkage information. Specifically, coreference resolution methods should be applied to four types of cross-media data: (1) between the captions of images and context texts; (2) detecting HTML cross-media associations and quantifying the level of image and text block correlation (3) between the texts embedded in images and context texts; (4) between the transcribed texts from the speech in video clips (via automatic speech recognition) and context texts. We can apply a similarity graph to incorporate virtual linkages. For example, when we see images of two web documents containing the same object, we can raise our confidence that such documents are semantically correlated even if the two web documents are from different sources.

4.2 Uncertainty Reduction

When we combine information from images and their associated texts (e.g. meta-data, captions, surrounding text, transcription), one of the challenges lies in the uncertainty of text representation. Therefore it is important to study both how to learn good models from different sources with different kinds of associated uncertainty, and how to make use of these, along with their level of uncertainty in supporting coherent decisions, taking into account characteristics of the data as well as of its source.

The descriptions are usually generated by humans and thus are prone to error or subjectivity. The images, especially the web images, are typically labeled by different users in different languages and cultural backgrounds. It is unrealistic to expect descriptions to be consistent. In speech conversations, many facts are often embedded in questions such as "*It's OK to put Democratic career politicians at the Pentagon and the Justice Department if they're Democrats but not if they're Republicans, is that right?*" This challenge can be generally addressed by strengthening semantic attribute classification methods for Modality, Polarity and Genericity. And if the data sources are comparable, a more direct method of committee-based voting can also be exploited.

However, the fusion process may itself cause data uncertainties. We can follow the co-training framework as described in section 3.1 to reduce uncertainty in fusion. To handle the missing labels, a promising approach is to use graph-based label propagation (Deshpande et al., 2009), which can capture complex uncertainties and correlations in the data in a uniform manner. It's also worth importing the multi-dimensional uncertainty analysis framework described in data mining community (Aggarwal, 2010). The multi-dimensional uncertainty analysis method exactly suits the multi-media fusion needs: it allows us to combine first-order logic with probabilities, modeling inferential uncertainty about multiple aspects - both the context of facts and intended meanings.

4.3 Joint Modeling

IE is generally applied on top of machine generated transcription and automatic structuring that suffer from errors compared to the true content of relations and events. In the context of information fusion we can divide the problem of adaptation into two types: (1) radical adaptation such as from newswire to biomedical articles; (2) modest adaptation such as from newswire to wikipedia or automatic speech recognition (ASR) output. (1) requires a great deal of new development such as ontology definition and data annotation; while (2) can be partially addressed during the information fusion process.

For example, while dealing with speech input, IE systems need to be robust to the noise introduced by earlier speech processing tasks such as ASR, sentence segmentation, salience detection and and speaker identification. Some earlier work (Makhoul et al., 2005; Favre et al., 2008)

showed that using an IE system trained from newswire, the performance degrades notably when the system is tested on automatic speech recognition output. But no general solutions have been proposed to address the genre-specific challenges for speech data.

More specifically, pronoun resolution is one of the major challenges (Jing et al., 2007). For example, in wikipedia a lot of pronouns may refer to the entry entity; while in speech conversation we will need to resolve first and second person pronouns based on automatic speaker role identification; and improve cross-sentence third pronoun resolution by exploiting gender and animacy knowledge discovery methods.

The processing methods of text and other media are typically organized as a pipeline architecture of processing stages (e.g. from pattern recognition, to information fusion, and to summarization). Each of these stages has been studied separately and quite intensively over the past decade. It's critical to move away from approaches that make chains of independent local decisions, and instead toward methods that make multiple decisions jointly using global information. Joint inference techniques (Roth and Yih, 2004; Ji et al., 2005; McCallum, 2006) can transform the integration of multi-media into a benefit by reducing the errors in individual stages. In doing so, we can take advantage (among other properties) of the coherence of a discourse: that a correct analysis of a text discourse reveals a large number of connections from the image information in its context, and so (in general) a more tightly connected analysis is more likely to be correct. For example, prior work has demonstrated the benefit of jointly modeling name tagging and n-best hypotheses, ASR lattices or word confusion networks (Hakkani-Tür et al., 2006).

5 Conclusion

In the current information explosion era, IE technology is facing new challenges of dealing with heterogeneous data sources from different documents, languages and media which may contain a multiplicity of aspects on particular entities, relations and events. This new phenomena requires IE to perform both traditional lower level processing as well as information fusion of factual data based on implicit inferences. This paper investigated the issues of information fusion on a massive scale and the challenges have not been discussed in previous work. We specified the requirements and possible solutions for various dimensions to perform information fusion. We also overviewed some recent work to demonstrate how these goals can be achieved.

The field of information fusion is relatively new; and the nature of different data sources provides new ideas and challenges which are not present in other research. While much research has been performed in the area of data fusion, the context of automatic extraction provides a different perspective in which the fusion is performed in the context of a lot of uncertainty and noise. This new task will provide connections between NLP and other areas such as data mining and knowledge discovery. The progress on this task would save, anybody concerned with staying informed, an enormous amount of time. These are certainly ambitious goals and require long-term development of fusion and adaptation methods. But we hope that this outline of the research challenges will bring us closer to the goal.

Acknowledgement

This work was supported by the U.S. Army Research Laboratory under Cooperative Agreement Number W911NF-09-2-0053, the U.S. NSF CAREER Award under Grant IIS-0953149, Google, Inc., DARPA GALE Program, CUNY Research Enhancement Program, PSC-CUNY Research Program, Faculty Publication Program and GRTI Program. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

References

Charu Aggarwal. 2010. On Multi-dimensional Sharpening of Uncertain Data. *SIAM: SIAM Conference on Data Mining (SDM10).*

A. Appriou-, A. Ayoun, et al. 2001. Fusion: General concepts and characteristics. *International Journal of Intelligent Systems* 16(10).

Javier Artiles, Julio Gonzalo and Satoshi Sekine. 2007. The SemEval-2007 WePS Evaluation: Establishing a benchmark for the Web People Search Task. *Proc. Semeval-2007.*

Amit Bagga and Breck Baldwin. 1999. Cross-document Event Coreference: Annotations, Experiments, and Observations. *Proc. ACL1999 Workshop on Coreference and Its Applications*.

K. Balog, L. Azzopardi, M. de Rijke. 2008. Personal Name Resolution of Web People Search. *Proc. WWW2008 Workshop: NLP Challenges in the Information Explosion Era (NLPIX 2008).*

Michele Banko, Michael J Cafarella, Stephen Soderland and Oren Etzioni. 2007. Open Information Extraction from the Web. *Proc. IJCAI 2007*.

Alex Baron and Marjorie Freedman. 2008. Who is Who and What is What: Experiments in Cross-Document Co-Reference. *Proc. EMNLP 2008*.

Steven Bethard and James H. Martin. 2008. Learning semantic links from a corpus of parallel temporal and causal relations. *Proc. ACL-HLT 2008*.

Avrim Blum and Tom Mitchell. 1998. Combining Labeled and Unlabeled Data with Co-training. *Proc. of the Workshop on Computational Learning Theory.* Morgan Kaufmann Publishers.

T. Brants and R. Stolle. 2002. Finding Similar Documents in Document Collections. *Proc. LREC Workshop on Using Semantics for Information Retrieval and Filtering.*

Yunbo Cao, Hang Li and Li Lian. 2003. Uncertainty Reduction in Collaborative Bootstrapping: Measure and Algorithm. *Proc. ACL 2003*.

Nathanael Chambers and Dan Jurafsky. 2009. Unsupervised Learning of Narrative Schemas and their Participants. *Proc. ACL 09.*

Zheng Chen and Heng Ji. 2009. Can One Language Bootstrap the Other: A Case Study on Event Extraction. *Proc. HLT-NAACL Workshop on Semi-supervised Learning for Natural Language Processing*. Boulder, Co.

Zheng Chen, Heng Ji and Robert Harallick. 2009. A Pairwise Coreference Model, Feature Impact and Evaluation for Event Coreference Resolution. *Proc. RANLP 2009 workshop on Events in Emerging Text Types.*

Amol Deshpande, Lise Getoor and Prithviraj Sen. 2009. Graphical Models for Uncertain Data. *Managing and Mining Uncertain Data (Edited by Charu Aggarwal).* Springer.

Doug Downey, Oren Etzioni, and Stephen Soderland. 2005. A Probabilistic Model of Redundancy in Information Extraction. *Proc. IJCAI 2005*.

Vladimir Eidelman. 2008. Inferring Activity Time in News through Event Modeling. *Proc. ACL-HLT 2008*.

Gunes Erkan and Dragomir R. Radev. 2004. LexPageRank: Prestige in multi-document text summarization. *Proc. EMNLP 2004.*

Oren Etzioni, Kobi Reiter, Stephen Soderland and Marcus Sammer. 2007. Lexical Translation with Application to Image Search on the Web. *Proc. Machine Translation Summit XI*.

Benoit Favre, Ralph Grishman, Dustin Hillard, Heng Ji, Dilek Hakkani-Tur and Mari Ostendorf. 2008. Punctuating Speech for Information Extraction. *Proc. ICASSP 2008*.

Elena Filatova and Eduard Hovy. 2001. Assigning Time-Stamps to Event-Clauses. *Proc. ACL 2001 Workshop on Temporal and Spatial Information Processing.*

Jenny Rose Finkel, Trond Grenager and Christopher Manning. 2005. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. *Proc. ACL 2005*.

E. Gregoire. 2006. An unbiased approach to iterated fusion by weakening. Information Fusion. 7(1).

Prashant Gupta and Heng Ji. 2009. Predicting Unknown Time Arguments based on Cross-event propagation. *Proc. ACL-IJCNLP 2009.*

Dilek Hakkani-Tür, Frédéric Béchet, Giuseppe Riccardi, Gokhan Tur. 2006. Beyond ASR 1-Best: Using Word Confusion Networks in Spoken Language Understanding. *Journal of Computer Speech and Language*, Vol. 20, No. 4, pp. 495-514.

Laura Hasler and Constantin Orasan. 2009. Do coreferential arguments make event mentions coreferential? Proc. the *7th Discourse Anaphora and Anaphor Resolution Colloquium (DAARC 2009)*.

Jose Iria and Joao Magalhaes. 2009. Exploiting Cross-Media Correlations in the Categorization of Multimedia Web Documents. *Proc. CIAM 2009*.

H. V. Jagadish, Jason Madar, and Raymond Ng. 1999. Semantic compression and pattern extraction with fascicles. *VLDB*, pages 186–197.

Heng Ji, David Westbrook and Ralph Grishman. 2005. Using Semantic Relations to Refine Coreference Decisions. *Proc. HLT/EMNLP 05.*

Heng Ji and Ralph Grishman. 2008. Refining Event Extraction Through Cross-document Inference. *Proc. ACL 2008.*

Heng Ji. 2009. Mining Name Translations from Comparable Corpora by Creating Bilingual Information Networks. *Proc. ACL-IJCNLP 2009 workshop on Building and Using Comparable Corpora (BUCC 2009): from parallel to non-parallel corpora.*

Heng Ji, Ralph Grishman, Dayne Freitag, Matthias Blume, John Wang, Shahram Khadivi, Richard Zens and Hermann Ney. 2009a. Name Transla-

tion for Distillation. Book chapter for *Global Automatic Language Exploitation*.

Heng Ji, Ralph Grishman, Zheng Chen and Prashant Gupta. 2009b. Cross-document Event Extraction, Ranking and Tracking. *Proc. RANLP 2009.*

Hongyan Jing, Nanda Kambhatla and Salim Roukos. 2007. Extracting Social Networks and Biographical Facts From Conversational Speech Transcripts. *Proc. ACL 2007*.

A. Klementiev and D. Roth. 2006. Named Entity Transliteration and Discovery from Multilingual Comparable Corpora. *Proc. HLT-NAACL 2006*.

Joao Magalhaes, Fabio Ciravegna and Stefan Ruger. 2008. Exploring Multimedia in a Keyword Space. *Proc. ACM Multimedia*.

Inderjeet Mani, Barry Schiffman and Jianping Zhang. 2003. Inferring Temporal Ordering of Events in News. *Proc. HLT-NAACL 2003.*

John Makhoul, Alex Baron, Ivan Bulyko, Long Nguyen, Lance Ramshaw, David Stallard, Richard Schwartz and Bing Xiang. 2005. The Effects of Speech Recognition and Punctuation on Information Extraction Performance. *Proc. Interspeech*.

Gideon Mann. 2007. Multi-document Relationship Fusion via Constraints on Probabilistic Databases. *Proc. HLT/NAACL 2007*.

Andrew McCallum. 2006. Information Extraction, Data Mining and Joint Inference. *Proc. SIGKDD*.

Paul McNamee and Hoa Dang. 2009. Overview of the TAC 2009 Knowledge Base Population Track. *Proc. TAC 2009 Workshop*.

Gilad Mishne and Maarten de Rijke. 2006. Capturing Global Mood Levels using Blog Posts. *Proc. AAAI 2006 Spring Symposium on Computational Approaches to Analysing Weblogs.*

Vincent Ng. 2009. Graph-Cut-Based Anaphoricity Determination for Coreference Resolution. *Proc. HLT-NAACL 2009.*

Lawrence Page, Sergey Brin, Rajeev Motwani and Terry Winograd. 1998. The PageRank Citation Ranking: Bringing Order to the Web. *Proc. WWW.*

Siddharth Patwardhan and Ellen Riloff. 2009. A Unified Model of Phrasal and Sentential Evidence for Information Extraction. 2009. *Proc. EMNLP.*

Matt Richardson and Pedro Domingos. 2006. Markov Logic Networks. *Machine Learning*. 62:107-136.

Ellen Riloff, Charles Schafer, and David Yarowsky. 2002. Inducing Information Extraction Systems for New Languages via Cross-Language Projection. *Proc. COLING 2002*.

Alan Ritter; Stephen Soderland; Doug Downey; Oren Etzioni. 2009. It's a Contradiction – no, it's not: A Case Study using Functional Relations. *Proc. EMNLP 2009.*

Dan Roth and Wen-tau Yih. 2004. A Linear Programming Formulation for Global Inference in Natural Language Tasks. *Proc. CONLL2004*.

Saggion, H., Cunningham, H., Bontcheva, K., Maynard, D., Hamza, O., and Wilks, Y. 2004. Multimedia indexing through multi-source and multi-language information extraction: the MUMIS project. *Data Knowlege Engineering*, 48, 2, pp. 247-264.

Roser Saurı and Marc Verhagen and James Pustejovsky. 2006. Annotating and Recognizing Event Modality in Text. *Proc. FLAIRS 2006*.

Len Seligman, Peter Mork, Alon Halevy, Ken Smith, Michael J. Carey, Kuang Chen, Chris Wolf, Jayant Madhavan and Akshay Kannan. 2010. OpenII: An Open Source Information Integration Toolkit. *Proc. the 2010 international conference on Management of data.*

Charles Sutton and Andrew McCallum. 2004. Collective Segmentation and Labeling of Distant Entities in Information Extraction. *Proc. ICML Workshop on Statistical Relational Learning and Its Connections to Other Fields.*

Roman Yangarber. 2006. Verification of Facts across Document Boundaries. *Proc. International Workshop on Intelligent Information Access*.

Alexander Yates and Oren Etzioni. 2009. Unsupervised Methods for Determining Object and Relation Synonyms on the Web. *Journal of Artificial Intelligence*. Res. (*JAIR*) 34: 255-296.

Xiaoxin Yin, Jiawei Han and Philip S. Yu. 2008. Truth Discovery with multiple conflicting information providers on the web. *IEEE Trans. Knowledge and Data Eng.*, 20:796-808.

Effective Constituent Projection across Languages

Wenbin Jiang and **Yajuan Lü** and **Yang Liu** and **Qun Liu**

Key Laboratory of Intelligent Information Processing
Institute of Computing Technology
Chinese Academy of Sciences
{jiangwenbin, lvyajuan, yliu, liuqun}@ict.ac.cn

Abstract

We describe an effective constituent projection strategy, where constituent projection is performed on the basis of dependency projection. Especially, a novel measurement is proposed to evaluate the candidate projected constituents for a target language sentence, and a PCFG-style parsing procedure is then used to search for the most probable projected constituent tree. Experiments show that, the parser trained on the projected treebank can significantly boost a state-of-the-art supervised parser. When integrated into a tree-based machine translation system, the projected parser leads to translation performance comparable with using a supervised parser trained on thousands of annotated trees.

1 Introduction

In recent years, supervised constituent parsing has been well studied and achieves the state-of-the-art for many resource-rich languages (Collins, 1999; Charniak, 2000; Petrov et al., 2006). Because of the cost and difficulty in treebank construction, researchers have also investigated the utilization of unannotated text, including the unsupervised parsing which totally uses unannotated data (Klein and Manning, 2002; Klein and Manning, 2004; Bod, 2006; Seginer, 2007), and the semi-supervised parsing which uses both annotated and unannotated data (Sarkar, 2001; Steedman et al., 2003; McClosky et al., 2006).

Because of the higher complexity and lower performance of unsupervised methods, as well as the need of reliable priori knowledge in semi-supervised methods, it seems promising to project the syntax structures from a resource-rich language to a resource-scarce one across a bilingual corpus. Lots of researches have so far been devoted to dependency projection (Hwa et al., 2002; Hwa et al., 2005; Ganchev et al., 2009; Smith and Eisner, 2009). While for constituent projection there is few progress. This is due to the fact that the constituent syntax describes the language structure in a more detailed way, and the degree of isomorphism between constituent structures appears much lower.

In this paper we propose for constituent projection a stepwise but totally automatic strategy, which performs constituent projection on the basis of dependency projection, and then use a constraint EM optimization algorithm to optimized the initially projected trees. Given a word-aligned bilingual corpus with source sentences parsed, we first project the dependency structures of these constituent trees to the target sentences using a dynamic programming algorithm, then we generate a set of candidate constituents for each target sentence and design a novel evaluation function to calculate the probability of each candidate constituent, finally, we develop a PCFG-style parsing procedure to search for the most probable projected constituent tree in the evaluated candidate constituent set. In addition, we design a constraint EM optimization procedure to decrease the noise in the initially projected constituent treebank.

Experimental results validate the effectiveness of our approach. On the Chinese-English FBIS corpus, we project the English parses produced by the Charniak parser across to the Chinese sen-

Coling 2008: Poster Volume, pages 516–524,
Beijing, August 2010

tences. A berkeley parser trained on this projected treebank can effectively boost the supervised parsers trained on bunches of CTB trees. Especially, the supervised parser trained on the smaller CTB 1.0 benefits a significant F-measure increment of more than 1 point from the projected parser. When using the projected parser in a tree-based translation model (Liu et al., 2006), we achieve translation performance comparable with using a state-of-the-art supervised parser trained on thousands of CTB trees. This surprising result gives us an inspiration that better translation would be achieved by combining both projected parsing and supervised parsing into a hybrid parsing schema.

2 Stepwise Constituent Projection

We first introduce the dynamic programming procedure for dependency projection, then describe the PCFG-style algorithm for constituent projection which is conducted on projected dependent structures, and finally show the constraint EM procedure for constituent optimization.

2.1 Dependency Projection

For dependency projection we adopt a dynamic programming algorithm, which searches the most probable projected target dependency structure according to the source dependency structure and the word alignment.

In order to mitigate the effect of word alignment errors, multiple GIZA++ (Och and Ney, 2000) results are combined into a compact representation called alignment matrix. Given a source sentence with m words, represented as $E_{1:m}$, and a target sentence with n words, represented as $F_{1:n}$, their word alignment matrix A is an $m \times n$ matrix, where each element $A_{i,j}$ denotes the probability of the source word E_i aligned to the target word F_j.

Using $P(D_F|D_E, A)$ to denote the probability of the projected target dependency structure D_F conditioned on the source dependency structure D_E and the alignment matrix A, the projection algorithm aims to find

$$\tilde{D_F} = \underset{D_F}{\operatorname{argmax}} P(D_F|D_E, A) \quad (1)$$

Algorithm 1 Dependency projection.

1: **Input**: F, and P_e for all word pairs in F
2: **for** $\langle i, j\rangle \subseteq \langle 1, |F|\rangle$ in topological order **do**
3: $\quad \mathbf{buf} \leftarrow \emptyset$
4: $\quad$ **for** $k \leftarrow i..j-1$ **do** ▷ all partitions
5: $\quad\quad$ **for** $l \in \mathbf{V}[i,k]$ and $r \in \mathbf{V}[k+1,j]$ **do**
6: $\quad\quad\quad$ insert DERIV(l, r, P_e) into $\mathbf{buf}$
7: $\quad\quad\quad$ insert DERIV(r, l, P_e) into $\mathbf{buf}$
8: $\quad \mathbf{V}[i,j] \leftarrow$ top K derivations of $\mathbf{buf}$
9: **Output:** the best derivation of $\mathbf{V}[1, |F|]$
10: **function** DERIV(p, c, P_e)
11: $\quad d \leftarrow p \cup c \cup \{p \cdot root \curvearrowright c \cdot root\}$ ▷ new derivation
12: $\quad d \cdot evl \leftarrow$ EVAL(d, P_e) ▷ evaluation function
13: $\quad$ **return** d

$P(D_F|D_E, A)$ can be factorized into each dependency edge $x \curvearrowright y$ in D_F

$$P(D_F|D_E, A) = \prod_{x \curvearrowright y \in D_F} P_e(x \curvearrowright y|D_E, A)$$

P_e can then be obtained by simple accumulation across all possible situations of correspondence

$$P_e(x \curvearrowright y|D_E, A) = \sum_{1 \le x', y' \le |E|} A_{x,x'} \times A_{y,y'} \times \delta(x', y'|D_E)$$

where $\delta(x', y'|D_E)$ is a 0-1 function that equals 1 only if the dependent relation $x' \curvearrowright y'$ holds in D_E.

The search procedure needed by the argmax operation in equation 1 can be effectively solved by the Chu-Liu-Edmonds algorithm used in (McDonald et al., 2005). In this work, however, we adopt a more general and simple dynamic programming algorithm as shown in Algorithm 1, in order to facilitate the possible expansions. In practice, the cube-pruning strategy (Huang and Chiang, 2005) is used to speed up the enumeration of derivations (loops started by line 4 and 5).

2.2 Constituent Projection

The PCFG-style parsing procedure searches for the most probable projected constituent tree in a shrunken search space determined by the projected dependency structure and the target constituent tree. The shrunken search space can be built as following. First, we generates the candidate constituents of the source tree and the candidate spans of the target sentence, so as to enumerate the candidate constituents of the target sentence. Then we compute the consistent degree for

each pair of candidate constituent and span, and further estimate the probability of each candidate constituent for the target sentence.

2.2.1 Candidate Constituents and Spans

For the candidate constituents of the source tree, using only the original constituents imposes a strong hypothesis of isomorphism on the constituent projection between two languages, since it requires that each couple of constituent and span must be strictly matched. While for the candidate spans of the target sentences, using all subsequences makes the search procedure suffer from more perplexity. Therefore, we expand the candidate constituent set and restrict the candidate span set:

- Candidate Constituent: Suppose a production in the source constituent tree, denoted as $p \rightarrow c_1 c_2 .. c_h .. c_{|p|}$, and c_h is the head child of the parent p. Each constituent, p or c, is a triple $\langle lb, rb, nt \rangle$, where nt denotes its non-terminal, while lb and rb represent its left- and right bounds of the sub-sequence that the constituent covers. The candidate constituent set of this production consists the head of the production itself, and a set of incomplete constituents,

$$\{\langle l, r, p \cdot nt* \rangle | c_1 \cdot lb \leq l \leq c_h \cdot lb \wedge \\ c_h \cdot rb \leq r \leq c_{|p|} \cdot rb \wedge \\ (l < c_h \cdot lb \vee r > c_h \cdot rb)\}$$

  where the symbol $*$ indicates an incomplete non-terminal. The candidate constituent set of the entire source tree is the unification of the sets extracted from all productions of the tree.

- Candidate Span: A candidate span of the target sentence is a tuple $\langle lb, rb \rangle$, where lb and rb indicate the same as in a constituent. We define the candidate span set as the spans of all *regular dependent segment*s in the corresponding projected dependency structure. A regular dependency segment is a dependent segment that every modifier of the root is a complete dependency structure. Suppose a dependency structure rooted at word p, denoted as $c_{lL} .. c_{l2} c_{l1} \curvearrowleft p \curvearrowright c_{r1} c_{r2} .. c_{rR}$, it has L ($L \geq 0$) modifiers on its left and R ($R \geq 0$) modifiers on its right, each of them is a smaller complete dependency structure. Then the word p itself is a regular dependency segment without any modifier, and

$$\{c_{li} .. c_{l1} \curvearrowleft p \curvearrowright c_{r1} .. c_{rj} | 0 \leq i \leq L \wedge \\ 0 \leq j \leq R \wedge \\ (i > 0 \vee j > 0)\}$$

  is a set of regular dependency structures with at least one modifier. The regular dependency segments of the entire projected dependency structure can simply be accumulated across all dependency nodes.

2.2.2 Span-to-Constituent Correspondence

After determining the candidate constituent set of the source tree, denoted as Φ_E, and the candidate span set of the target sentence, denoted as Ψ_F, we then calculate the consistent degree for each pair of candidate constituent and candidate span.

Given a candidate constituent $\phi \in \Phi_E$ and a candidate span $\psi \in \Psi_F$, their consistent degree $\mathcal{C}(\psi, \phi | A)$ is the probability that they are aligned to each other according to A.

We display the derivations from bottom to up. First, we define the alignment probability from a word i in the span ψ to the constituent ϕ as

$$P(i \mapsto \phi | A) = \frac{\sum_{\phi \cdot lb \leq j \leq \phi \cdot rb} A_{i,j}}{\sum_j A_{i,j}}$$

Then we define the alignment probability from the span ψ to the constituent ϕ as

$$P(\psi \mapsto \phi | A) = \prod_{\psi \cdot lb \leq i \leq \psi \cdot rb} P(i \mapsto \phi | A)$$

Note that we use i to denote both a word and its index for simplicity without causing confusion. Finally, we define $\mathcal{C}(\phi, \psi | A)$ as

$$\mathcal{C}(\psi, \phi | A) = P(\psi \mapsto \phi | A) \times P(\phi \mapsto \psi | A^T) \quad (2)$$

Where $P(\phi \mapsto \psi | A^T)$ denotes the alignment probability from the constituent ϕ to the span ψ, it can be calculated in the same manner.

2.2.3 Constituent Projection Algorithm

The purpose of constituent projection is to find the most probable projected constituent tree for the target sentence conditioned on the source constituent tree and the word alignment

$$\tilde{T_F} = \underset{T_F \subseteq \Phi_F}{\operatorname{argmax}} P(T_F|T_E, A) \quad (3)$$

Here, we use Φ_F to denote the set of candidate constituents of the target sentence

$$\begin{aligned}\Phi_F &= \Psi_F \otimes NT(\Phi_E) \\ &= \{\phi_F | \psi(\phi_F) \in \Psi_F \wedge nt(\phi_F) \in NT(\Phi_E)\}\end{aligned}$$

where $\psi(\cdot)$ and $nt(\cdot)$ represent the span and the non-terminal of a constituent respectively, and $NT(\cdot)$ represents the set of non-terminals extracted from a constituent set. Note that T_F is a subset of Φ_F if we treat a tree as a set of constituents.

The probability of the projected tree T_F can be factorized into the probabilities of the projected constituents that composes the tree

$$P(T_F|T_E, A) = \prod_{\phi_F \in T_F} P_\phi(\phi_F|T_E, A)$$

while the probability of the projected source constituent can be defined as a statistics of span-to-constituent- and constituent-to-constituent consistent degrees

$$P_\phi(\phi_F|T_E, A) = \frac{\sum_{\phi_E \in \Phi_E} \mathcal{C}(\phi_F, \phi_E|A)}{\sum_{\phi_E \in \Phi_E} \mathcal{C}(\psi(\phi_F), \phi_E|A)}$$

where $\mathcal{C}(\phi_F, \phi_E|A)$ in the numerator denotes the consistent degree for each pair of constituents, which can be calculated based on that of span and constituent described in Formula 2

$$\mathcal{C}(\phi_F, \phi_E) = \begin{cases} 0 & \textbf{if } \phi_F \cdot nt \neq \phi_E \cdot nt \\ \mathcal{C}(\psi(\phi_F), \phi_E) & \textbf{else} \end{cases}$$

Algorithm 2 shows the pseudocode for constituent projection. A PCFG-style parsing procedure searches for the best projected constituent tree in the constrained space determined by Ψ_F. Note that the projected trees are binarized, and can be easily recovered according to the asterisks at the tails of non-terminals.

Algorithm 2 Constituent projection.

Input: Ψ_F, Φ_F, and P_ϕ for all spans in Ψ_F
for $\langle i, j \rangle \in \Psi$ in topological order **do**
  $\mathbf{buf} \leftarrow \emptyset$
  for $p \in \Phi_F$ s.t. $\psi(p) = \langle i, j \rangle$ **do**
    for $k \leftarrow i..j-1$ **do** ▷ all partitions
      for $l \in \mathbf{V}[i, k]$ and $r \in \mathbf{V}[k+1, j]$ **do**
        insert DERIV(l, r, p, P_ϕ) into $\mathbf{buf}$
  $\mathbf{V}[i, j] \leftarrow$ top K derivations of $\mathbf{buf}$
Output: the best derivation of $\mathbf{V}[1, |F|]$
function DERIV(l, r, p, P_ϕ)
  $d \leftarrow l \cup r \cup \{p\}$ ▷ new derivation
  $d \cdot evl \leftarrow$ EVAL(d, P_ϕ) ▷ evaluation function
  return d

2.3 EM Optimization

Since the constituent projection is conducted on each sentence pair separately, the projected treebank is apt to suffer from more noise caused by free translation and word alignment error. It can be expected that an EM iteration over the whole projected treebank will lead to trees with higher consistence.

We adopt the inside-outside algorithm to improve the quality of the initially projected treebank. Different from previous works, all expectation and maximization operations for a single tree are performed in a constrained space determined by the candidate span set of the projected target dependency structure. That is to say, all the summation operations, both for calculating α/β values and for re-estimating the rule probabilities, only consider the spans in the candidate span set. This means that the projected dependency structures are supposed believable, and the noise is mainly introduced in the following constituent projection procedure.

Here we give an overall description of the treebank optimization procedure. First, an initial PCFG grammar G_F^0 is estimated from the original projected treebank. Then several iterations of α/β calculation and rule probability re-estimation are performed. For example in the i-the iteration, α/β values are calculated based on the current grammar G_F^{i-1}, afterwards the optimized grammar G_F^i is obtained based on these α/β values. The iterative procedure terminates when the likelihood of whole treebank increases slowly. Finally, with the optimized grammar, a constrained PCFG parsing procedure is conducted on each of the initial pro-

jected trees, so as to obtain an optimized treebank.

3 Applications of Constituent Projection

The most direct contribution of constituent projection is pushing an initial step for the statistical constituent parsing of resource-scarce languages. It also has some meaningful applications even for the resource-rich languages. For instances, the projected treebank, due to its large scale and high coverage, can used to boost an traditional supervised-trained parser. And, the parser trained on the projected treebank can adopted to conduct tree-to-string machine translation, since it give parsing results with larger isomorphism with the target language than a supervised-trained parser dose.

3.1 Boost an Traditional Parser

We first establish a unified framework for the enhanced parser where a projected parser is adopted to guide the parsing procedure of the baseline parser.

For a given target sentence S, the enhanced parser selected the best parse $\tilde{T}$ among the set of candidates $\Omega(S)$ according to two evaluation functions, given by the baseline parser $\mathbb{B}$ and the projected guide parser $\mathbb{G}$, respectively.

$$\tilde{T} = \underset{T \in \Omega(S)}{\operatorname{argmax}} P(T|\mathbb{B}) \times P(T|\mathbb{G})^{\lambda} \quad (4)$$

These two evaluation functions can be integrated deeply into the decoding procedure (Carreras et al., 2008; Zhang and Clark, 2008; Huang, 2008), or can be integrated at a shallow level in a reranking manner (Collins, 2000; Charniak and Johnson, 2005). For simplicity and generability, we adopt the reranking strategy. In k-best reranking, $\Omega(S)$ is simply a set of candidate parses, denoted as $\{T_1, T_2, ..., T_k\}$, and we use the single parse of the guide parser, $T_{\mathbb{G}}$, to re-evaluate these candidates. Formula 4 can be redefined as

$$\tilde{T}(T_{\mathbb{G}}) = \underset{T \in \Omega(S)}{\operatorname{argmax}} \mathbf{w} \cdot \mathbf{f}(T, T_{\mathbb{G}}) \quad (5)$$

Here, $\mathbf{f}(T, T_{\mathbb{G}})$ and $\mathbf{w}$ represent a high dimensional feature representation and a corresponding weight vector, respectively. The first feature $f_1(T, T_{\mathbb{G}}) = logP(T|\mathbb{B})$ is the log probability of the baseline parser, while the remaining features are integer-valued guide features, and each of them represents the guider parser's predication result for a particular configuration in candidate parse T, so as to utilize the projected parser's knowledge to guide the parsing procedure of the traditional parser.

In our work a guide feature is composed of two parts, the non-terminal of a certain constituent ϕ in the candidate parse T,[1] and the non-terminal at the corresponding span $\psi(\phi)$ in the projected parse $T_{\mathbb{G}}$. Note that in the projected parse this span does not necessarily correspond to a constituent. In such situations, we simply use the non-terminal of the constituent that just be able to cover this span, and attach a asterisk at the tail of this non-terminal. Here is an example of the guide features

$$f_{100}(T, T_{\mathbb{G}}) = VP \in T \circ PP* \in T_{\mathbb{G}}$$

It represents that a VP in the candidate parse corresponds to a segment of a PP in the projected parse. The quantity of its weight w_{100} indicates how probably a span can be predicated as VP if the span corresponds to a partial PP in the projected parse.

We adopt the perceptron algorithm to train the reranker. To reduce overfitting and produce a more stable weight vector, we also use a refinement strategy called averaged parameters (Collins, 2002).

3.2 Using in Machine Translation

Researchers have achieved promising improvements in tree-based machine translation (Liu et al., 2006; Huang et al., 2006). Such models use a parsed tree as input and converts it into a target tree or string. Given a source language sentence, first we use a traditional source language parser to parse the sentence to obtain the syntax tree T, and then use the translation decoder to search for the best derivation $\tilde{d}$, where a derivation d is a sequence of transformations that converts the source tree into the target language string

$$\tilde{d} = \underset{d \in D}{\operatorname{argmax}} P(d|T) \quad (6)$$

[1] Using non-terminals as features brings no improvement in the reranking experiments, so as to examine the impact of the projected parser.

Here D is the candidate set of d, and it is determined by the source tree T and the transformation rules.

Since the tree-based models are based on the synchronous transformational grammars, they suffer much from the isomerism between the source syntax and the target sentence structure. Considering that the parsed tree produced by a projected parser may have larger isomorphism with the target language, it would be a promising idea to adopt the projected parser to parse the input sentence for the subsequent translation decoding procedure.

4 Experiments

In this section, we first invalidate the effect of constituent projection by evaluating a parser trained on the projected treebank. Then we investigate two applications of the projected parser: boosting an traditional supervised-trained parser, and integration in a tree-based machine translation system. Following the previous works, we depict the parsing performance by F-score on sentences with no more than 40 words, and evaluate the translation quality by the case-sensitive BLEU-4 metric (Papineni et al., 2002) with 4 references.

4.1 Constituent Projection

We perform constituent projection from English to Chinese on the FBIS corpus, which contains 239K sentence pairs with about 6.9M/8.9M words in Chinese/English. The English sentences are parsed by the Charniak Parser and the dependency structures are extracted from these parses according to the head-finding rules of (Yamada and Matsumoto, 2003). The word alignment matrixes are obtained by combining the 10-best results of GIZA++ according to (Liu et al., 2009).

We first project the dependency structures from English to Chinese according to section 2.1, and then project the constituent structures according to section 2.2. We define an assessment criteria to evaluate the confidence of the final projected constituent tree

$$c = \sqrt[n]{P(D_F|D_E, A) \times P(T_F|T_E, A)}$$

where n is the word count of a Chinese sentence in our experiments. A series of projected Chinese treebanks with different scales are obtained by specifying different c as the filtering threshold. The state-of-the-art Berkeley Parser is adopted to train on these treebanks because of its high performance and independence of head word information.

Thres c	**#Resrv**	**Cons-**F_1	**Span-**F_1
0.5	12.6K	23.9	32.7
0.4	17.8K	23.9	33.4
0.3	27.2K	25.4	35.7
0.2	45.1K	26.6	38.0
0.1	87.0K	27.8	40.4

Table 1: Performances of the projected parsers on the CTB test set. #Resrv denotes the amount of reserved trees within threshold c. Cons-F_1 is the traditional F-measure, while Span-F_1 is the F-measure without consideration of non-terminals.

Table 1 shows the performances of these projected parsers on the standard CTB test set, which is composed of sentences in chapters 271-300. We find that along with the decrease of the filtering threshold c, more projected trees are reserved and the performance of the projected parser constantly increases. We also find that the traditional F-value, Cons-F_1, is obviously lower than the one without considering non-terminals, Span-F_1. This indicates that the constituent projection procedure introduces more noise because of the higher complexity of constituent correspondence. In all the rest experiments, however, we simply use the projected treebank filtered by threshold $c = 0.1$ and do not try any smaller thresholds, since it already takes more than one weak to train the Berkeley Parser on the 87 thousands trees resulted by this threshold.

The constrained EM optimization procedure described in section 2.3 is used to alleviate the noise in the projected treebank, which may be caused by free translation, word alignment errors, and projection on each single sentence pair. Figure 1 shows the log-likelihood on the projected treebank after each EM iteration. It is obvious that the log-likelihood increases very slowly after 10 iterations. We terminate the EM procedure after 40 iterations.

Finally we train the Berkeley Parser on the optimized projected treebank, and test its perfor-

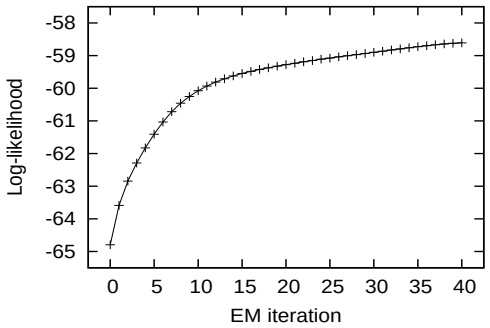


Figure 1: Log-likelihood of the 87K-projected treebank after each EM interation.

Train Set	Cons-F_1	Span-F_1
Original 87K	27.8	40.4
Optimized 87K	22.8	40.2

Table 2: Performance of the parser trained on the optimized projected treebank, compared with that of the original projected parser.

Train Set	Baseline	Bst-Ini	Bst-Opt
CTB 1.0	75.6	76.4	76.9
CTB 5.0	85.2	85.5	85.7

Table 3: Performance improvement brought by the projected parser to the baseline parsers trained on CTB 1.0 and CTB 5.0, respectively. Bst-Ini/Bst-Opt: boosted by the parser trained on the initial/optimized projected treebank.

mance on the standard CTB test set. Table 2 shows the performance of the parser trained on the optimized projected treebank. Unexpectedly, we find that the constituent F_1-value of the parser trained on the optimized treebank drops sharply from the baseline, although the span F_1-value remains nearly the same. We assume that the EM procedure gives the original projected treebank more consistency between each single tree while the revised treebank deviates from the CTB annotation standard, but it needs to be validated by the following experiments.

4.2 Boost an Traditional Parser

The projected parser is used to help the reranking of the k-best parses produced by another state-of-the-art parser, which is called the baseline parser for convenience. In our experiments we choose the revised Chinese parser (Xiong et al., 2005) based on Collins model 2 (Collins, 1999) as the baseline parser.[2]

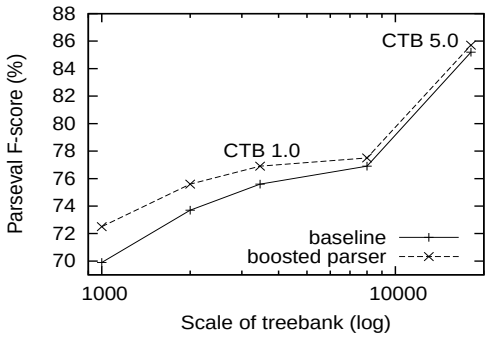


Figure 2: Boosting performance of the projected parser on a series of baseline parsers that are trained on treebanks of different scales.

The baseline parser is respectively trained on CTB 1.0 and CTB 5.0. For both corpora we follow the traditional corpus splitting: chapters 271-300 for testing, chapters 301-325 for development, and else for training. Experimental results are shown in Table 3. We find that both projected parsers bring significant improvement to the baseline parsers. Especially the later, although performs worse on CTB standard test set, gives a larger improvement than the former. This to some degree confirms the previous assumption. However, more investigation must be conducted in the future.

We also observe that for the baseline parser trained on the much larger CTB 5.0, the boosting performance of the projected parser is relatively lower. To further investigate the regularity that the boosting performance changes according to the scale of training treebank of the baseline parser, we train a series of baseline parsers with different amounts of trees, then use the projected parser trained on the optimized treebank to enhance these baseline parsers. Figure 2 shows the experimental results. From the curves we can see that the smaller the training corpus of the baseline parser, the more significant improvement can be obtained. This is a good news for the resource-scarce languages that have no large treebanks.

[2]The Berkeley Parser fails to give k-best parses for some sentences when trained on small treebanks, and these sentences have to be deleted in the k-best reranking experiments.

4.3 Using in Machine Translation

We investigate the effect of the projected parser in the tree-based translation model on Chinese-to-English translation. A series of contrast translation systems are built, each of which uses a supervised Chinese parser (Xiong et al., 2005) trained on a particular amount of CTB trees.

We use the FBIS Chinese-English bitext as the training corpus, the 2002 NIST MT Evaluation test set as our development set, and the 2005 NIST MT Evaluation test set as our test set. We first extract the tree-to-string translation rules from the training corpus by the algorithm of (Liu et al., 2006), and train a 4-gram language model on the Xinhua portion of GIGAWORD corpus with Kneser-Ney smoothing using the SRI Language Modeling Toolkit (Stolcke and Andreas, 2002). Then we use the standard minimum error-rate training (Och, 2003) to tune the feature weights to maximize the system's BLEU score.

Figure 3 shows the experimental results. We find that the translation system using the projected parser achieves the performance comparable with the one using the supervised parser trained on CTB 1.0. Considering that the F-score of the projected parser is only 22.8%, which is far below of the 75.6% F-score of the supervised parser trained on CTB 1.0, we can give more confidence to the assumption that the projected parser is apt to describe the syntax structure of the counterpart language. This surprising result also gives us an inspiration that better translation would be achieved by combining projected parsing and supervised parsing into hybrid parsing schema.

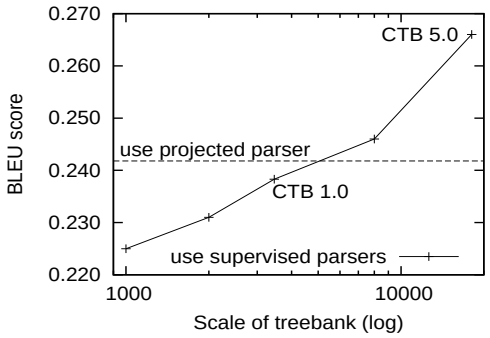


Figure 3: Performances of the translation systems, which use the projected parser and a series of supervised parsers trained CTB trees.

5 Conclusion

This paper describes an effective strategy for constituent projection, where dependency projection and constituent projection are consequently conducted to obtain the initial projected treebank, and an constraint EM procedure is then performed to optimized the projected trees. The projected parser, trained on the projected treebank, significantly boosts an existed state-of-the-art supervised-trained parser, especially trained on a smaller treebank. When using the projected parser in tree-based translation, we achieve the translation performance comparable with using a supervised parser trained on thousands of human-annotated trees.

As far as we know, this is the first time that the experimental results are systematically reported about the constituent projection and its applications. However, many future works need to do. For example, more energy needs to be devoted to the treebank optimization, and hybrid parsing schema that integrates the strengths of both supervised-trained parser and projected parser would be valuable to be investigated for better translation.

Acknowledgments

The authors were supported by 863 State Key Project No. 2006AA010108, National Natural Science Foundation of China Contract 60873167, Microsoft Research Asia Natural Language Processing Theme Program grant (2009-2010), and National Natural Science Foundation of China Contract 90920004. We are grateful to the anonymous reviewers for their thorough reviewing and valuable suggestions.

References

Bod, Rens. 2006. An all-subtrees approach to unsupervised parsing. In *Proceedings of the COLING-ACL*.

Carreras, Xavier, Michael Collins, and Terry Koo. 2008. Tag, dynamic programming, and the perceptron for efficient, feature-rich parsing. In *Proceedings of the CoNLL*.

Charniak, Eugene and Mark Johnson. 2005. Coarse-to-fine-grained n-best parsing and discriminative reranking. In *Proceedings of the ACL*.

Charniak, Eugene. 2000. A maximum-entropy-inspired parser. In *Proceedings of the NAACL*.

Collins, Michael. 1999. Head-driven statistical models for natural language parsing. In *Ph.D. Thesis*.

Collins, Michael. 2000. Discriminative reranking for natural language parsing. In *Proceedings of the ICML*, pages 175–182.

Collins, Michael. 2002. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *Proceedings of the EMNLP*, pages 1–8, Philadelphia, USA.

Ganchev, Kuzman, Jennifer Gillenwater, and Ben Taskar. 2009. Dependency grammar induction via bitext projection constraints. In *Proceedings of the 47th ACL*.

Huang, Liang and David Chiang. 2005. Better k-best parsing. In *Proceedings of the IWPT*, pages 53–64.

Huang, Liang, Kevin Knight, and Aravind Joshi. 2006. Statistical syntax-directed translation with extended domain of locality. In *Proceedings of the AMTA*.

Huang, Liang. 2008. Forest reranking: Discriminative parsing with non-local features. In *Proceedings of the ACL*.

Hwa, Rebecca, Philip Resnik, Amy Weinberg, and Okan Kolak. 2002. Evaluating translational correspondence using annotation projection. In *Proceedings of the ACL*.

Hwa, Rebecca, Philip Resnik, Amy Weinberg, Clara Cabezas, and Okan Kolak. 2005. Bootstrapping parsers via syntactic projection across parallel texts. In *Natural Language Engineering*, volume 11, pages 311–325.

Klein, Dan and Christopher D. Manning. 2002. A generative constituent-context model for improved grammar induction. In *Proceedings of the ACL*.

Klein, Dan and Christopher D. Manning. 2004. Corpusbased induction of syntactic structure: Models of dependency and constituency. In *Proceedings of the ACL*.

Liu, Yang, Qun Liu, and Shouxun Lin. 2006. Tree-to-string alignment template for statistical machine translation. In *Proceedings of the ACL*.

Liu, Yang, Tian Xia, Xinyan Xiao, and Qun Liu. 2009. Weighted alignment matrices for statistical machine translation. In *Proceedings of the EMNLP*.

McClosky, David, Eugene Charniak, and Mark Johnson. 2006. Reranking and self-training for parser adaptation. In *Proceedings of the ACL*.

McDonald, Ryan, Fernando Pereira, Kiril Ribarov, and Jan Hajič. 2005. Non-projective dependency parsing using spanning tree algorithms. In *Proceedings of HLT-EMNLP*.

Och, Franz J. and Hermann Ney. 2000. Improved statistical alignment models. In *Proceedings of the ACL*.

Och, Franz Joseph. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41th Annual Meeting of the Association for Computational Linguistics*, pages 160–167.

Papineni, Kishore, Salim Roukos, Todd Ward, and Weijing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the ACL*.

Petrov, Slav, Leon Barrett, Romain Thibaux, and Dan Klein. 2006. Learning accurate, compact, and interpretable tree annotation. In *Proceedings of the ACL*.

Sarkar, Anoop. 2001. Applying co-training methods to statistical parsing. In *Proceedings of NAACL*.

Seginer, Yoav. 2007. Fast unsupervised incremental parsing. In *Proceedings of the ACL*.

Smith, David and Jason Eisner. 2009. Parser adaptation and projection with quasi-synchronous grammar features. In *Proceedings of EMNLP*.

Steedman, Mark, Miles Osborne, Anoop Sarkar, Stephen Clark, Rebecca Hwa, Julia Hockenmaier, Paul Ruhlen, Steven Baker, and Jeremiah Crim. 2003. Bootstrapping statistical parsers from small datasets. In *Proceedings of the EACL*.

Stolcke and Andreas. 2002. Srilm - an extensible language modeling toolkit. In *Proceedings of the International Conference on Spoken Language Processing*, pages 311–318.

Xiong, Deyi, Shuanglong Li, Qun Liu, and Shouxun Lin. 2005. Parsing the penn chinese treebank with semantic knowledge. In *Proceedings of IJCNLP 2005*, pages 70–81.

Yamada, H and Y Matsumoto. 2003. Statistical dependency analysis using support vector machines. In *Proceedings of IWPT*.

Zhang, Yue and Stephen Clark. 2008. A tale of two parsers: investigating and combining graph-based and transition-based dependency parsing using beam-search. In *Proceedings of EMNLP*.

A Comparative Study on Ranking and Selection Strategies for Multi-Document Summarization

Feng Jin, Minlie Huang, Xiaoyan Zhu
State Key Laboratory of Intelligent Technology and Systems
Tsinghua National Laboratory for Information Science and Technology
Dept. of Computer Science and Technology, Tsinghua University
jinfengfeng@gmail.com,{aihuang,zxy-dcs}@tsinghua.edu.cn

Abstract

This paper presents a comparative study on two key problems existing in extractive summarization: the ranking problem and the selection problem. To this end, we presented a systematic study of comparing different learning-to-rank algorithms and comparing different selection strategies. This is the first work of providing systematic analysis on these problems. Experimental results on two benchmark datasets demonstrate three findings: (1) pairwise and listwise learning-to-rank algorithms outperform the baselines significantly; (2) there is no significant difference among the learning-to-rank algorithms; and (3) the integer linear programming selection strategy generally outperformed Maximum Marginal Relevance and Diversity Penalty strategies.

1 Introduction

As the rapid development of the Internet, document summarization has become an important task since document collections are growing larger and larger. Document summarization, which aims at producing a condensed version of the original document(s), helps users to acquire information that is both important and relevant to their information need. So far, researchers have mainly focused on extractive methods which choose a set of salient textual units to form a summary. Such textual units are typically sentences, sub-sentences (Gillick and Favre, 2009), or excerpts (Sauper and Barzilay, 2009).

Almost all extractive summarization methods face two key problems: the first problem is how to rank textual units, and the second one is how to select a subset of those ranked units. The ranking problem requires systems model the relevance of a textual unit to a topic or a query. In this paper, the ranking problem refers to either sentence ranking or concept ranking. Concepts can be unigrams, bigrams, semantic content units, etc., although in our experiment, only bigrams are used as concepts. The selection problem requires systems improve diversity or remove redundancy so that more relevant information can be covered by the summary as its length is limited. As our paper focuses on extractive summarization, the selection problem refers to selecting sentences. However, the selection framework presented here is universal for selecting arbitrary textual units, as discussed in Section 4.

There have been a variety of studies to approach the ranking problem. These include both unsupervised sentence ranking (Luhn, 1958; Radev and Jing, 2004, Erkan and Radev, 2004), and supervised methods (Ouyang et al., 2007; Shen et al., 2007; Li et al., 2009). Even given a list of ranked sentences, it is not trivial to select a subset of sentences to form a good summary which includes diverse information within a length limit. Three common selection strategies have been studied to address this problem: Maximum Marginal Relevance (MMR) (Carbonell and Goldstein, 1998), Diversity Penalty (DiP) (Wan, 2007), and integer linear programming (ILP) (McDonald, 2007; Gillick and Favre, 2009). As different methods were often evaluated on different datasets, it is of great value to systematically compare ranking and selection strategies on the same dataset. However, to the best of our knowledge, there is still no work to compare different ranking strategies or compare different selection strategies.

In this paper, we presented a comparative study on the ranking problem and the selection

Coling 2008: Poster Volume, pages 525–533,
Beijing, August 2010

problem for extractive summarization. We compared three genres of learning-to-rank methods for ranking sentences or concepts: SVR, a pointwise ranking algorithm; RankNet, a pairwise learning-to-rank algorithm; and ListNet, a listwise learning-to-rank algorithm. We adopted an ILP framework that is able to select sentences based on sentence ranking or concept ranking. We compared it with other selection strategies such as MMR and Diversity Penalty. We conducted our comparative experiments on the TAC 2008 and TAC 2009 datasets, respectively. Our contributions are two-fold: First, to the best of our knowledge, this is the first work of presenting systematic and in-depth analysis on comparing ranking strategies and comparing selection strategies. Second, this is the first work using pairwise and liswise learning-to-rank algorithms to perform concept (word bigram) ranking for extractive summarization.

The rest of this paper is organized as follows. We introduce the related work in Section 2. In Section 3, we present three ranking algorithms, SVR, RankNet, and ListNet. We describe the sentence selection problem with an ILP framework described in Section 4. We introduce features in Section 5. Evaluation and experiments are presented in Section 6. Finally, we conclude this paper in Section 7.

2 Related Work

A number of extractive summarization studies used unsupervised methods with surface features, linguistic features, and statistical features to guide sentence ranking (Edmundson, 1969; McKeown and Radev, 1995; Radev et al., 2004; Nekova et al., 2006). Recently, graph-based ranking methods have been proposed for sentence ranking and scoring, such as LexRank (Erkan and Radev, 2004) and TextRank (Mihalcea and Tarau, 2004).

There are also a variety of studies on supervised learning methods for sentence ranking and selection. Kupiec et al. (1995) developed a naive Bayes classifier to decide whether a sentence is worthy to extract. Recently, Conditional Random Field (CRF) and Structural SVM have been employed for single document summarization (Shen et al., 2007; Li et al., 2009).

Besides ranking sentences directly, there are some approaches that select sentences based on concept ranking. Radev et al. (2004) used centroid words whose *tf***idf* scores are above a threshold. Filatova and Hatzivassiloglou (2004) used atomic event as concept. Moreover, summarization evaluation metrics such as Basic Element (Hovy et al., 2006), ROUGE (Lin and Hovy, 2003) and Pyramid (Passonneau et al., 2005) are all counting the concept overlap between generated summaries and human-written summaries.

Another important issue existing in extractive summarization is to find an optimal sentence subset which can cover diverse information. Maximal Marginal Relevance (MMR) (Carbonell and Goldstein, 1998) and Diversity Penalty (Wan, 2007) are most widely used approaches to reduce redundancy. The two methods are essentially based on greedy search. By contrast, ILP based approaches view summary generation as a global optimization problem. McDonald (2007) proposed a sentence-level ILP solution. Sauper and Barzilay (2009) presented an excerpt-level ILP method to generate Wikipedia articles. Gillick and Favre (2009) proposed a concept-level ILP, but they used document frequency to score concepts (bigrams), without any learning process. Some recent studies (Gillick and Favre, 2009; Martins and Smith, 2009) also modeled sentence selection and compression jointly using ILP. Our ILP framework proposed here is based on these studies. Although various selection strategies have been proposed, there is no work to systematically compare these strategies yet.

Learning to rank attracts much attention in the information retrieval community recently. Pointwise, pairwise and listwise learning-to-rank approaches have been extensively studied (Liu, 2009). Some of those have been applied to document summarization, such as SVR (Ouyang et al., 2007), classification SVM (Wang et al., 2007), and RankNet (Svore et al., 2007). Again, there is no work to systematically compare these ranking algorithms. To the best of our knowledge, this is the first time that a listwise learning-to-rank algorithm, ListNet (Cao et al., 2007), is adapted to document summarization in this paper. Moreover, pairwise and listwise learning-to-rank algorithms have never been used to perform concept ranking for extractive summarization.

3 Ranking Sentences or Concepts

Given a query and a collection of relevant documents, an extractive summarization system is required to generate a summary consisting of a set of text units (usually sentences). The first problem we need to consider is to determine the importance of these sentences according to the input query. We approach this ranking problem in two ways: the first way is to score sentences directly using learning-to-rank algorithms, and thus the goal of summarization is to select a subset of sentences, considering both relevance and redundancy. The second way is to score concepts within the document collection, and then the summarization task is to select a sentence subset that can cover those important concepts maximally. The problem of sentence selection will be described in Section 4.

Suppose the relevant document collection for a query q is D_q. From this collection, we obtain a set of sentences or concepts (e.g., word bigrams), $S=\{s_1,s_2,...,s_n\}$ or $C=\{c_1,c_2,..., c_n\}$. Before training, each s_i or c_i is associated with a gold standard score, y_i. A feature vector, $x_j=\Phi(s_j/c_j,q,D_q)$, is constructed for each sentence or concept. The learning algorithm will learn a ranking function $f(x_j)$ from a collection of query-document pairs $\{(q_i,D_{qi})|i= 1, 2,...,m\}$.

We investigated three learning-to-rank methods to learn $f(x_j)$. The first one is a pointwise ranking algorithm, support vector regression (SVR). This algorithm treats sentences (or concepts) independently. The second method is a pairwise ranking algorithm, RankNet, which learns a ranking function from a list of sentence (or concept) pairs. Each pair is labeled as 1 if the first sentence s_i (or concept c_i) ranks ahead of the second s_j (or c_j), and 0 otherwise.

The listwise ranking algorithm, ListNet, learns the ranking function $f(x_j)$ in a different way. A list of sentences (or concepts) is treated as a whole. Both RankNet and ListNet take into account the dependency between sentences (or concepts).

3.1 Support Vector Regression

Support Vector Regression (SVR), a generalization of the classical SVM formulation, attempts to learn a regression model. SVR has been applied to summarization in (Ouyang et al., 2007; Metzler and Kanungo, 2008). In our work, we train the SVR model to fit the gold standard score of each sentence or concept.

Formally, the objective of SVR is to minimize the following objective:

$$\Im(w,b,\xi) = \left\{\frac{1}{2}\| w\|^2 + C\left(v\cdot\xi + \frac{1}{N}\sum_{x_i} L(y_i - f(x_i))\right)\right\} \quad (1)$$

where $L(x)=|x|-\xi$ if $x > \xi$ and otherwise $L(x)=0$; y_i is the gold standard score of x_i; $f(x) =w^Tx+b$, the predicted score of x; C and v are two parameters; and N is the total number of training examples.

3.2 RankNet

RankNet is a pairwise learning-to-rank method (Burges et al., 2005). In this algorithm, training examples are handled pair by pair. Given a pair of feature vectors (x_i, x_j), the gold standard probability $\overline{P}_{ij}$ is set to be 1 if the label of the pair is 1, which means x_i ranks ahead of x_j. The gold standard probability is 0 if the label of the pair is 0. Then the predicted probability P_{ij}, which defines the probability of x_i ranking ahead of x_j by the model, is represented as a logistic function:

$$P_{ij} = \frac{\exp(f(x_i) - f(x_j))}{1+\exp(f(x_i) - f(x_j))} \quad (2)$$

where $f(x)$ is the ranking function. The objective of the algorithm is to minimize the cross entropy between the gold standard probability and the predicted probability, which is defined as follows:

$$C_{ij}(f) = -\overline{P}_{ij}\log P_{ij} - (1-\overline{P}_{ij})\log(1-P_{ij}) \quad (3)$$

A three-layer neural network is used as the ranking function, as follows:

$$f(x_n) = g^3\left(\sum_j w_{ij}^{32} g^2\left(\sum_k w_{jk}^{21} x_{nk} + b_j^2\right) + b_i^3\right) \quad (4)$$

where for weights w and bias b, the superscripts indicate the node layer while the subscripts indicate the node indexes within each layer. And x_{nk} is the k-th component of input feature vector x_n. Then a gradient descent method is used to learn the parameters. For details, refer to (Burges et al., 2005).

3.3 ListNet

ListNet takes a list of items as input in the learning process. More specifically, suppose we have

a list of feature vectors ($x_1, x_2, ..., x_n$) and each feature vector x_i has an gold standard score y_i, which has been assigned before training. Accordingly, we have a list of gold standard scores ($y_1, y_2, ..., y_n$). We also have a list of scores assigned by the algorithm during training, say, ($f(x_1), f(x_2), ..., f(x_n)$). Given a score list $S=\{s_1, s_2, ..., s_n\}$, the probability that x_j will rank the first place among the n items is defined as follows:

$$P_s(j) = \frac{\Phi(s_j)}{\sum_{k=1}^{n}\Phi(s_k)} = \frac{\exp(s_j)}{\sum_{k=1}^{n}\exp(s_k)} \quad (5)$$

It is easy to prove that ($P_s(1), P_s(2), ..., P_s(n)$) is a probability distribution, as the sum of them equals to 1. Therefore, the cross entropy can be used to define the loss between the gold standard distribution $P_y(j)$ and the distribution $P_f(j)$, as follows:

$$L(y, f) = -\sum_{j=1}^{n} P_y(j)\log P_f(j) \quad (6)$$

where y represents the gold standard score list ($y_1, y_2, ..., y_n$) and $f=(f(x_1), f(x_2), ..., f(x_n))$ is the score list output by the ranking algorithm.

The function f is defined as a linear function, as follows:

$$f_w(x_i) = w^T x_i \quad (7)$$

Then the gradient of loss function $L(y,f)$ with respect to the parameter vector w can be calculated as follows:

$$\Delta w = \frac{\partial L(y, f_w)}{\partial w} = -\sum_{j=1}^{n} P_y(x_j)\frac{\partial f_w(x_j)}{\partial w} + \frac{1}{\sum_{j=1}^{n}\exp(f_w(x_j))}\sum_{j=1}^{n}\exp(f_w(x_j))\frac{\partial f_w(x_j)}{\partial w} \quad (8)$$

During training, w is updated in a gradient descent manner: $w=w-\eta\Delta w$ and η is the learning rate. For details, refer to (Cao et al., 2007).

4 ILP-based Selection Framework

After we have a way of ranking sentences or concepts, we face a sentence selection problem: selecting an optimal subset of sentences as the final summary. To integrate sentence/concept ranking, we adopted an integer linear programming (ILP) framework to find the optimal sentence subset (Filatova and Hatzivassiloglou, 2004; McDonald, 2007; Gillick and Favre, 2009; Takamura and Okumura, 2009). ILP is a global optimization problem whose objective and constraints are linear in a set of integer variables.

Formally, we define the problem of sentence selection as follows:

$$\text{maximize: } \left\{\sum_i f(x_i) * z_i^x\right\} \quad (9)$$

$$s.t. \quad \sum_j z_j^u * |u_j| \le Lim$$

$$\sum_j z_j^u * I(i,j) \ge z_i^x, \quad \forall i$$

$$(z_i^x + z_j^x) * sim(x_i, x_j) < \delta \quad \forall i, j$$

$$z_i^x, z_j^u \in \{0,1\}, \quad \forall i, j$$

where:

$\boldsymbol{x_i}$ – the representation unit, such as a sentence or a concept. We term it representation unit because the summary quality is represented by the set of included x_i;

$\boldsymbol{f(x_i)}$ - the ranking function given by the learning-to-rank algorithms;

$\boldsymbol{u_j}$ - the selection unit, for instance, a sentence in this paper. $|u_j|$ is the number of words in u_j;

$\boldsymbol{z_i^x}$ - the indicator variable which denotes the presence or absence of x_i in the summary;

$\boldsymbol{z_j^u}$ - the indicator variable which denotes inclusion or exclusion of u_j;

$\boldsymbol{I(i, j)}$ - a binary constant indicating that whether x_i appears in u_j. It is either 1 or 0;

Lim - the length limit;

$\boldsymbol{sim(x_i, x_j)}$ - a similarity measure for considering the redundancy;

$\boldsymbol{\delta}$ - the redundancy threshold.

The first constraint indicates the length limit. The second constraint asserts that if a representation unit x_i is included in a summary, at least one selection unit that contains x_i must be selected. The third constraint considers redundancy. If the representation unit is sentence, the similarity measure is defined as *tf*idf* similarity, and $\delta/2$ is the similarity threshold, which was set to be 1 here. For concepts, the similarity measure can be defined as

$$sim(x_i, x_j) = \begin{cases} 1, & x_i = x_j \\ 0, & \text{otherwise} \end{cases}$$

However, other definition is also feasible, depending on what has been selected as representation unit.

Note that this framework is very general. If the representation unit x_i is a sentence, the ranking function is defined on sentence. Thus the ILP framework will find a set of sentences that can optimize the total scores of selected sentences, subject to several constraints. If the representation unit is a concept, the ranking function measures the importance of a concept to be included in a summary. Thus the goal of ILP is to find a set of sentences by maximizing the scores of concepts covered by those selected sentences.

D_q	relevant document collection in response to query q
d	one single document
w_i	unigram
w_iw_{i+1}	bigram
S	sentence
$tf_d(w_i)$	the frequency of w_i occurring in d
$df_D(w_i)$	the number of documents containing w_i in collection D

Table 1. Notations for features.

5 Features

To facilitate the following description, some notations are defined in Table 1. In our dataset, each query has a title and narrative to precisely define an information need. The following is a query example from the TAC 2008 test dataset:

```
<topic id = "D0801A">
  <title> Airbus A380 </title>
  <narrative>
      Describe developments in the production and
      launch of the Airbus A380.
  </narrative>
</topic>
```

Features for sentence ranking and concept ranking are listed in the following. We use word bigrams as concept here.

Sentence Features

(1) Cluster frequency: $\sum_{w_i \in S} tf_{D_q}(w_i)$

(2) Title frequency: $\sum_{w_i \in S} tf_d(w_i)$ where d is a new document that consists of all the titles of documents in D_q.

(3) Query frequency: $\sum_{w_i \in S} tf_d(w_i)$ where d is a document consisting of the title and narrative fields of the current topic.

(4) Theme frequency: $\sum_{w_i \in S \wedge w_i \in T} tf_{D_q}(w_i)$ where T is the top 10% frequent unigram words in D_q.

(5) Document frequency of bigrams in the sentence: $\sum_{w_iw_{i+1} \in S} df_D(w_iw_{i+1})$.

(6) PageRank score: as described in (Mihalcea and Tarau, 2004), each sentence in D_q is a node in the graph and the cosine similarity between a pair of sentences is used as edge weight.

Concept Features

(1) Cluster frequency: $tf_{D_q}(w_iw_{i+1})$, the frequency of w_iw_{i+1} occurring in D_q.

(2) Title frequency: $tf_d(w_iw_{i+1})$, where d is a document consisting of all the titles of documents in D_q.

(3) Query Frequency: the frequency of the bigram occurring in the topic title and narrative.

(4) Average term frequency: $\sum_{d \in D_q} tf_d(w_iw_{i+1}) / |D_q|$. $|D_q|$ is the number of documents in the set.

(5) Document frequency: the document frequency of this bigram.

(6) Minimal position: the minimal position of this bigram relative to the document length.

(7) Average position: the average position of this bigram in collection D_q.

6 Experimental Results

6.1 Data Preprocessing

We conducted experiments on the TAC 2008 and TAC 2009 datasets. The task requires producing a 100-word summary for each query (also termed topic sometimes). There are 48 queries in TAC 2008 and 44 queries in TAC 2009. A query example has been given in Section 5. Relevant documents for these queries have been specified. And four human-written summaries were supplied as reference summaries for each query.

We segmented the relevant documents into sentences using the LingPipe toolkit [1] and stemmed words using the Porter Stemmer. Word bigrams are used as concepts in this paper. If the two words in a bigram are both stopwords, the bigram will be discarded. The sen-

[1] http://alias-i.com/lingpipe/index.html

tence features and bigram features are then calculated. As our focus is on comparing different ranking strategies and selection strategies, we did not apply any sophisticated linguistic or semantic processing techniques (as pre- or post-processing). Thus we did not compare our results to those submitted to the TAC conferences.

We train the learning algorithms on one dataset and then evaluate the algorithms on the other. The generated summaries are evaluated using the ROUGE toolkit (Lin and Hovy, 2003).

6.2 Preparing Training Samples

As our work includes both sentence ranking and concept ranking, we need to establish two types of training data. Fortunately, we are able to do this based on the reference summaries and annotation results provided by the TAC conferences.

For the sentence ranking problem, we compute the average ROUGE-1 score for each sentence by comparing it to the four reference summaries for each query. This score is treated as the gold-standard score. In ListNet, these scores are directly used (see formula (5)). While in RankNet, the sentences for a query are grouped into 10 bins according to their ROUGE-1 scores, and then we extract sentences from different bins respectively to form a pair. We assume that a sentence in a higher scored bin should rank ahead of those sentences in lower scored bins.

As for the concept ranking problem, gold-standard scores are obtained from the human annotated Pyramid data. The weight of each semantic content unit (SCU) is the number of reference summaries in which the SCU appears. So straightforwardly, the gold-standard score of a bigram is the largest weight of all SCUs that contain the bigram. And if a bigram does not occur in any SCU, its score will be 0. Thus the bigram scores belong to the set {0,1,2,3,4} as there are four human-written summaries for each query. These scores are directly used in ListNet (see formula (5)). And in RankNet, bigram pairs are constructed according to the gold-standard scores.

6.3 Learning Parameters

For SVR, the radial basis kernel function is employed and the optimal values for parameters *C*, *v* and *g* (for the kernel) are found using the *gridregression.py* tool provided by LibSVM (Chang and Lin, 2001) with a 5-fold cross validation on the training set.

RankNet applies a three-layer (one hidden layer) neural network with only one node in the output layer, as described in (Burges et al., 2005). The number of hidden neurons was empirically set to be 10. The learning rate was set to 0.001 for sentence ranking and 0.01 for bigram ranking.

As for ListNet, the learning rate for sentence ranking and concept ranking are both set to be 0.1 empirically.

6.4 Comparing Ranking Strategies

In this section, we compared different ranking strategies for both sentence ranking and concept ranking. The sentence selection strategies were fixed to the ILP selection framework as shown in Section 4. We chose ILP as the selection strategy because we want to compare our system with the following two methods (as baselines):

(1) ***SENT_ILP***: A sentence-level method proposed by McDonald (2007) with ILP formulation. We implemented the query-focused version of the formulae as TAC 2008 and 2009 required query-focused summarization.

(2) ***DF_ILP***: A concept-level ILP method using document frequency to score word bigrams (Gillick and Favre, 2009), without any learning process.

The differences between our framework and SENT_ILP are: a) SENT_ILP used a redundancy factor in the objective function whereas we modeled redundancy as constraints; b) SENT_ILP used *tf***idf* similarity to compute relevance scores whereas we used learning algorithms.

The ROUGE-1 and ROUGE-2 measures for each method are presented in Table 2 and Table 3. Note that the performance on the TAC 2008 dataset was obtained from the models that were trained on the TAC 2009 dataset. Then, the datasets were interchanged for training and testing, respectively. Different learning-to-rank strategies (SVR, RankNet, ListNet) do not show significant differences between one and another, but they all outperform *SENT_ILP* substantially (p-value < 0.0001). And for concept ranking, RankNet and ListNet both achieve significantly better ROUGE-2 results (p-value < 0.005) than

DF_ILP. This infers that considering more features will have better results than using document frequency to score concepts. The Wilcoxon signed-rank test (Wilcoxon, 1945) is used for significance tests in our experiment. A good ranking strategy for modeling relevance is important for extractive summarization. RankNet which used a three-layer network (non-linear function) as the ranking function performs slightly better than ListNet which is based on a linear ranking function.

Dataset	Method	ROUGE-1	ROUGE-2
TAC 2008	SVR	0.35086	0.08447
	RankNet	**0.36025**	**0.09291**
	ListNet	0.35365	0.09129
	SENT_ILP	0.31546	0.06500
TAC 2009	SVR	0.36125	0.09659
	RankNet	**0.36216**	**0.09778**
	ListNet	0.35480	0.09126
	SENT_ILP	0.31962	0.07034

Table 2. Results of sentence ranking strategies.

Dataset	Method	ROUGE-1	ROUGE-2
TAC 2008	SVR	0.36555	0.10291
	RankNet	**0.37564**	**0.11213**
	ListNet	0.36863	0.10660
	DF_ILP	0.36922	0.10373
TAC 2009	SVR	0.37126	0.10698
	RankNet	**0.37513**	**0.11364**
	ListNet	0.37499	0.11313
	DF_ILP	0.36347	0.10156

Table 3. Results of concept ranking strategies.

It is worth noting that Pyramid annotations may not cover all important bigrams, partly because SCUs in reference summaries have been rephrased by human annotators. Note that we simply extract original sentences to form a summary, thus it is possible that a bigram which is important in the original sentences does not appear in any rephrased SCUs at all. Such bigrams will have a gold-standard score of 0, which is erroneous supervision. For example, the bigrams *hurricane katrina* in topic D0804A about *Katrina pet rescue* and *life support* in D0806A about *Terri Schiavo case* are not annotated in any SCUs, but these bigrams are both key terms for the topics.

6.5 Comparing Selection Strategies

In order to study the influence of different selection strategies, we compare the ILP selection strategy (as introduced in Section 4) with other popular selection strategies, based on the same sentence ranking algorithm (we chose sentence-level RankNet). The baselines to be compared are as follows:

(1) ***MMR***: As shown in (Carbonell and Goldstein, 1998), the formula of MMR is:

$$MMR = \arg\max_{s_i \in R-S} \left\{ \lambda D_1(q, s_i) - (1-\lambda) \max_{s_j \in S} D_2(s_i, s_j) \right\}$$

where q is the given query; R is the set of all sentences; S is the set of already included sentences; D_1 is the normalized ranking score $f(x_i)$ of s_i, and D_2 is the cosine similarity of the feature vectors for s_i and s_j. Our implementation was similar to the MMR strategy in the MEAD[2]summarizer.

(2) ***DiP***: Diversity penalty which penalizes the score of candidate sentences according to the already selected ones (Wan, 2007).

Dataset	Method	ROUGE-1	ROUGE-2
TAC 2008	ILP	**0.36025**	**0.09291**
	MMR	0.35459	0.09086
	DiP	0.35263	0.08689
TAC 2009	ILP	**0.36216**	**0.09778**
	MMR	0.35148	0.08881
	DiP	0.34714	0.08672

Table 4. Comparing selection strategies.

The corresponding ROUGE scores are presented in Table 4. ILP outperforms other selection strategies significantly on the TAC 2009 dataset (both ILP vs. MMR and ILP vs. DiP). Although improvements are observed with ILP on the TAC 2008 dataset, the difference is not significant (using ILP vs. using MMR). MMR is comparable to DiP as they are both based on greedy search in nature.

To investigate the difference between these strategies, we present in-depth analysis here. First, the average length of summaries generated by ILP is 97.1, while that by MMR and DiP are 95.5 and 92.7, respectively. Note that the required summary length is 100 and that more words can potentially cover more information. Thus, ILP can generate summaries with more information. This is because ILP is a global optimization algorithm, subject to the length constraint. Second, the average rank of sentences selected by ILP is 12.6, while that by MMR and

[2] http://www.summarization.com/mead/

DiP is about 5, which is substantially different. ILP can search down the ranked list while the other two methods tend to only select the very top sentences. Third, there are 4.1 sentences on average in each ILP-generated summary, while the number for MMR and DiP generated summaries are 2.7 and 2.5, respectively. Thus ILP tend to select shorter sentences than MMR and DiP. This may help reduce redundancy as longer sentences may contain more topic irrelevant clauses or phrases.

6.6 Discussions

Interestingly, although the learning-to-rank algorithms combined with the ILP selection strategy perform well in summarization, the performance is still far from that of manual summarization. In this study, we investigate the upper bound performance. We used the presented ILP framework to generate summaries based on the gold-standard scores, rather than the scores given by the learning algorithms. In other words, $f(x_i)$ in formula (9) is replaced by the gold-standard scores. The ROUGE results are shown in Table 5. We also listed the best/worst/average ROUGE scores of human summaries in TAC by comparing one human summary (as generated summary) to the other three human summaries (as reference summaries). These results are substantially better than those by the learning algorithms. Sentence- and concept- level ranking produces very close results to best human summaries. Some ROUGE-2 scores are even higher than those of human summaries. This is reasonable as human annotators may have difficulty in organizing content when there are many documents and sentences. The results reflect that there is a remarkable gap between the gold-standard scores and the learned scores.

Dataset	Method	ROUGE-1	ROUGE-2
TAC 2008	Sentence-level	0.44216	0.14842
	Concept-level	0.42222	**0.16018**
	Human Best	**0.44220**	0.13079
	Human Average	0.41417	0.11606
	Human Worst	0.38005	0.10736
TAC 2009	Sentence-level	0.45500	0.15565
	Concept-level	0.43526	**0.17118**
	Human Best	**0.45663**	0.14864
	Human Average	0.44443	0.12680
	Human Worst	0.39652	0.11109

Table 5. Upper bound performance.

7 Conclusion and Future Work

We presented systematic and extensive analysis on studying two key problems in extractive summarization: the ranking problem and the selection problem. We compared three genres of learning-to-rank algorithms for the ranking problem, and investigated ILP, MMR, and Diversity Penalty strategies for the selection problem. To the best of our knowledge, this is the first work of presenting systematic comparison and analysis on studying these problems. We also at the first time proposed to use learning-to-rank algorithms to perform concept ranking for extractive summarization.

Our future work will focus on: (1) exploiting more features that can reflect summary quality; (2) optimizing summarization evaluation metrics directly with new learning algorithms.

Acknowledgments

This work was partly supported by the Chinese Natural Science Foundation under grant No. 60973104 and No. 60803075, and with the aid of a grant from the International Development Research Center, Ottawa, Canada IRCI project from the International Development.

References

Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton and Greg Hullender. 2005. Learning to Rank Using Gradient Descent. In *Proceedings of the 22nd International Conference on Machine Learning*.

Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai and Hang Li. 2007. Learning to Rank: from Pairwise Approach to Listwise Approach. In *Proceedings of ICML 2007*.

Jaime Carbonell and Jade Goldstein. 1998. The Use of MMR, Diversity-Based Reranking for Reordering Documents and Producing Summaries. In *Proceedings of SIGIR*, August 1998, pp. 335 - 336.

Chih-Chung Chang and Chih-Jen Lin. 2001. LIBSVM: a Library for Support Vector Machines. Software available at *http://www.csie.ntu.edu.tw/~cjlin/libsvm*

H. P. Edmundson. 1969. New Methods in Automatic Extracting. *Journal of the ACM (JACM) Archive*, Volume 16, Issue 2 (April 1969) Pages: 264 - 285.

G. Erkan and Dragomir R. Radev. 2004. LexPageRank: Prestige in Multi-Document Text Summa-

rization. In *Proceedings of EMNLP 2004*, Barcelona, Spain.

Elena Filatova and Vasileios Hatzivassiloglou. 2004. Event-based Extractive Summarization. In *Proceedings of ACL Workshop on Summarization*, volume 111.

Dan Gillick and Benoit Favre. 2009. A Scalable Global Model for Summarization. In *Proceedings of the Workshop on Integer Linear Programming for Natural Language Processing*.

Eduard Hovy, Chin-yew Lin, Liang Zhou and Junichi Fukumoto. 2006. Automated Summarization Evaluation with Basic Elements. In *Proceedings of the Fifth Conference on Language Resources and Evaluation*.

Julian Kupiec, Jan Pedersen and Francine Chen. 1995. A Trainable Document Summarizer. In *Proceedings of SIGIR'95*, pages 68 - 73, New York, USA.

Liangda Li, Ke Zhou, Gui-Rong Xue, Hongyuan Zha and Yong Yu. 2009. Enhancing Diversity, Coverage and Balance for Summarization through Structure Learning. In *Proceedings of the 18th International Conference on World Wide Web*.

Chin-Yew Lin and Eduard Hovy. 2003. Automatic Evaluation of Summaries Using N-gram Co-Occurrence Statistics. In *Proceedings of HLT-NAACL*, pages 71-78.

Tie-Yan Liu. 2009. Learning to Rank for Information Retrieval, Foundation and Trends on Information Retrieval. Now Publishers.

H.P. Luhn. 1958. The Automatic Creation of Literature Abstracts. In *IBM Journal of Research and Development*, Vol. 2, No. 2, pp. 159-165, April 1958.

André F. T. Martins and Noah A. Smith. 2009. Summarization with a Joint Model for Sentence Extraction and Compression. In *Proceedings of the Workshop on Integer Linear Programming for Natural Langauge Processing*\.

Ryan McDonald. 2007. A Study of Global Inference Algorithms in Multi-Document Summarization. In *Proceedings of the 29th ECIR*.

Kathleen McKeown and Dragomir R. Radev. 1995. Generating Summaries of Multiple News Articles. In *Proceedings of SIGIR'95*, pages 74–82.

Donald Metzler and Tapas Kanungo. 2008. Machine Learned Sentence Selection Strategies for Query-Biased Summarization. *SIGIR Learning to Rank Workshop*.

Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing Order into Texts. In *Proceedings of EMNLP 2004*, Barcelona, Spain, July 2004.

Ani Nenkova, Lucy Vanderwende and Kathleen McKeown. 2006. A Compositional Context Sensitive Multi-document Summarizer: Exploring the Factors that Influence Summarization. In *Proceedings of SIGIR 2006*.

You Ouyang, Sujian Li, Wenjie Li. 2007. Developing Learning Strategies for Topic-based Summarization. In *Proceedings of the sixteenth ACM Conference on Information and Knowledge Management*, 2007.

Rebecca J. Passonneau, Ani Nenkova, Kathleen McKeown and Sergey Sigelman. 2005. Applying the Pyramid Method in DUC 2005. *DUC 2005 Workshop*.

Dragomir R. Radev, Hongyan Jing, Malgorzata Stys, and Daniel Tam. 2004. Centroid-based Summarization of Multiple Documents. *Information Processing and Management*, 40:919–938.

Christina Sauper and Regina Barzilay. 2009. Automatically Generating Wikipedia Articles: A Structure-Aware Approach. In *Proceedings of ACL* 2009.

Dou Shen, Jian-Tao Sun, Hua Li, QiangYang and Zheng Chen. 2007. Document Summarization Using Conditional Random Fields. In *IJCAI*, pages 2862 - 2867, 2007.

Krysta Svore, Lucy Vanderwende, and Chris Burges. 2007. Enhancing Single-Document Summarization by Combining RankNet and Third-Party Sources. In *Proceedings of EMNLP-CoNLL (2007)*, pp. 448-457..

Hiroya Takamura and Manabu Okumura. Text Summarization Model Based on Maximum Coverage Problem and its Variant. In *Proceedings EACL*, 2009.

Xiaojun Wan and Jianguo Xiao. 2007. Towards a Unified Approach Based on Affinity Graph to Various Multi-document Summarizations. *ECDL 2007*, 297-308.

Changhu Wang, Feng Jing, Lei Zhang and Hong-Jiang Zhang. 2007. Learning Query-Biased Web Page Summarization. In *Proceedings of the sixteenth ACM Conference on Information and Knowledge Management*.

Frank Wilcoxon. 1945. Individual comparisons by ranking methods. Biometrics, 1, 80-83.

Identifying Contradictory and Contrastive Relations between Statements to Outline Web Information on a Given Topic

Daisuke Kawahara[†] **Kentaro Inui**[†‡] **Sadao Kurohashi**[†§]

[†]National Institute of Information and Communications Technology
[‡]Graduate School of Information Sciences, Tohoku University
[§]Graduate School of Informatics, Kyoto University

dk@nict.go.jp, inui@ecei.tohoku.ac.jp, kuro@i.kyoto-u.ac.jp

Abstract

We present a method for producing a bird's-eye view of statements that are expressed on Web pages on a given topic. This method aggregates statements that are relevant to the topic, and shows contradictory and contrastive relations among them. This view of contradictions and contrasts helps users acquire a top-down understanding of the topic. To realize this, we extract such statements and relations, including cross-document implicit contrastive relations between statements, in an unsupervised manner. Our experimental results indicate the effectiveness of our approach.

1 Introduction

The quantity of information on the Web is increasing explosively. Online information includes news reports, arguments, opinions, and other coverage of innumerable topics. To find useful information from such a mass of information, people generally use conventional search engines such as Yahoo! and Google. They input keywords to a search engine as a query and obtain a list of Web pages that are relevant to the keywords. They then use the list to check several dozen top-ranked Web pages one by one.

This method of information access does not provide a bird's-eye view of the queried topic; therefore it can be highly time-consuming and difficult for a user to gain an overall understanding of what is written on the topic. Also, browsing only top-ranked Web pages may provide the user with biased information. For example, when a user searches for information on "agaricus," claimed to be a health food, using a conventional search engine, many commercial pages touting its health benefits appear at the top of the ranks, while other pages remain low-ranked. The user may miss an existing Web page that indicates its unsubstantiated health benefits, and could be unintentionally satisfied by biased or one-sided information.

direct contrastive statement	"A is more P than B"
contrastive keyword pair	(A, B)
contradictory relation	"A is P" $\Leftrightarrow$ "A is not P"
contrastive relation	"A is P" $\leftrightarrow$ "B is P (not P)"

Table 1: Overview of direct contrastive statements, contrastive keyword pairs and contradictory/contrastive relations. Note that "P" is a predicate.

This paper proposes a method for producing a bird's-eye view of statements that are expressed on Web pages on a given query (topic). In particular, we focus on presenting contradictory/contrastive relations and statements on the topic. This presentation enables users to grasp what arguing points exist and furthermore to see contradictory/contrastive relations between them at a glance. Presenting these relations and statements is thought to facilitate users' understanding of the topic. This is because people typically think about contradictory and contrastive entities and issues for decision-making in their daily lives.

Our system presents statements and relations that are important and relevant to a given topic, including the statements and relations listed in Table 1. *Direct contrastive statements* compare two entities or issues in a single sentence. The contrasted entities or issues are also extracted as *contrastive keyword pairs*. In addition to them, our

Coling 2008: Poster Volume, pages 534–542,
Beijing, August 2010

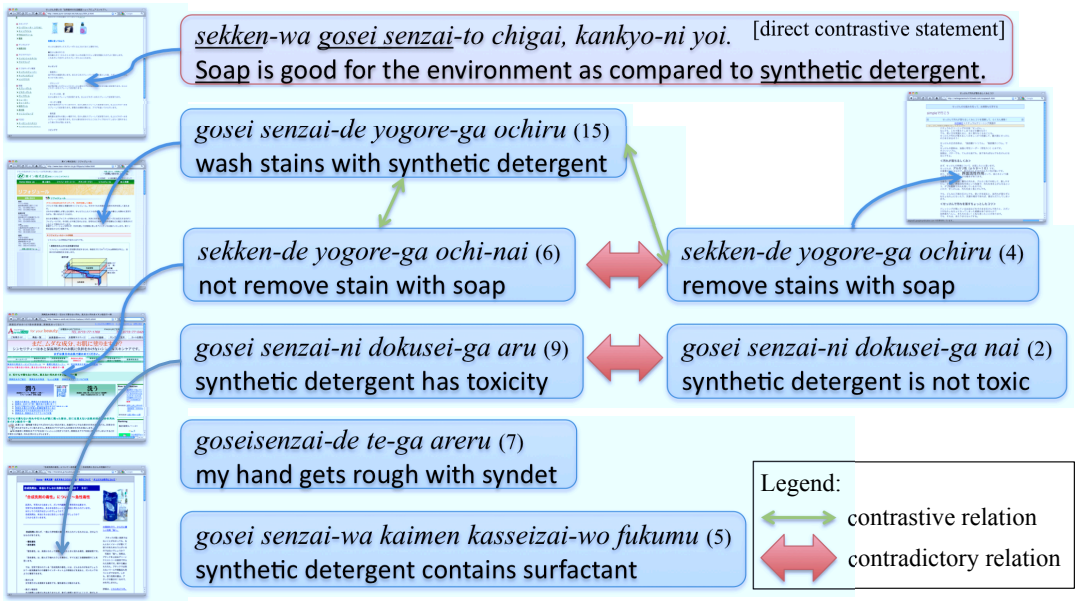


Figure 1: Examples of statements on "*gosei senzai*" (synthetic detergent), which are represented by rounded rectangles. Each statement is linked with the pages from which it is extracted. The number in a parenthesis represents the number of pages.

system shows *contradictory and contrastive relations* between statements. Contradictory relations are the relations between statements that are contradictory about an entity or issue. Contrastive relations are the relations between statements in which two entities or issues are contrasted.

In particular, we have the following two novel contributions.

- We identify contrastive relations between statements, which consist of in-document and cross-document implicit relations. These relations complement direct contrastive statements, which are explicitly mentioned in a single sentence.
- We precisely extract direct contrastive statements and contrastive keyword pairs in an unsupervised manner, whereas most previous studies used supervised methods (Jindal and Liu, 2006b; Yang and Ko, 2009).

Our system focuses on the Japanese language. For example, Figure 1 shows examples of extracted statements on the topic "*gosei senzai*" (synthetic detergent). Rounded rectangles represent statements relevant to this topic. The first statement is a direct contrastive statement, which refers to a contrastive keyword pair, "*gosei senzai*" (synthetic detergent) and "*sekken*" (soap). The pairs of statements connected with a broad arrow have contradictory relations. The pairs of statements connected with a thin arrow have contrastive relations. Users not only can see what is written on this topic at a glance, but also can check out the details of a statement by following its links to the original pages.

2 Related Work

Studies have been conducted on automatic extraction of direct contrastive sentences (comparative sentences) for English (Jindal and Liu, 2006b) and for Korean (Yang and Ko, 2009). They prepared a set of keywords that serve as clues to direct contrastive sentences and proposed supervised techniques on the basis of tagged corpora. We propose an unsupervised method for extracting direct contrastive sentences without constructing tagged corpora.

From direct contrastive sentences, Jindal and Liu (2006a) and Satou and Okumura (2007) proposed methods for extracting quadruples of (target, basis, attribute, evaluation). Jindal and Liu (2006a) extracted these quadruples and obtained an F-measure of 70%-80% for the extraction of "target" and "basis." Since this extraction was

not their main target, they did not perform error analysis on the extracted results. Satou and Okumura (2007) extracted quadruples from blog posts. They provided a pair of named entities for "target" and "basis," whereas we automatically identify such pairs. Ganapathibhotla and Liu (2008) proposed a method for detecting which entities ("target" and "basis") in a direct contrastive statement are preferred by its author.

There is also related work that focuses on non-contrastive sentences. Ohshima et al. (2006) extracted coordinated terms, which are semantically broader than our contrastive keyword pairs, using hit counts from a search engine. They made use of syntactic parallelism among coordinated terms. Their task was to input one of coordinated terms as a query, which is different from ours. Somasundaran and Wiebe (2009) presented a method for recognizing a stance in online debates. They formulated this task as debate-side classification and solved it by using automatically learned probabilities of polarity.

To aggregate statements and detect relations between them, one of important modules is recognition of synonymous, entailed, contradictory and contrastive statements. Studies on rhetorical structure theory (Mann and Thompson, 1988) and recognizing textual entailment (RTE) deal with these relations. In particular, evaluative workshops on RTE have been held and this kind of research has been actively studied (Bentivogli et al., 2009). The recent workshops of this series set up a task that recognizes contradictions. Harabagiu et al. (2006), de Marneffe et al. (2008), Voorhees (2008), and Ritter et al. (2008) focused on recognizing contradictions. For example, Harabagiu et al. (2006) used negative expressions, antonyms and contrast discourse relations to recognize contradictions. These methods only detect relations between given sentences, and do not create a bird's-eye view.

To create a kind of bird's-eye view, Kawahara et al. (2008), Statement Map (Murakami et al., 2009) and Dispute Finder (Ennals et al., 2010) identified various relations between statements including contradictory relations, but do not handle contrastive relations, which are one of the important relations for taking a bird's-eye view on a topic. Lerman and McDonald (2009) proposed a method for generating contrastive summaries about given two entities on the basis of KL-divergence. This study is related to ours in the aspect of extracting implicit contrasts, but contrastive summaries are different from contrastive relations between statements in our study.

3 Our Method

We propose a method for grasping overall information on the Web on a given query (topic). This method extracts and presents statements that are relevant to a given topic, including direct contrastive statements and contradictory/contrastive relations between these statements.

As a unit for statements, we use a predicate-argument structure (also known as a case structure and logical form). A predicate-argument structure represents a "who does what" event. Processes such as clustering, summarization, comparison with other knowledge and logical consistency verification, which are required for this study and further analysis, are accurately performed on the basis of predicate-argument structures. The extraction of our target relations and statements is performed via identification and aggregation of synonymous, contrastive, and contradictory relations between predicate-argument structures.

As stated in section 1, we extract direct contrastive statements, contrastive keyword pairs, relevant statements, contrastive relations and contradictory relations. We do this with the following steps:

1. Extraction and aggregation of predicate-argument structures
2. Extraction of contrastive keyword pairs and direct contrastive statements
3. Identification of contradictory relations
4. Identification of contrastive relations

Below, we first describe our method of extracting and aggregating predicate-argument structures. Then, we explain our method of extracting direct contrastive statements with contrastive keyword pairs, and identifying contradictory and contrastive relations in detail.

3.1 Extraction and Aggregation of Predicate-argument Structures

A predicate-argument structure consists of a predicate and one or more arguments that have a dependency relation to the predicate.

We extract predicate-argument structures from automatic parses of Web pages on a given topic by using the method of Kawahara et al. (2008). We apply the following procedure to Web pages that are retrieved from the TSUBAKI (Shinzato et al., 2008) open search engine infrastructure, by inputting the topic as a query.

1. Extract important sentences from each Web page. Important sentences are defined as sentences neighboring the topic word(s).
2. Obtain results of morphological analysis (JUMAN[1]) and dependency parsing (KNP[2]) of the important sentences, and extract predicate-argument structures from them.
3. Filter out functional and meaningless predicate-argument structures, which are not relevant to the topic. Pointwise mutual information between the entire Web and the target Web pages for a topic is used.

Note that the analyses in step 2 are performed beforehand and stored in an XML format (Shinzato et al., 2008).

Acquired predicate-argument structures vary widely in their representations of predicates and arguments. In particular, many separate predicate-argument structures have the same meaning due to spelling variations, transliterations, synonymous expressions and so forth. To cope with this problem, we apply "keyword distillation" (Shibata et al., 2009), which is a process of absorbing spelling variations, synonymous expressions and keywords with part-of relations on a set of Web pages about a given topic. As a knowledge source to merge these expressions, this process uses a knowledge base that is automatically extracted from an ordinary dictionary and the Web. For instance, the following predicate-argument structures are judged to be synonymous[3].

(1) a. *sekken-wo tsukau*
soap-ACC use

b. *sopu-wo tsukau*
soap-ACC use

c. *sekken-wo shiyou-suru*
soap-ACC utilize

We call the predicate-argument structures that are obtained as the result of the above procedure **statement candidates**. The final output of our system consists of direct contrastive statements (with contrastive keyword pairs), top-N statements (**major statements**) in order of frequency of statement candidates, and statements with contradictory/contrastive relations. Contradictory/contrastive relations are identified against major statements by searching statement candidates.

Another outcome of keyword distillation is a resultant set of keywords that are important and relevant to the topic. We call this set of keywords **relevant keywords**, which also include words or phrases in the query. Relevant keywords are used to extract contrastive keyword pairs.

3.2 Extraction of Contrastive Keyword Pairs and Direct Contrastive Statements

We extract contrastive keyword pairs from contrastive constructs, which are manually specified as patterns of predicate-argument structures. Statements that contain contrastive constructs are defined as direct contrastive statements.

For example, the following sentence is a typical direct contrastive statement, which contains a contrastive verb "*chigau*" (differ).

(2) *sekken-wa gosei senzai-to chigai, · · ·*
soap-TOP synthetic detergent-ABL differ

(soap differs from synthetic detergent, · · ·)

From this sentence, a contrastive keyword pair, "*sekken*" (soap) and "*gosei senzai*" (synthetic detergent), is extracted. The above sentence is extracted as a direct contrastive statement.

We preliminarily evaluated this simple pattern-based method and found that it has the following three problems.

[1] http://nlp.kuee.kyoto-u.ac.jp/nl-resource/juman-e.html

[2] http://nlp.kuee.kyoto-u.ac.jp/nl-resource/knp-e.html

[3] In this paper, we use the following abbreviations: NOM (nominative), ACC (accusative), DAT (dative), ABL (ablative), CMI (comitative), GEN (genitive) and TOP (topic marker).

- Keyword pairs that are mentioned in a contrastive construct are occasionally not relevant to the given topic.
- Non-contrastive keyword pairs are erroneously extracted due to omissions of attributes and targets of comparisons.
- Non-contrastive keyword pairs that have an is-a relation are erroneously extracted.

To deal with the first problem, we filter out keyword pairs that are contrastive but that are not relevant to the topic. For this purpose, we apply filtering by using relevant keywords, which are described in section 3.1.

As an example of non-contrastive keyword pairs (the second problem), from the following sentence, a keyword pair, "*tokkyo seido*" (patent system) and "*nihon*" (Japan), is incorrectly extracted by the pattern-based method.

(3) *amerika-no tokkyo seido-wa nihon-to*
America-GEN patent system-TOP Japan-ABL
kotonari, $\cdots$
different
(patent system of America is different from ϕ of Japan $\cdots$)

In this sentence, "*nihon*" (Japan) has a meaning of "*nihon-no tokkyo seido*" (patent system of Japan). That is to say, "*tokkyo seido*" (patent system), which is the attribute of comparison, is omitted.

In this study, in addition to patterns of contrastive constructs, we use checking and filtering on the basis of similarity. The use of similarity is inspired by the semantic parallelism between contrasted keywords. As this similarity, we employ distributional similarity (Lin, 1998), which is calculated using automatic dependency parses of 100 million Japanese Web pages. By searching similar keywords from the above sentence, we successfully extract a contrastive keyword pair, "*amerika*" (America) and "*nihon*" (Japan), and the above sentence as a direct contrastive statement.

Similarly, a target of comparison can be omitted as in the following sentence.

(4) *nedan-wa gosei senzai-yori takaidesu*
price-TOP synthetic detergent-ABL high
(price of ϕ is higher than synthetic detergent)

In this example, the similarity between "*nedan*" (price) and "*gosei senzai*" (synthetic detergent) is lower than a threshold, and this sentence and the extracts from it are filtered out.

As for the third problem, we may extract non-contrastive keyword pairs that have an is-a relation. From the following sentence, we incorrectly extract a contrastive keyword pair, "*konbini*" (convenience store) and "*7-Eleven*," which cannot be filtered out due to its high similarity.

(5) *7-Eleven-wa hokano konbini-to*
7-Eleven-TOP other convenience store-ABL
kurabete, $\cdots$
compare
(7-Eleven is $\cdots$ compared to other convenience stores)

To deal with this problem, we use a filter on the basis of a set of words that indicate the existence of hypernyms, such as "*hokano*" (other) and *ippanno* (general). We prepare six words for this purpose.

To sum up, we use the following procedure to identify contrast keyword pairs.

1. Extract predicate-argument structures that do not match the above is-a patterns and match one of the following patterns. They are extracted from the statement candidates.
   - X-*wa* Y-*to* {*chigau* | *kotonaru* | *kuraberu*}
     (X {differ | vary | compare} from/with Y)
   - X-*wa* Y-*yori* [adjective]
     (X is more $\cdots$ than Y)

   Note that each of X and Y is a noun phrase in the argument position.
2. Extract (x, y) that satisfies both the following conditions as a contrastive keyword pair. Note that (x, y) is part of a word sequence in (X, Y), respectively.
   - Both x and y are included in a set of relevant keywords.
   - (x, y) has the highest similarity among any other candidates of (x, y), and this similarity is higher than a threshold.

Note that the threshold is determined based on a preliminary experiment using a set of synonyms (Aizawa, 2007). We extract the sentence that contains the predicate-argument structure used in step 1 as a direct contrastive statement.

3.3 Identification of Contradictory Relations

We identify contradictory relations between statement candidates. In this paper, contradictory relations are defined as the following two types (Kawahara et al., 2008).

negation of predicate

If the predicate of a candidate statement is negated, its contradiction has the same or synonymous predicate without negation. If not, its contradiction has the same or synonymous predicate with negation.

(6) a. *sekken-ga kankyou-ni yoi*
soap-NOM environment-DAT good

b. *sekken-ga kankyou-ni yoku-nai*
soap-NOM environment-DAT not good

antonym of predicate

The predicate of a contradiction is an antonym of that of a candidate statement. To judge antonymous relations, we use an antonym lexicon extracted from a Japanese dictionary (Shibata et al., 2008). This lexicon consists of approximately 2,000 entries.

(7) a. *gosei senzai-ga anzen-da*
synthetic detergent-NOM safe

b. *gosei senzai-ga kiken-da*
synthetic detergent-NOM dangerous

To identify contradictory relations between statements in practice, we search statement candidates that satisfy one of the above conditions against major statements.

3.4 Identification of Contrastive Relations

We identify contrastive relations between statement candidates. In this paper, we define a contrastive relation as being between a pair of statement candidates whose arguments are contrastive keyword pairs and whose predicates have synonymous or contradictory relations. Contradictory relations of predicates are defined in the same way as section 3.3.

In the following example, (a, b) and (a, c) have a contrastive relation. Also, (b, c) has a contradictory relation.

(8) a. *gosei senzai-de yogore-ga ochiru*
synthetic detergent-CMI stain-NOM wash

Topic: bio-ethanol
(bio-ethanol fuel, gasoline)
(bio-ethanol car, electric car)
Topic: citizen judgment system
(citizen judgment system, jury system)
(citizen judgment system, lay judge system)
Topic: patent system
(patent system, utility model system)
(large enterprise, small enterprise)
Topic: Windows Vista
(Vista, XP)

Table 2: Examples of extracted contrastive keyword pairs (translated into English).

b. *sekken-de yogore-ga ochiru*
soap-CMI stain-NOM wash

c. *sekken-de yogore-ga ochi-nai*
soap-CMI stain-NOM not wash

The process of identifying contrastive relations between statements is performed in the same way as the identification of contradictory relations. That is to say, we search statement candidates that satisfy the definition of contrastive relations against major statements.

4 Experiments

We conducted experiments for extracting contrastive keyword pairs, direct contrastive statements and contradictory/contrastive relations on 50 topics, such as age of adulthood, anticancer drug, bio-ethanol, citizen judgment system, patent system and Windows Vista.

We retrieve at most 1,000 Web pages for a topic from the search engine infrastructure, TSUBAKI. As major statements, we extract 10 statement candidates in order of frequency.

Below, we first evaluate the extracted contrastive keyword pairs and direct contrastive statements, and then evaluate the identified contradictory and contrastive relations between statements.

4.1 Evaluation of Contrastive Keyword Pairs and Direct Contrastive Statements

Contrastive keyword pairs and direct contrastive statements were extracted on 30 of 50 topics. 99 direct contrastive statements and 73 unique contrastive keyword pairs were obtained on 30 topics. The average number of obtained contrastive keyword pairs for a topic was approximately 2.4. Ta-

Topic: "*tyosakuken hou*" (copyright law)		
"*syouhyouken-wa tyosakuken-yori zaisantekina kachi-wo motsu.*"		
The trademark right has more financial value than the copyright.		
"*tyosakuken hou-de hogo-sareru*" protected by the copyright law	⇔	"*tyosakuken hou-de hogo-sare-nai*" not protected by the copyright law
"*tyosakuken-wo shingai-suru*" infringe the copyright	⇔	"*tyosakuken-wo shingai-shi-nai*" not infringe the copyright
	↖↘	"*syouhyouken-wo shingai-shi-nai*" not infringe the trademark right
Topic: "*genshiryoku hatsuden syo*" (nuclear power plant)		
"*genshiryoku hatsuden syo-wa karyoku hatsuden syo-to chigau.*"		
Nuclear power plants are different from thermoelectric power plants.		
"*CO2-wo hassei-shi-nai*" not emit carbon dioxide	⇔	"*CO2-wo hassei-suru*" emit carbon dioxide
"*genpatsu-wo tsukuru*" construct a nuclear power plant	⇔	"*genshiryoku hatsuden syo-wo tsukura-nai*" not construct a nuclear power plant
	↖↘	"*karyoku hatsuden syo-wo tsukuru*" construct a thermoelectric power plant

Table 3: Examples of identified direct contrastive statements, contradictory relations and contrastive relations. The sentences with two underlined parts are direct contrastive statements. The arrows "⇔" and "↔" represent a contradictory relation and a contrastive relation, respectively.

ble 2 lists examples of obtained contrastive keyword pairs. We successfully extracted not only contrastive keyword pairs including topic words, but also those without them.

Our manual evaluation of the extracted contrastive keyword pairs found that 89% (65/73) of the contrastive keyword pairs are actually contrasted in direct contrastive statements. Correct contrastive keyword pairs were extracted on 28 of 30 topics. We also evaluated the contrastive keyword pairs extracted without similarity filtering. In this case, 190 contrastive keyword pairs on 41 topics were extracted and 44% (84/190) of them were correct. Correct contrastive keyword pairs were extracted on 31 of 41 topics. Therefore, similarity filtering did not largely decrease the recall, but significantly increased the precision.

We have eight contrastive keyword pairs that were incorrectly extracted by our proposed method. These contrastive keyword pairs accidentally have similarity that is higher than the threshold. Major errors were caused by the ambiguity of Japanese ablative keyword "*yori*."

(9) *heisya-wa* *bitWallet sya-yori*
our company-TOP bitWallet, Inc.-ABL

Edy gifuto-no gyomu itaku-wo ukete-imasu
Edy gift-GEN entrustment-ACC have

(Our company is entrusted with Edy gift by bitWallet, Inc.)

In this example, "*yori*" means not the basis of contrast but the source of action. The similarity filtering usually prevents incorrect extraction from such a non-contrastive sentence. However, in this case, the pair of "*heisya*" (our company) and "*bitWallet sya*" (bitWallet, Inc.) was not filtered due to the high similarity between them. To cope with this problem, it is necessary to use linguistic knowledge such as case frames.

4.2 Evaluation of Contradictory and Contrastive Relations

Contradictory relations were identified on 49 of 50 topics. For 49 topics, 268 contradictory relations were identified. The average number of identified contradictory relations for a topic was 5.5. Contrastive relations were identified on 18 of 30 topics, on which contrastive keyword pairs were extracted. For the 18 topics, 60 contrastive relations were identified. The average number of identified contrastive relations for a topic was 3.3.

Table 3 lists examples of the identified contradictory and contrastive relations as well as direct contrastive statements. We manually evaluated the identified contradictory relations and the contrastive relations that were identified for correct contrastive keyword pairs. As a result, we concluded that they completely obey our definitions.

We also classified each of the obtained contradictory and contrastive relations into two classes: "cross-document" and "in-document." "Cross-

Topic: age of adulthood
lower the age of adulthood to 18
↔ lower the voting age to 18
Topic: anticancer drug
anticancer drugs have side effects
↔ anticancer drugs have effects

Table 4: Examples of unidentified contrastive relations (translated into English).

document" means that a contradictory/contrastive relation is obtained not from a single page but across multiple pages. If a relation can be obtained from both, we classified it into "in-document." As a result, 67% (179/268) of contradictory relations and 70% (42/60) of contrastive relations were "cross-document." We can see that many cross-document implicit relations that cannot be retrieved from a single page were successfully identified.

4.3 Discussions

We successfully identified contradictory relations on almost all the topics. However, out of 50 topics, we extracted contrastive keyword pairs on 30 topics and contrastive relations on 18 topics. To investigate the resultant contrastive relations from the viewpoint of recall, we manually checked whether there were unidentified contrastive relations among 100 statement candidates for each topic. We actually checked 20 topics and found six unidentified contrastive relations in total. Table 4 lists examples of the unidentified contrastive relations. Out of 20 topics, in total, 44 contrastive relations are manually discovered on 13 topics, but out of 13 topics, 38 contrastive relations are identified on eight topics by our method. Therefore, we achieved a recall of 86% (38/44) at relation level and 62% (8/13) at topic level. We can see that our method was able to cover a relatively wide range of contrastive relations on the topics on which our method successfully extracted contrastive keyword pairs.

To detect such unidentified contrastive relations, it is necessary to robustly extract contrastive keyword pairs. In the future, we will employ a bootstrapping approach to identify patterns of direct contrastive statements and contrastive keyword pairs. We will also use patterns of contrastive discourse structures as well as those of predicate-argument structures.

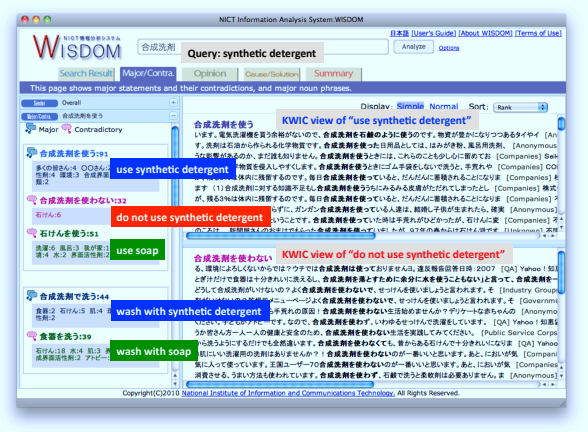


Figure 2: A view of major, contradictory and contrastive statements in WISDOM.

5 Conclusion

This paper has described a method for producing a bird's-eye view of statements that are expressed in Web pages on a given topic. This method aggregates statements relevant to the topic and shows the contradictory/contrastive relations and statements among them.

In particular, we successfully extracted direct contrastive statements in an unsupervised manner. We specified only several words for the extraction patterns and the filtering. Therefore, our method for Japanese is thought to be easily adapted to other languages. We also proposed a novel method for identifying contrastive relations between statements, which included cross-document implicit relations. These relations complemented direct contrastive statements.

We have incorporated our proposed method into an information analysis system, WISDOM[4] (Akamine et al., 2009), which can show multi-faceted information on a given topic. Now, this system can show contradictory/contrastive relations and statements as well as their contexts as a view of KWIC (keyword in context) (Figure 2). This kind of presentation facilitates users' understanding of an input topic.

[4]http://wisdom-nict.jp/

References

Aizawa, Akiko. 2007. On calculating word similarity using web as corpus. In *Proceedings of IEICE Technical Report, SIG-ICS*, pages 45–52 (in Japanese).

Akamine, Susumu, Daisuke Kawahara, Yoshikiyo Kato, Tetsuji Nakagawa, Kentaro Inui, Sadao Kurohashi, and Yutaka Kidawara. 2009. WISDOM: A web information credibility analysis system. In *Proceedings of the ACL-IJCNLP 2009 Software Demonstrations*, pages 1–4.

Bentivogli, Luisa, Ido Dagan, Hoa Dang, Danilo Giampiccolo, and Bernardo Magnini. 2009. The fifth PASCAL recognizing textual entailment challenge. In *Proceedings of TAC 2009 Workshop*.

de Marneffe, Marie-Catherine, Anna N. Rafferty, and Christopher D. Manning. 2008. Finding contradictions in text. In *Proceedings of ACL-08: HLT*, pages 1039–1047.

Ennals, Rob, Beth Trushkowsky, and John Mark Agosta. 2010. Highlighting disputed claims on the web. In *Proceedings of WWW 2010*.

Ganapathibhotla, Murthy and Bing Liu. 2008. Mining opinions in comparative sentences. In *Proceedings of COLING 2008*, pages 241–248.

Harabagiu, Sanda, Andrew Hickl, and Finley Lacatusu. 2006. Negation, contrast and contradiction in text processing. In *Proceedings of AAAI-06*.

Jindal, Nitin and Bing Liu. 2006a. Identifying comparative sentences in text documents. In *Proceedings of SIGIR 2006*.

Jindal, Nitin and Bing Liu. 2006b. Mining comparative sentences and relations. In *Proceedings of AAAI-06*.

Kawahara, Daisuke, Sadao Kurohashi, and Kentaro Inui. 2008. Grasping major statements and their contradictions toward information credibility analysis of web contents. In *Proceedings of WI'08, short paper*, pages 393–397.

Lerman, Kevin and Ryan McDonald. 2009. Contrastive summarization: An experiment with consumer reviews. In *Proceedings of NAACL-HLT 2009, Companion Volume: Short Papers*, pages 113–116.

Lin, Dekang. 1998. Automatic retrieval and clustering of similar words. In *Proceedings of COLING-ACL98*, pages 768–774.

Mann, William and Sandra Thompson. 1988. Rhetorical structure theory: toward a functional theory of text organization. *Text*, 8(3):243–281.

Murakami, Koji, Eric Nichols, Suguru Matsuyoshi, Asuka Sumida, Shouko Masuda, Kentaro Inui, and Yuji Matsumoto. 2009. Statement map: Assisting information credibility analysis by visualizing arguments. In *Proceedings of WICOW 2009*.

Ohshima, Hiroaki, Satoshi Oyama, and Katsumi Tanaka. 2006. Searching coordinate terms with their context from the web. In *Proceedings of WISE 2006*, pages 40–47.

Ritter, Alan, Stephen Soderland, Doug Downey, and Oren Etzioni. 2008. It's a contradiction – no, it's not: A case study using functional relations. In *Proceedings of EMNLP 2008*, pages 11–20.

Satou, Toshinori and Manabu Okumura. 2007. Extraction of comparative relations from Japanese weblog. In *IPSJ SIG Technical Report 2007-NL-181*, pages 7–14 (in Japanese).

Shibata, Tomohide, Michitaka Odani, Jun Harashima, Takashi Oonishi, and Sadao Kurohashi. 2008. SYNGRAPH: A flexible matching method based on synonymous expression extraction from an ordinary dictionary and a web corpus. In *Proceedings of IJCNLP 2008*, pages 787–792.

Shibata, Tomohide, Yasuo Banba, Keiji Shinzato, and Sadao Kurohashi. 2009. Web information organization using keyword distillation based clustering. In *Proceedings of WI'09, short paper*, pages 325–330.

Shinzato, Keiji, Tomohide Shibata, Daisuke Kawahara, Chikara Hashimoto, and Sadao Kurohashi. 2008. TSUBAKI: An open search engine infrastructure for developing new information access methodology. In *Proceedings of IJCNLP 2008*, pages 189–196.

Somasundaran, Swapna and Janyce Wiebe. 2009. Recognizing stances in online debates. In *Proceedings of ACL-IJCNLP 2009*, pages 226–234.

Voorhees, Ellen M. 2008. Contradictions and justifications: Extensions to the textual entailment task. In *Proceedings of ACL-08: HLT*, pages 63–71.

Yang, Seon and Youngjoong Ko. 2009. Extracting comparative sentences from korean text documents using comparative lexical patterns and machine learning techniques. In *Proceedings of ACL-IJCNLP 2009 Conference Short Papers*, pages 153–156.

Generative Alignment and Semantic Parsing for Learning from Ambiguous Supervision

Joohyun Kim
Department of Computer Science
The University of Texas at Austin
scimitar@cs.utexas.edu

Raymond J. Mooney
Department of Computer Science
The University of Texas at Austin
mooney@cs.utexas.edu

Abstract

We present a probabilistic generative model for learning semantic parsers from ambiguous supervision. Our approach learns from natural language sentences paired with world states consisting of multiple potential logical meaning representations. It disambiguates the meaning of each sentence while simultaneously learning a semantic parser that maps sentences into logical form. Compared to a previous generative model for semantic alignment, it also supports full semantic parsing. Experimental results on the Robocup sportscasting corpora in both English and Korean indicate that our approach produces more accurate semantic alignments than existing methods and also produces competitive semantic parsers and improved language generators.

1 Introduction

Most approaches to learning semantic parsers that map sentences into complete logical forms (Zelle and Mooney, 1996; Zettlemoyer and Collins, 2005; Kate and Mooney, 2006; Wong and Mooney, 2007b; Lu et al., 2008) require fully-supervised corpora that provide full formal logical representations for each sentence. Such corpora are expensive and difficult to construct. Several recent projects on "grounded" language learning (Kate and Mooney, 2007; Chen and Mooney, 2008; Chen et al., 2010; Liang et al., 2009) exploit more easily and naturally available training data consisting of sentences paired with world states consisting of multiple potential semantic representations. This setting is partially motivated by a desire to model how children naturally learn language in the context of a rich, ambiguous perceptual environment.

In particular, Chen and Mooney (2008) introduced the problem of learning to sportscast by simply observing natural language commentary on simulated Robocup robot soccer games. The training data consists of natural language (NL) sentences ambiguously paired with logical meaning representations (MRs) describing recent events in the game extracted from the simulator. Most sentences describe one of the extracted recent events; however, the specific event to which it refers is unknown. Therefore, the learner has to figure out the correct matching (*alignment*) between NL and MR before inducing a semantic parser or language generator. Based on an approach introduced by Kate and Mooney (2007), Chen and Mooney (2008) repeatedly retrain both a supervised semantic parser and language generator using an iterative algorithm analogous to Expectation Maximization (EM). However, this approach is somewhat ad hoc and does not exploit a well-defined probabilistic generative model or real EM training.

On the other hand, Liang et al. (2009) introduced a probabilistic generative model for learning semantic correspondences in ambiguous training data consisting of sentences paired with observed world states. Compared to Chen and Mooney (2008), they demonstrated improved alignment results on Robocup sportscasting data. However, their model only produces an NL–MR alignment and does *not* learn either an effective

Coling 2008: Poster Volume, pages 543–551,
Beijing, August 2010

semantic parser or language generator. In addition, they use a combination of a simple Markov model and a bag-of-words model when generating natural language for MRs, therefore, they do not model context-free linguistic syntax.

Motivated by the limitations of these previous methods, we propose a new generative alignment model that includes a full semantic parsing model proposed by Lu et al. (2008). Our approach is capable of disambiguating the mapping between language and meanings while also learning a complete semantic parser for mapping sentences to logical form. Experimental results on Robocup sportscasting show that our approach outperforms all previous results on the NL–MR matching (alignment) task and also produces competitive performance on semantic parsing and improved language generation.

2 Related Work

The conventional approach to learning semantic parsers (Zelle and Mooney, 1996; Ge and Mooney, 2005; Kate and Mooney, 2006; Zettlemoyer and Collins, 2007; Zettlemoyer and Collins, 2005; Wong and Mooney, 2007b; Lu et al., 2008) requires detailed supervision unambiguously pairing each sentence with its logical form. However, developing training corpora for these methods requires expensive expert human labor.

Chen and Mooney (2008) presented methods for grounded language learning from ambiguous supervision that address three related tasks: NL–MR alignment, semantic parsing, and natural language generation. They solved the problem of aligning sentences and meanings by iteratively retraining an existing supervised semantic parser, WASP (Wong and Mooney, 2007b) or KRISP (Kate and Mooney, 2006), or an existing supervised natural-language generator, WASP^{-1} (Wong and Mooney, 2007a). During each iteration, the currently trained parser (generator) is used to produce an improved NL–MR alignment that is used to retrain the parser (generator) in the next iteration. However, this approach does not use the power of a probabilistic correspondence between an NL and MRs during training.

On the other hand, Liang et al. (2009) proposed a probabilistic generative approach to produce a Viterbi alignment between NL and MRs. They use a hierarchical semi-Markov generative model that first determines which facts to discuss and then generates words from the predicates and arguments of the chosen facts. They report improved matching accuracy in the Robocup sportscasting domain. However, they only addressed the alignment problem and are unable to parse new sentences into meaning representations or generate natural language from logical forms. In addition, the model uses a weak bag-of-words assumption when estimating links between NL segments and MR facts. Although it does use a simple Markov model to order the generation of the different fields of an MR record, it does not utilize the full syntax of the NL or MR or their relationship.

Chen et al. (2010) recently reported results on utilizing the improved alignment produced by Liang et al. (2009)'s model to initialize their own iterative retraining method. By combining the approaches, they produced more accurate NL–MR alignments and improved semantic parsers.

Motivated by this prior research, our approach combines the generative alignment model of Liang et al. (2009) with the generative semantic parsing model of Lu et al. (2008) in order to fully exploit the NL syntax and its relationship to the MR semantics. Therefore, unlike Liang et al.'s simple Markov + bag-of-words model for generating language, it uses a tree-based model to generate grammatical NL from structured MR facts.

3 Background

This section describes existing models and algorithms employed in the current research. Our model is built on top of the generative semantic parsing model developed by Lu et al. (2008). After learning a probabilistic alignment and parsing model, we also used the WASP and WASP^{-1} systems to produce additional parsing and generation results. In particular, since our current system is incapable of effectively generating NL sentences from MR logical forms, in order to demonstrate how our matching results can aid NL generation, we use WASP^{-1} to learn a generator. This follows the experimental scheme of Chen et al. (2010), which demonstrated that an improved NL–MR

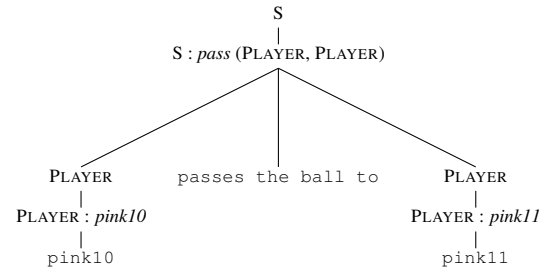


Figure 1: Sample hybrid tree from English sportscasting dataset where $(\mathbf{w}, \mathbf{m})$ = (pink10 passes the ball to pink11, *pass*(*pink10*, *pink11*))

matching from Liang et al. (2009) results in better overall parsing and generation. Finally, our overall generative model uses the IGSL (Iterative Generation Strategy Learning) method of Chen and Mooney (2008) to initially estimate the prior probability of each event-type generating a natural-language comment.

3.1 Generative Semantic Parsing

Lu et al. (2008) introduced a generative semantic parsing model using a hybrid-tree framework. A *hybrid tree* is defined over a pair, $(\mathbf{w}, \mathbf{m})$, of a natural-language sentence and its logical meaning representation. The tree expresses a correspondence between word segments in the NL and the grammatical structure of the MR. In a hybrid tree, MR production rules constitute the internal nodes, while NL words (or phrases) constitute the leaves. A sample hybrid tree from the English Robocup data is given in Figure 1.

A generative model based on hybrid trees is defined as follows: starting from a root semantic category, the model generates a production of the MR grammar, and then subsequently generates a mixed hybrid pattern of NL words and child semantic categories. This process is repeated until all leaves in the hybrid tree are NL words (or phrases). Each generation step is only dependent on the parent step, thus, generation is assumed to be a Markov process.

Lu et al. (2008)'s generative parsing model estimates the joint probability $P(\mathcal{T}, \mathbf{w}, \mathbf{m})$, which represents the probability of generating a hybrid tree $\mathcal{T}$ with NL $\mathbf{w}$, and MR $\mathbf{m}$. This probability is computed as the product of the probabilities of the steps in the generative process. Since there are multiple ways to construct a hybrid tree given a pair of NL and MR, the data likelihood of the pair $(\mathbf{w}, \mathbf{m})$ given by the learned model is calculated by summing $P(\mathcal{T}, \mathbf{w}, \mathbf{m})$ over all the possible hybrid trees for NL $\mathbf{w}$ and MR $\mathbf{m}$.

The model is normally trained in a fully supervised setting using NL–MR pairs. In order to learn from ambiguous supervision, we extend this model to include an additional generative process for selecting the subset of available MRs used to generate NL sentences.

3.2 WASP and WASP^{-1}

WASP (Word-Alignment-based Semantic Parsing) is a semantic parsing system that uses syntax-based statistical machine translation techniques. It induces a probabilistic synchronous context-free grammar (PSCFG) for generating corresponding NL–MR pairs. Since a PSCFG is symmetric with respect to the two languages it generates, the same learned model can be used for both semantic parsing (mapping NL to MR) and natural language generation (mapping MR to NL), Since there is no prespecified formal grammar for the NL, the WASP^{-1} system learns an n-gram language model for the NL side and uses it to choose the most probable NL translation for a given MR using a noisy-channel model.

3.3 IGSL

Chen and Mooney (2008) introduced the IGSL method for determining which event types a human commentator is more likely to describe in natural language. This is sometimes called *strategic generation* or *content selection*, the process of choosing *what to say*; as opposed to *tactical generation*, which determines *how to say it*. IGSL uses a method analogous to EM to train on ambiguously supervised data and iteratively improve probability estimates for each event type, specifying how likely each MR predicate is to elicit a comment. The algorithm alternates between two processes: calculating the expected probability of an NL–MR matching based on the currently learned estimates, and updating the probability of each event type based on the expected match counts. IGSL was shown to be quite effective at predicting which events in a Robocup game

	English	Korean
# of NL comments	2036	1999
# of extracted MR events	10452	10668
# of NLs w/ matching MRs	1868	1913
# of MRs w/ matching NLs	4670	4610
Avg. # of MRs per NL	2.50	2.41

Table 1: Stats for Robocup sportscasting data

a human would comment upon. In our proposed model, we use IGSL probability scores as initial priors for our event selection model.

4 Evaluation Dataset

In our experiments, we use the Robocup sportscasting data produced by Chen et al. (2010), which includes both English and Korean commentaries. The data was collected by having both English and Korean speakers commentate the final games from the RoboCup simulation soccer league for each year from 2001 through 2004. Table 1 presents some statistics on this sportscasting data. To construct the ambiguous training data, each NL commentary sentence is paired with MRs for all extracted simulation events that occurred in the previous 5 seconds (an average of 2.5 events).

Figure 2 shows a sample trace from the Robocup English data. Each NL commentary sentence normally has several possible MR matches that occurred within the 5-second window, indicated by edges between the NL and MR. Bold edges represent gold standard matches constructed solely for evaluation purposes. Note that not every NL has a gold matching MR. This occurs because the sentence refers to unrecognized or undetected events or situations or because the matching MR lies outside the 5-second window.

5 Generative Model

Like Liang et al. (2009)'s generative alignment model, our model is designed to estimate $P(\mathbf{w}|\mathbf{s})$, where $\mathbf{w}$ is an NL sentence and $\mathbf{s}$ is a world state containing a set of possible MR logical forms that can be matched to $\mathbf{w}$. However, our approach is intended to support both determining the most likely match between an NL and its MR in its world state, **and** semantic parsing, i.e. finding the

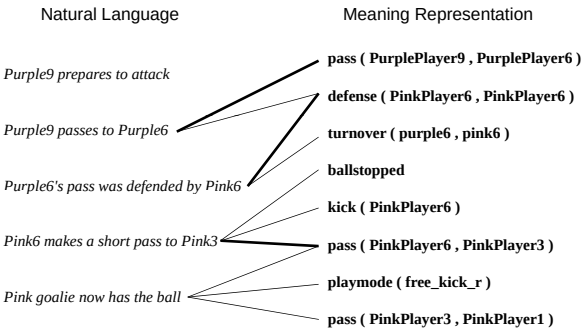


Figure 2: Sample trace from Robocup English data.

most probable mapping from a given NL sentence to an MR logical form.

Our generative model consists of two stages:

- Event selection: $P(\mathbf{e}|\mathbf{s})$, chooses the event $\mathbf{e}$ in the world state $\mathbf{s}$ to be described.
- Natural language generation: $P(\mathbf{w}|\mathbf{e})$, models the probability of generating natural-language sentence $\mathbf{w}$ from the MR specified by event $\mathbf{e}$.

5.1 Event selection model

The event selection model specifies the probability distribution for picking an event that is likely to be commented upon amongst the multiple MR logical forms in the world state $\mathbf{s}$. The probability of picking an event is assumed to depend only on its event type as given by the predicate of its MR. For example, the MR $pass(pink10, pink11)$ has event type $pass$ and arguments $pink10$ and $pink11$.

Our model is similar to Liang et al. (2009)'s *record choice* model, but we only model their notion of *salience*, denoting that some event types are more likely to be described than others. We do not model their notion of *coherence*, which models the order of event types in the commentary. We found that for sportscasting the order of described events depends only on the sequence of events in the game and does not exhibit any additional detectable pattern due to linguistic preferences.

The probability of picking an event e of type t_e is denoted by $p(t_e)$. If there are multiple events of type t in a world state $\mathbf{s}$, then an event of type t is selected uniformly from the set $\mathbf{s}(t)$ of events

of type t in state $\mathbf{s}$. Therefore, the probability of picking an event is given by:

$$P(\mathbf{e}|\mathbf{s}) = p(t_e)\frac{1}{|\mathbf{s}(t_e)|} \quad (1)$$

5.2 Natural language generation model

The natural-language generation model defines the probability distribution of NL sentences given an MR specified by the previously selected event. We use Lu et al. (2008)'s generative model for this step, in which:

$$P(\mathbf{w}|\mathbf{e}) = \sum_{\forall \mathcal{T}\ over\ (\mathbf{w},\mathbf{m})} P(\mathcal{T}, \mathbf{w}|\mathbf{m}) \quad (2)$$

where $\mathbf{m}$ is the MR logical form defined by event $\mathbf{e}$ and $\mathcal{T}$ is a hybrid tree defined over the NL–MR pair $(\mathbf{w}, \mathbf{m})$.

The probability $P(\mathcal{T}, \mathbf{w}|\mathbf{m})$ is calculated using the generative semantic parsing model of Lu et al. (2008) using the joint probability of the NL–MR pair $(\mathbf{w}, \mathbf{m})$, i.e. the inside probability of generating $(\mathbf{w}, \mathbf{m})$. The likelihood of a sentence $\mathbf{w}$ is then the sum over all possible hybrid trees defined by the NL–MR pair $(\mathbf{w}, \mathbf{m})$. [1]

The natural language generation model covers the roles of both the *field choice* model and *word choice* models of Liang et al. (2009). Since our event selection model only chooses an event based on its type, the order of its arguments still needs to be addressed. However, Lu et al.'s generative model includes ordering the MR arguments (as specified by MR production rules) as well as the generation of NL words and phrases to express these arguments. Thus, it is unnecessary to separately model argument ordering in our approach.[2]

[1] Lu et al. (2008) propose 3 models for generative semantic parsing: unigram, bigram, and mixgram (interpolation between the two). We used the bigram model, where the generation of a hybrid-tree component (NL word or semantic category) depends on the previously generated component as well as the parent MR production. The bigram model always performed the best on all tasks in our experimental evaluation.

[2] We also tried using a Markov model to order arguments like Liang et al. (2009), but preliminary experimental results showed that this additional component actually decreased performance rather than improving it.

6 Learning and Inference

This composite generative model is trained using conventional EM methods. The process is similar to Lu et al. (2008)'s, an inside-outside style algorithm using dynamic programming to generate a hybrid tree from the NL–MR pair $(\mathbf{w}, \mathbf{m})$, except our model's estimation process additionally deals with calculating expected counts under the posterior $P(\mathbf{e}|\mathbf{w}, \mathbf{s}; \theta)$ in the E-step and normalizing the counts to optimize parameters. The whole process is quite efficient; training time takes about 30 minutes to run on sportscasts of three games in either English or Korean.

Unfortunately, we found that EM tended to get stuck at local maxima with respect to learning the event-type selection probabilities, $p(t)$. Therefore, we also tried initializing these parameters with the corresponding strategic generation values learned by the IGSL method of Chen and Mooney (2008). Since IGSL was shown to be quite effective at predicting which event types were likely to be described, the use of IGSL priors provides a good starting point for our event selection model.

Our model is built on top of Lu et al. (2008)'s generative semantic parsing model, which is also trained in several steps in its best-performing version.[3] Thus, the overall model is vulnerable to getting stuck in local optima when running EM across these multiple steps. We also tried using random restarts with different initialization of parameters, but initializing with IGSL priors performed the best in our experimental evaluation.

7 Experimental Evaluation

We evaluated our proposed model on the Robocup sportscasting data described in Section 4. Our experimental results cover 3 tasks: NL–MR matching, semantic parsing, and tactical generation. Following Chen and Mooney (2008), the experiments were conducted using 4-fold (leave one game out) cross validation. Since the corpus contains data for four separate games, each fold uses 3 games for training and the remaining game for

[3] The bigram model of Lu et al. (2008), which is the one used in this paper, must be trained using parameters previously learned for the IBM Model 1 and unigram model in order to exhibit the best performance. We followed the same training scheme in our version.

testing for semantic parsing and tactical generation. Matching performance is measured in training data, since the goal is to disambiguate this data. All results are averaged across these 4 folds.

We also use the same performance metrics as Chen and Mooney (2008). The accuracy of matching and semantic parsing are measured using F-measure, the harmonic mean of precision and recall, where precision is the fraction of the system's annotations that are correct, and recall is the fraction of the annotations from the gold-standard that the system correctly produces. Generation is evaluated using BLEU score (Papineni et al., 2002) between generated sentences and reference NL sentences in the test set. We compare our results to previous results from Chen and Mooney (2008) and Chen et al. (2010) and to matching results on Robocup data from Liang et al. (2009).

7.1 NL–MR Matching

The goal of matching is to find the most probable NL–MR alignment for ambiguous examples consisting of an NL sentence and multiple potential MR logical forms. In Robocup sportscasting, the MRs for a given sentence correspond to all game events that occur within a 5-second window prior to the NL comment. Not all NL sentences have a matching MR in this window, but most do. During testing, an NL $\mathbf{w}$ is matched to an MR $\mathbf{m}$ if and only if the learned semantic parser produces $\mathbf{m}$ as the most probable parse of $\mathbf{w}$. Thus, our model does not force every NL to match an MR. If the most probable semantic parse of a sentence does not match *any* of the possible recent events, it is simply left unmatched. Matching is evaluated against the gold-standard matches supplied with the data, which are used for evaluation purposes only. The gold matching data is never used during training.

Table 2 shows the detailed results for both English and Korean data.[4] Our best approach outperforms all previous methods for both English and Korean by quite large margins. Note that initializing our EM training with IGSL's estimates improves performance significantly, and this approach outperforms Chen et al. (2010)'s best method, which also uses IGSL.

In particular, our proposed model outperforms the generative alignment model of Liang et al. (2009), indicating that the extra linguistic information and MR grammatical structure used by Lu et al. (2008)'s generative language model make our overall model more effective than a simple Markov + bag-of-words model for language generation.

	English	Korean
Chen and Mooney (2008)	0.681	0.753
Liang et al. (2009)	0.757	0.694
Chen et al. (2010)	0.793	0.841
Our model	0.832	0.800
Our model w/ IGSL init	**0.885**	**0.895**

Table 2: NL–MR Matching Results (F-measure). Results are the highest reported in the cited work.

	English	Korean
Chen and Mooney (2008)	0.702	0.720
Chen et al. (2010)	0.803	**0.812**
Our learned parser	0.742	0.764
Lu et al. + our matching	**0.810**	0.794
WASP + our matching	0.786	0.808
Lu et al. + Liang et al.	0.790	0.690
WASP + Liang et al.	0.803	0.740

Table 3: Semantic Parsing Results (F-measure). Results are the highest reported in the cited work.

7.2 Semantic Parsing

Semantic parsing is evaluated by determining how accurately NL sentences in the test set are correctly mapped to their meaning representations. Results are presented in Table 3.[5] [6] For our model, we report results using the parser learned directly from the ambiguous supervision, as well

[4] Since the Korean data was not yet available for use by either Chen and Mooney (2008) or Liang et al. (2009), we present the results reported by Chen et al. (2010) for these methods.

[5] The best result of Chen and Mooney (2008) is for WASPER-GEN, and that of Chen et al. (2010) is for WASPER with Liang et al.'s matching initialization for English and for WASER-GEN-IGSL-METEOR with Liang et al.'s initialization for Korean.

[6] Our semantic parsing results are based on our best matching results with IGSL initialization.

as results for training a supervised parser (both WASP and Lu et al. (2009)'s) on the NL–MR matching produced by our model. We also present results for training Lu et al.'s parser and WASP on Liang et al.'s NL–MR matchings.

Our initial learned semantic parser does not perform better than the best results reported by Chen et al. (2010), but it is clearly better than the initial results of Chen and Mooney (2008). Training WASP and Lu et al.'s supervised parser on our method's highly accurate set of disambiguated NL–MR pairs improved the results. Retraining Lu et al.'s parser gave the best overall results for English, and retraining WASP gave the second highest results for Korean, only failing to beat the very best results of Chen et al. (2010). It is somewhat surprising that simply retraining on the hardened set of most probable NL–MR matches gives better results than the parser trained using EM, which actually exploits the uncertainty in the underlying matches. Further investigations of this phenomenon are indicated.

Comparing with the corresponding results for training WASP and Lu et al.'s supervised parser on the NL–MR matchings produced by Liang et al.'s alignment method, it is clear that our matchings produce more accurate semantic parsers except when training WASP on English.

7.3 Tactical Generation

Tactical generation is evaluated based on how well the learned model generates accurate NL sentences from MR logical forms. Without integrating a language model for the NL, the existing generative model is not very effective for tactical generation. Lu et al. (2009) introduced an effective language generator for the hybrid tree framework using a Tree-CRF model; however, we did not have access to this system. Therefore, for tactical generation, we used the publicly available WASP^{-1} system (Wong and Mooney, 2007a) trained on disambiguated NL–MR matches. This approach also allows direct comparison with the results of Chen and Mooney (2008) and Chen et al. (2010), who also used WASP^{-1} for tactical generation. Our objective is to show that the more accurate matchings produced by our generative model can improve tactical generation.

	English	Korean
Chen and Mooney (2008)	0.4560	0.5575
Chen et al. (2010)	0.4599	0.6796
WASP^{-1} + Liang et al.	0.4580	0.5828
WASP^{-1} + our matching	**0.4727**	**0.7148**

Table 4: Tactical Generation Results (BLEU score). Results are the highest reported in the cited work.

The results are shown in Table 4.[7] [8] Overall, WASP^{-1} trained on the NL–MR matching from our alignment model performs better than all previous methods. In particular, using the matchings from our method to train WASP^{-1} produces better tactical generators than using matchings from Liang et al.'s approach.

7.4 Discussion

Overall, our model performs particularly well at matching NL and MRs under ambiguous supervision, and the difference is larger for English than Korean. However, improved matching results do not necessarily translate into significantly better semantic parsers. For English, the improvement in matching is almost 10 percentage points in F-measure, but the semantic parsing result trained with this more accurate matching shows only 1 point improvement.

Compared to Liang et al. (2009), our more accurate (i.e. higher F-measure) matchings provide a clear improvement in both semantic parsing and tactical generation. The only exception is English parsing using WASP, which seems to be due to some misleading noise in our alignments. WASP seems to be affected more than Lu et al.'s system by such extraneous noise. However, in tactical generation, this extraneous noise does not seem to lead to worse performance, and our approach always gives the best results. As discussed by Chen and Mooney (2008) and Chen et al. (2010), tactical generation is somewhat easier than semantic parsing in that semantic parsing needs to learn

[7] The best result of Chen and Mooney (2008) is for WASPER-GEN, and that of Chen et al. (2010) is for WASPER with Liang et al.'s matching initialization for English and for WASER-GEN with Liang et al. initialization for Korean.

[8] Our generation results are based on our best matching results with IGSL initialization.

to map a variety of synonymous natural-language expressions to the same meaning representation, while tactical generation only needs to learn one way to produce a correct natural language description of an event. This difference in the nature of semantic parsing and tactical generation may be the cause of the different trends in the results.

8 Conclusions and Future Work

We have presented a novel generative model capable of probabilistically aligning natural-language sentences to their correct meaning representations given the ambiguous supervision provided by a grounded language acquisition scenario. Our model is also capable of simultaneously learning to semantically parse NL sentences into their corresponding meaning representations. Experimental results in Robocup sportscasting show that the NL–MR matchings inferred by our model are significantly more accurate than those produced by all previous methods. Our approach also learns competitive semantic parsers and improved language generators compared to previous methods. In particular, we showed that our alignments provide a better foundation for learning accurate semantic parsers and tactical generators compared to those of Liang et al. (2009), whose generative model is limited by a simple bag-of-words assumption.

In the future, we plan to test our model on more complicated data with higher degrees of ambiguity as well as more complex meaning representations. One immediate direction is evaluating our approach on the datasets of weather forecasts and NFL football articles used by Liang et al. (2009). However, our current model does not support matching multiple meaning representations to the same natural-language sentence, and needs to be extended to allow multiple MRs to generate a single NL sentence.

Acknowledgements

We thank Wei Lu and Wee Sun Lee for sharing their software and giving helpful comments for the paper. We also thank Percy Liang for sharing his code and experimental results with us. Additionally, we thank David Chen in UTCS ML group for his comments and advice. Finally, we thank the anonymous reviewers for their comments. This work was funded by the NSF grant IIS. 0712907X. The experiments were executed and run on the Mastodon Cluster, provided by NSF Grant EIA-0303609.

References

Chen, David L. and Raymond J. Mooney. 2008. Learning to sportscast: a test of grounded language acquisition. In *ICML '08: Proceedings of the 25th International Conference on Machine Learning*, pages 128–135, New York, NY, USA. ACM.

Chen, David L., Joohyun Kim, and Raymond J. Mooney. 2010. Training a multilingual sportscaster: Using perceptual context to learn language. *Journal of Artificial Intelligence Research*, 37:397–435.

Ge, Ruifang and Raymond J. Mooney. 2005. A statistical semantic parser that integrates syntax and semantics. In *Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL-2005)*, pages 9–16, Ann Arbor, MI, July.

Kate, Rohit J. and Raymond J. Mooney. 2006. Using string-kernels for learning semantic parsers. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (COLING/ACL-06)*, pages 913–920, Morristown, NJ, USA. Association for Computational Linguistics.

Kate, Rohit J. and Raymond J. Mooney. 2007. Learning language semantics from ambiguous supervision. In *Proceedings of the Twenty-Second Conference on Artificial Intelligence (AAAI-07)*, pages 895–900, Vancouver, Canada, July.

Liang, Percy, Michael I. Jordan, and Dan Klein. 2009. Learning semantic correspondences with less supervision. In *ACL-IJCNLP '09: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1*, pages 91–99, Morristown, NJ, USA. Association for Computational Linguistics.

Lu, Wei, Hwee Tou Ng, Wee Sun Lee, and Luke S. Zettlemoyer. 2008. A generative model for parsing natural language to meaning representations. In *EMNLP '08: Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 783–792, Morristown, NJ, USA. Association for Computational Linguistics.

Lu, Wei, Hwee Tou Ng, and Wee Sun Lee. 2009. Natural language generation with tree conditional random fields. In *EMNLP '09: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 400–409, Morristown, NJ, USA. Association for Computational Linguistics.

Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL-2002)*, pages 311–318, Philadelphia, PA, July.

Wong, Yuk Wah and Raymond J. Mooney. 2007a. Generation by inverting a semantic parser that uses statistical machine translation. In *Proceedings of Human Language Technologies: The Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT-07)*, pages 172–179, Rochester, NY.

Wong, Yuk Wah and Raymond J. Mooney. 2007b. Learning synchronous grammars for semantic parsing with lambda calculus. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL-07)*, pages 960–967, Prague, Czech Republic, June.

Zelle, John M. and Raymond J. Mooney. 1996. Learning to parse database queries using inductive logic programming. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI-96)*, pages 1050–1055, Portland, OR, August.

Zettlemoyer, Luke S. and Michael Collins. 2005. Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. In *Proceedings of 21st Conference on Uncertainty in Artificial Intelligence (UAI-2005)*, Edinburgh, Scotland, July.

Zettlemoyer, Luke S. and Michael Collins. 2007. Online learning of relaxed CCG grammars for parsing to logical form. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL-07)*, pages 678–687, Prague, Czech Republic, June.

Local Space-Time Smoothing for Version Controlled Documents

Seungyeon Kim
Georgia Institute of Technology

Guy Lebanon
Georgia Institute of Technology

Abstract

Unlike static documents, version controlled documents are continuously edited by one or more authors. Such collaborative revision process makes traditional modeling and visualization techniques inappropriate. In this paper we propose a new representation based on local space-time smoothing that captures important revision patterns. We demonstrate the applicability of our framework using experiments on synthetic and real-world data.

1 Introduction

Most computational linguistics studies concentrate on modeling or analyzing documents as sequences of words. In this paper we consider modeling and visualizing version controlled documents which is the authoring process leading to the final word sequence. In particular, we focus on documents whose authoring process naturally segments into consecutive versions. The revisions, as the differences between consecutive versions are often called, may be authored by a single author or by multiple authors working collaboratively.

One popular way to keep track of version controlled documents is using a version control system such as CVS or Subversion (SVN). This is often the case with books or with large computer code projects. In other cases, more specialized computational infrastructure may be available, as is the case with the authoring API of Wikipedia.org, Slashdot.com, and Google Wave. Accessing such API provides information about what each revision contains, when was it submitted, and who edited it. In any case, we formally consider a version controlled document as a sequence of documents $d_1, \ldots, d_l$ indexed by their revision number where d_i typically contains some locally concentrated additions or deletions, as compared to d_{i-1}.

In this paper we develop a continuous representation of version controlled documents that generalizes the locally weighted bag of words representation (Lebanon et al., 2007). The representation smooths the sequence of version controlled documents across two axes-time t and space s. The time axis t represents the revision and the space axis s represents document position. The smoothing results in a continuous map from a space-time domain to the simplex of term frequency vectors

$$\gamma : \Omega \to \mathbb{P}_V \quad \text{where} \quad \Omega \subset \mathbb{R}^2, \quad \text{and} \quad (1)$$

$$\mathbb{P}_V = \left\{ w \in \mathbb{R}^{|V|} : w_i \geq 0, \ \sum_{i=1}^{|V|} w_i = 1 \right\}.$$

The mapping above (V is the vocabulary) captures the variation in the local distribution of word content across time and space. Thus $[\gamma(s,t)]_w$ is the (smoothed) probability of observing word w in space s (document position) and time t (version). Geometrically, γ realizes a divergence-free vector field (since $\sum_w [\gamma(s,t)]_w = 1$, γ has zero divergence) over the space-time domain Ω.

We consider the following four version controlled document analysis tasks. The first task is visualizing word-content changes with respect to space (how quickly the document changes its content), time (how much does the current version differs from the previous one), or mixed space-time. The second task is detecting sharp transitions or edges in word content. The third task is concerned with segmenting the space-time domain into a finite partition reflecting word content. The fourth task is predicting future revisions. Our main tool in addressing tasks 1-4 above is to analyze the values of the vector field γ and its first

Coling 2008: Poster Volume, pages 552–560,
Beijing, August 2010

order derivatives fields

$$\nabla\gamma = (\dot{\gamma}_s, \dot{\gamma}_t)\,. \tag{2}$$

2 Space-Time Smoothing for Version Controlled Documents

With no loss of generality we identify the vocabulary V with positive integers $\{1,\ldots,V\}$ and represent a word $w \in V$ by a unit vector[1] (all zero except for 1 at the w-component)

$$e(w) = (0,\ldots,0,1,0,\ldots,0)^\top \quad w \in V. \tag{3}$$

We extend this definition to word sequences thus representing documents $\langle w_1,\ldots,w_N\rangle$ ($w_i \in V$) as sequences of V-dimensional vectors $\langle e(w_1),\ldots,e(w_N)\rangle$. Similarly, a version controlled document is sequence of documents $d^{(1)},\ldots,d^{(l)}$ of potentially different lengths $d^{(j)} = \langle w_1^{(j)},\ldots,w_{N(j)}^{(j)}\rangle$. Using (3) we represent a version controlled document as the array

$$\begin{array}{ccc} e(w_1^{(1)}), & \ldots, & e(w_{N(1)}^{(1)}) \\ \vdots & \ddots & \vdots \\ e(w_1^{(l)}), & \ldots, & e(w_{N(l)}^{(l)}) \end{array} \tag{4}$$

where columns and rows correspond to space (document position) and time (versions).

The array (4) of high dimensional vectors represents the version controlled document without any loss of information. Nevertheless the high dimensionality of V suggests we smooth the vectors in (4) with neighboring vectors in order to better capture the local word content. Specifically we convolve each component of (4) with a 2-D smoothing kernel K_h to obtain a smooth vector field γ over space-time (Wand and Jones, 1995) e.g.,

$$\gamma(s,t) = \sum_{s'}\sum_{t'} K_h(s-s',t-t')e(w_{s'}^{(t')})$$
$$K_h(x,y) \propto \exp\left(-(x^2+y^2)/(2h^2)\right). \tag{5}$$

Thus as (s,t) vary over a continuous domain $\Omega \subset \mathbb{R}^2$, $\gamma(s,t)$, which is a weighted combination of neighboring unit vectors, traces a continuous surface in $\mathbb{P}_V \subset \mathbb{R}^V$. Assuming that the kernel K_h is a normalized density it can be shown that $\gamma(s,t)$ is a non-negative normalized vector i.e., $\gamma(s,t) \in \mathbb{P}_V$ (see (1) for a definition of $\mathbb{P}_V$) measuring the local distribution of words around the space-time location (s,t). It thus extends the concept of lowbow (locally weighted bag of words) introduced in (Lebanon et al., 2007) from single documents to version controlled documents.

One difficulty with the above scheme is that the document versions $d_1,\ldots,d_l$ may be of different lengths. We consider two ways to resolve this issue. The first pads shorter document versions with zero vectors as needed. We refer to the resulting representation γ as the non-normalized representation. The second approach normalizes all document versions to a common length, say $\prod_{j=1}^{l} N(j)$. That is each word in the first document is expanded into $\prod_{j\neq 1} N(j)$ words, each word in the second document is expanded into $\prod_{j\neq 2} N(j)$ words etc. We refer to the resulting representation γ as the normalized representation.

The non-normalized representation has the advantage of conveying absolute lengths. For example, it makes it possible to track how different portions of the document grow or shrink (in terms of number of words) with the version number. The normalized representation has the advantage of conveying lengths relative to the document length. For example, it makes it possible to track how different portions of the document grow or shrink with the version number relative to the total document length. In either case, the space-time domain Ω on which γ is defined (5) is a two dimensional rectangular domain $\Omega = [0,I]\times[0,J]$.

Before proceeding to examine how γ may be used in the four tasks described in Section 1 we demonstrate our framework with a simple low dimensional example. Assuming a vocabulary of two words $V = \{1,2\}$ we can visualize γ by displaying its first component as a grayscale image (since $[\gamma(s,t)]_2 = 1 - [\gamma(s,t)]_1$ the second component is redundant). Specifically, we created a version controlled document with three contiguous segments whose $\{1,2\}$ words were sampled from Bernoulli distributions with parameters 0.3 (first segment), 0.7 (second segment), and 0.5 (third segment). That is, the probability of getting 1 is highest for the second segment, equal for the third and lowest for the first segment. The initial lengths of the segments were

[1] Note the slight abuse of notation as V represents both a set of words and an integer $V = \{1,\ldots,V\}$ with $V = |V|$.

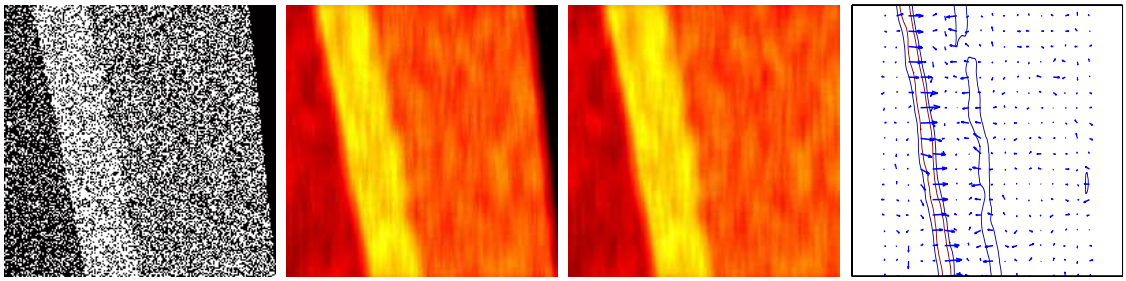


Figure 1: Four space-time representations of a simple synthetic version controlled document over $V = \{1, 2\}$ (see text for more details). The left panel displays the first component of (4) (non-smoothed array of unit vectors corresponding to words). The second and third panels display $[\gamma(s,t)]_1$ for the non-normalized and normalized representations respectively. The fourth panel displays the gradient vector field $(\dot{\gamma}_s(s,t), \dot{\gamma}_t(s,t))$ (contour levels represent the gradient magnitude). The black portions of the first two panels correspond to zero padding due to unequal lengths of the different versions.

30, 40 and 120 words with the first segment increasing and the third segment decreasing at half the rate of the first segment with each revision. The length of the second segment was constant across the different versions. Figure 1 displays the nonsmoothed ragged array (4) (left), the non-normalized $[\gamma(s,t)]_1$ (middle left) and the normalized $[\gamma(s,t)]_1$ (middle right).

While the left panel doesn't distinguish much between the second and third segment the two smoothed representations display a nice segmentation of the space-time domain into three segments, each with roughly uniform values. The non-normalized representation (middle left) makes it easy to see that the total length of the version controlled document is increasing but it is not easy to judge what happens to the relative sizes of the three segments. The normalized representation (middle right) makes it easy to see that the first segment increases in size, the second is constant, and the third decreases in size. It is also possible to notice that the growth rate of the first segment is higher than the decay rate of the third.

3 Visualizing Change in Space-Time

We apply the space-time representation to four tasks. The first task, visualizing change, is described in this section. The remaining three tasks are described in the next three section.

The space-time domain Ω represents the union of all document versions and all document positions. Some parts of Ω are more homogeneous and some are less in terms of their local word distribution. Locations in Ω where the local word distribution substantially diverges from its neighbors correspond to sharp content transitions. On the other hand, locations whose word distribution is more or less constant correspond to slow content variation.

We distinguish between three different types of changes. The first occurs when the word content changes substantially between neighboring document positions within a certain document version. As an example consider a document location whose content shifts from high level introductory motivation to a detailed technical description. Such change is represented by

$$\|\dot{\gamma}_s(s,t)\|^2 = \sum_{w=1}^{V} \left(\frac{\partial [\gamma(s,t)]_w}{\partial s} \right)^2 . \qquad (6)$$

A second type of change occurs when a certain document position undergoes substantial change in local word distribution across neighboring versions. An example is erroneous content in one version being heavily revised in the next version. Such change along the time axis corresponds to the magnitude of

$$\|\dot{\gamma}_t(s,t)\|^2 = \sum_{w=1}^{V} \left(\frac{\partial [\gamma(s,t)]_w}{\partial t} \right)^2 . \qquad (7)$$

Expression (6) may be used to measure the instantaneous rate of change in the local word distribution. Alternatively, integrating (6) provides a global measure of change

$$h(s) = \int \|\dot{\gamma}_s(s,t)\|^2 \, dt, \quad g(t) = \int \|\dot{\gamma}_t(s,t)\|^2 \, ds$$

with $h(s)$ describing the total amount of spatial change across all revisions and $g(t)$ describing

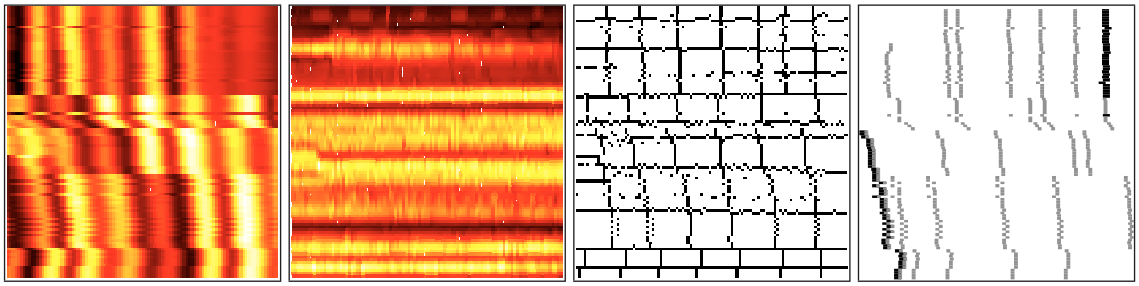


Figure 2: Gradient and edges for a portion of the version controlled Wikipedia Religion article. The left panel displays $\|\dot{\gamma}_s(s,t)\|^2$ (amount of change across document locations for different versions). The second panel displays $\|\dot{\gamma}_t(s,t)\|^2$ (amount of change across versions for different document positions). The third panel displays the local maxima of $\|\dot{\gamma}_s(s,t)\|^2 + \|\dot{\gamma}_t(s,t)\|^2$ which correspond to potential edges, either vertical lines (section and subsection boundaries) or horizontal lines (between substantial revisions). The fourth panel displays boundaries of sections and subsections as black and gray lines respectively.

the total amount of version change across different document positions. $h(s)$ may be used to detect document regions undergoing repeated substantial content revisions and $g(t)$ may be used to detect revisions in which substantial content has been modified across the entire document.

We conclude with the integrated directional derivative

$$\int_0^1 \|\dot{\alpha}_s(r)\dot{\gamma}_s(\alpha(r)) + \dot{\alpha}_t(r)\dot{\gamma}_t(\alpha(r))\|^2 \, dr \quad (8)$$

where $\alpha : [0,1] \to \Omega$ is a parameterized curve in the space-time and $\dot{\alpha}$ its tangent vector. Expression (8) may be used to measure change along a dynamically moving document anchor such as the boundary between two book chapters. The space coordinate of such anchor shifts with the version number (due to the addition and removal of content across versions) and so integrating the gradient across one of the two axis as in (7) is not appropriate. Defining $\alpha(r)$ to be a parameterized curve in space-time realizing the anchor positions $(s,t) \in \Omega$ across multiple revisions, (8) measures the amount of change at the anchor point.

3.1 Experiments

The right panel of Figure 1 shows the gradient vector field corresponding to the synthetic version controlled document described in the previous section. As expected, it tends to be orthogonal to the segment boundaries. Its magnitude is displayed by the contour lines which show highest magnitudes around segment boundaries.

Figure 2 shows the norm $\|\dot{\gamma}_s(s,t)\|^2$ (left), $\|\dot{\gamma}_t(s,t)\|^2$ (middle left) and the local maxima of $\|\dot{\gamma}_s(s,t)\|^2 + \|\dot{\gamma}_t(s,t)\|^2$ (middle right) for a portion of the version controlled Wikipedia Religion article. The first panel shows the amount of change in local word distribution within documents. High values correspond to boundaries between sections, topics or other document segments. The second panel shows the amount of change as one version is replaced with another. It shows which revisions change the word distributions substantially and which result in a relatively minor change. The third panel shows only the local maxima which correspond to edges between topics or segments (vertical lines) or revisions (horizontal lines).

4 Edge Detection

In many cases documents may be divided to semantically coherent segments. Examples of text segments include individual news stories in streaming broadcast news transcription, sections in article or books, and individual messages in a discussion board or an email trail. For non-version controlled documents finding the text segments is equivalent to finding the boundaries or edges between consecutive segments. See (Hearst, 1997; Beeferman et al., 1999; McCallum et al., 2000) for several recent studies in this area.

Things get a bit more complicated in the case of version controlled documents. Segments, and their boundaries exist in each version. As in case of image processing, we may view segment boundaries as edges in the space-time domain Ω. These boundaries separate the segments from each other, much like borders separate countries

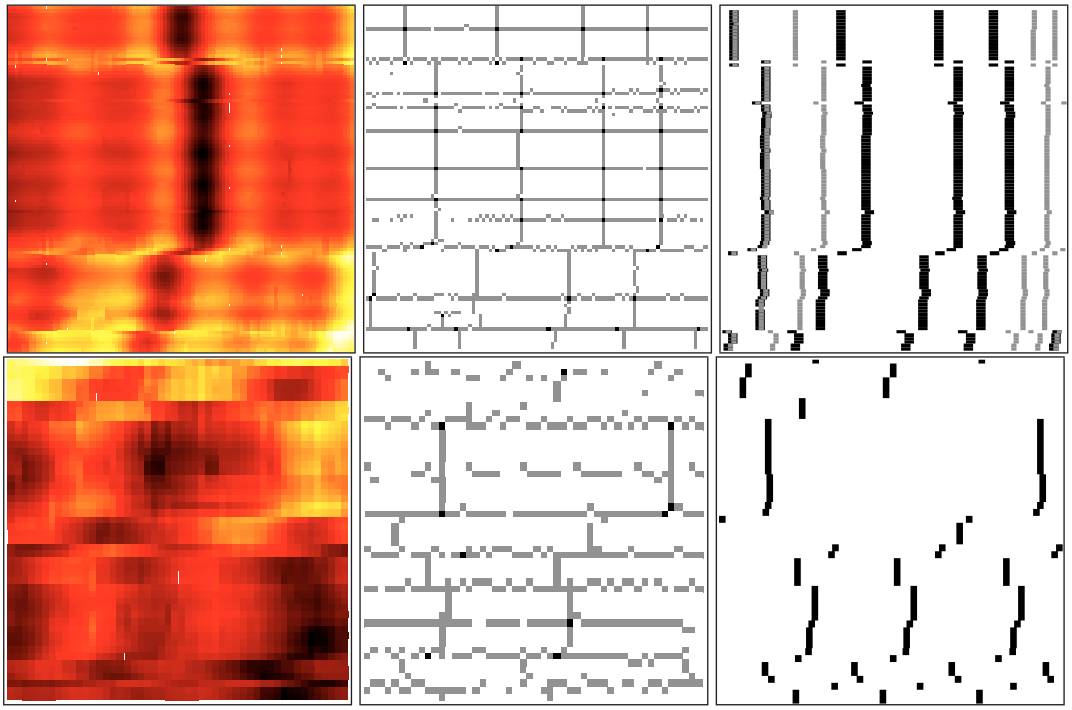


Figure 3: Gradient and edges of a portion of the version controlled Atlanta Wikipedia article (top row) and the Google Wave Amazon Kindle FAQ (bottom row). The left column displays the magnitude of the gradient in both space and time $\|\dot{\gamma}_s(s,t)\|^2 + \|\dot{\gamma}_t(s,t)\|$. The middle column displays the local maxima of the gradient magnitude (left column). The right column displays the actual segment boundaries as vertical lines (section headings for Wikipedia and author change in Google Wave). The gradient maxima corresponding to vertical lines in the middle column matches nicely the Wikipedia section boundaries. The gradient maxima corresponding to horizontal lines in the middle column correspond nicely to major revisions indicated by a discontinuities in the location of the section boundaries.

in a two dimensional geographical map.

Assuming all edges are correctly identified, we can easily identify the segments as the interior points of the closed boundaries. In general, however, attempts to identify segment boundaries or edges will only be partially successful. As a result predicted edges in practice are not closed and do not lead to interior segments. We consider now the task of predicting segment boundaries or edges in Ω and postpone the task of predicting a segmentation to the next section.

Edges, or transitions between segments, correspond to abrupt changes in the local word distribution. We thus characterize them as points in Ω having high gradient value. In particular, we distinguish between vertical edges (transitions across document positions), horizontal edges (transitions across versions), and diagonal edges (transitions across both document position and version). These three types of edges may be diagnosed based on the magnitudes of $\dot{\gamma}_s$, $\dot{\gamma}_t$, and $\dot{\alpha}_1\gamma_s + \dot{\alpha}_2\gamma_t$ respectively.

4.1 Experiments

Besides the synthetic data results in Figure 2, we conducted edge detection experiments on six different real world datasets. Five datasets are Wikipedia.com articles: Atlanta, Religion, Language, European Union, and Beijing. Religion and European Union are version controlled documents with relatively frequent updates, while Atlanta, language, and Beijing have less frequent changes. The sixth dataset is the Google Wave Amazon Kindle FAQ which is a less structured version controlled document.

Preprocessing included removing html tags and pictures, word stemming, stop-word removal, and removing any non alphabetic characters (numbers and punctuations). The section heading information of Wikipedia and the information of author of each posting in Google Wave is used as ground truth for segment boundaries. This information was separated from the dataset and was used for training and evaluation (on testing set).

Figure 3 displays a gradient information, local maxima, and ground truth segment boundaries for

Article	Rev.	Voc. Size	$p(y)$	Error Rate			F1 Measure		
				a	b	c	a	b	c
Atlanta	2000	3078	0.401	0.401	0.424	0.339	0.000	0.467	0.504
Religion	2000	2880	0.403	0.404	0.432	0.357	0.000	0.470	0.552
Language	2000	3727	0.292	0.292	0.450	0.298	0.000	0.379	0.091
European Union	2000	2382	0.534	0.467	0.544	0.435	0.696	0.397	0.663
Beijing	2000	3857	0.543	0.456	0.474	0.391	0.704	0.512	0.682
Amazon Kindle FAQ	100	573	0.339	0.338	0.522	0.313	0.000	0.436	0.558

Figure 4: Test set error rate and F1 measure for edge prediction (section boundaries in Wikipedia articles and author change in Google Wave). The space-time domain Ω was divided to a grid with each cell labeled edge ($y = 1$) or no edge ($y = 0$) depending on whether it contained any edges. Method a corresponds to a predictor that always selects the majority class. Method b corresponds to the TextTiling test segmentation algorithm (Hearst, 1997) without paragraph boundaries information. Method c corresponds to a logistic regression classifier whose feature set is composed of statistical summaries (mean, median, max, min) of $\dot{\gamma}_s(s,t)$ within the grid cell in question as well as neighboring cells.

the version controlled Wikipedia articles Religion and Atlanta. The local gradient maxima nicely match the segment boundaries which lead us to consider training a logistic regression classifier on a feature set composed of gradient value statistics (min, max, mean, median of $\|\dot{\gamma}_s(s,t)\|$ in the appropriate location as well as its neighbors (the space-time domain Ω was divided into a finite grid where each cell either contained an edge ($y = 1$) or did not ($y = 0$)). The table in Figure 4 displays the test set accuracy and F1 measure of three predictors: our logistic regression (method c) as well as two baselines: predicting edge/no-edge based on the marginal $p(y)$ distribution (method a) and TextTiling (method b) (Hearst, 1997) which is a popular text segmentation algorithm. Since we do not assume paragraph information in our experiment we ignored this component and considered the document as a sequence with $w = 20$ and 29 minimum depth gaps parameters (see (Hearst, 1997)). We conclude from the figure that the gradient information leads to better prediction than TextTiling (on both accuracy and F1 measure).

5 Segmentation

As mentioned in the previous section, predicting edges may not result in closed boundaries. It is possible to analyze the location and direction of the predicted edges and aggregate them into a sequence of closed boundaries surrounding the segments. We take a different approach and partition points in Ω to k distinct values or segments based on local word content and space-time proximity.

For two points $(s_1, t_2), (s_2, t_2) \in \Omega$ to be in the same segment we expect $\gamma(s_1, t_1)$ to be similar to $\gamma(s_2, t_2)$ and for (s_1, t_1) to be close to (s_2, t_2). The first condition asserts that the two locations discuss the same topic. The second condition asserts that the two locations are not too far from each other in the space time domain. More specifically, we propose to segment Ω by clustering its points based on the following geometry

$$d((s_1,t_1),(s_2,t_2)) = d_H(\gamma(s_1,t_1),\gamma(s_2,t_2)) + \sqrt{c_1(s_1-s_2)^2 + c_2(t_1-t_2)^2} \quad (9)$$

where $d_H : \mathbb{P}_V \times \mathbb{P}_V \to \mathbb{R}$ is Hellinger distance

$$d_H^2(u,v) = \sum_{i=1}^{V} (\sqrt{u_i} - \sqrt{v_i})^2. \quad (10)$$

The weights c_1, c_2 are used to balance the contributions of word content similarity with the similarity in time and space.

5.1 Experiments

Figure 5 displays the ground truth segment boundaries and the segmentation results obtained by applying k-means clustering ($k = 11$) to the metric (9). The figure shows that the predicted segments largely match actual edges in the documents even though no edge or gradient information was used in the segmentation process.

6 Predicting Future Operations

The fourth and final task is predicting a future revision d_{l+1} based on the smoothed representation of the present and past versions $d_1, \ldots, d_l$. In

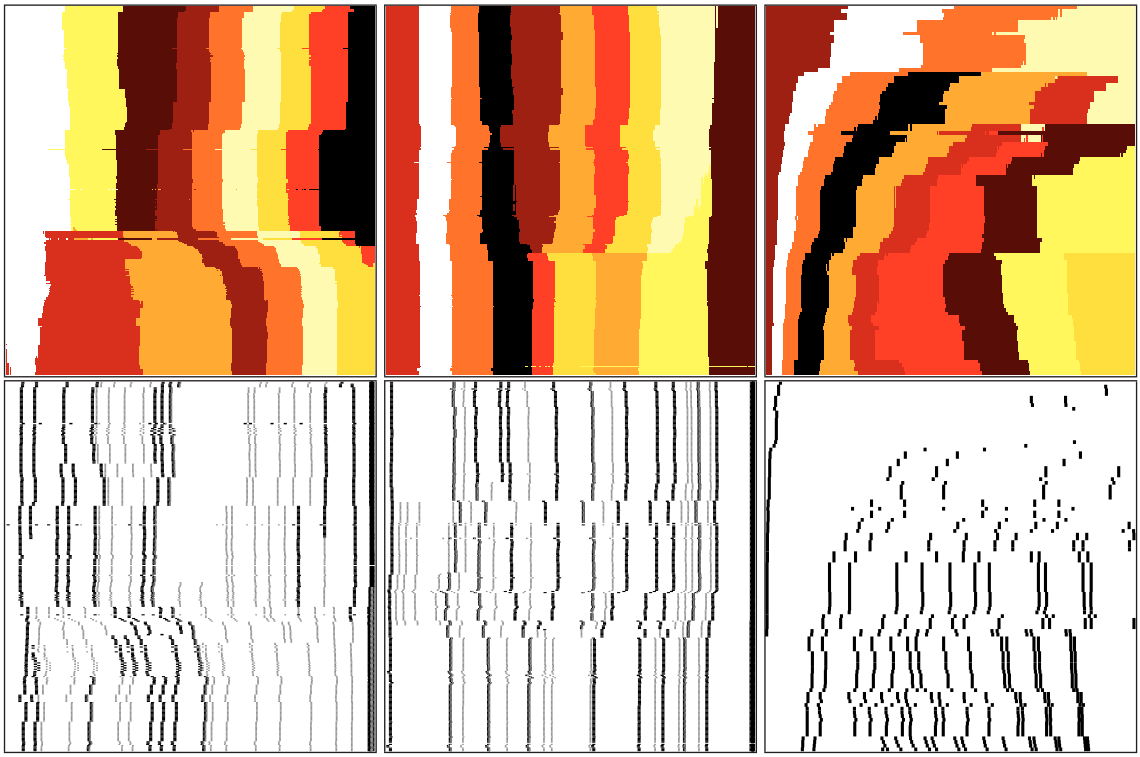


Figure 5: Predicted segmentation (top) and ground truth segment boundaries (bottom) of portions of the version controlled Wikipedia articles Religion (left), Atlanta (middle) and the Google Wave Amazon Kindle FAQ(right). The predicted segments match the ground truth segment boundaries. Note that the first 100 revisions are used in Google Wave result. The proportion of the segments that appeared in the beginning is keep decreasing while the revisions increases and new segments appears.

terms of Ω, this means predicting features associated with $\gamma(s,t), t \geq t'$ based on $\gamma(s,t), t < t'$.

6.1 Experiments

We concentrate on predicting whether Wikipedia edits are reversed in the next revision. This action, marked by a label UNDO or REVERT in the Wikipedia API, is important for preventing content abuse or removing immature content (by predicting ahead of time suspicious revisions).

We predict whether a version will undergo UNDO in the next version using a support vector machine based on statistical summaries (mean, median, min, max) of the following feature set $\|\dot{\gamma}_s(s,t)\|$, $\|\ddot{\gamma}_s(s,t)\|$, $\|\dot{\gamma}_t(s,t)\|$), $\|\dot{\gamma}_t(s,t)\|$, $g(h)$, and $h(s)$. Figure 6 shows the test set error and F1 measure for the logistic regression based on the smoothed space-time representation (method c), as well as two baselines. The first baseline (method a) predicts the majority class and the second baseline (method b) is a logistic regression based on the term frequency content of the current test version. Using the derivatives of γ, we obtain a prediction that is better than choosing majority class or logistic regression based on word content. We thus conclude that the derivatives above provide more useful information (resulting in lower error and higher F1) for predicting future operations than word content features.

7 Related Work

While document analysis is a very active research area, there has been relatively little work on examining version controlled documents. Our approach is the first to consider version controlled documents as continuous mappings from a space-time domain to the space of local word distributions. It extends the ideas in (Lebanon et al., 2007) of using kernel smoothing to create a continuous representation of documents. In fact, our framework generalizes (Lebanon et al., 2007) as it reverts to it in the case of a single revision.

Other approaches to sequential analysis of documents concentrate on discrete spaces and discrete models, with the possible extension of (Wang et al., 2009). Related papers on segmentation and sequential document analysis are (Hearst,

Article	Rev.	Voc. Size	$p(y)$	Error Rate			F1 Measure		
				a	b	c	a	b	c
Atlanta	2000	3078	0.218	0.219	0.313	0.212	0.000	0.320	0.477
Religion	2000	2880	0.123	0.122	0.223	0.125	0.000	0.294	0.281
Language	2000	3727	0.189	0.189	0.259	0.187	0.000	0.334	0.455
European Union	2000	2382	0.213	0.208	0.331	0.209	0.000	0.275	0.410
Beijing	2000	3857	0.137	0.137	0.219	0.136	0.000	0.247	0.284

Figure 6: Error rate and F1 measure over held out test set of predicting future UNDO operation in Wikipedia articles. Method a corresponds to a predictor that always selects the majority class. Method b corresponds to a logistic regression based on the term frequency vector of the current version. Method c corresponds a logistic regression that uses summaries (mean, median, max, min) of $\|\dot{\gamma}_s(s,t)\|$, $\|\dot{\gamma}_s(s,t)\|$, $g(t)$, and $h(s)$.

1997; Beeferman et al., 1999; McCallum et al., 2000) with (Hearst, 1997) being the closest in spirit to our approach. An influential model for topic modeling within and across documents is latent Dirichlet allocation (Blei et al., 2003; Blei and Lafferty, 2006). Our approach differs in being fully non-parametric and in that it does not require iterative parametric estimation or integration. The interpretation of local word smoothing as a non-parametric statistical estimator (Lebanon et al., 2007) may be extended to our paper in a straightforward manner.

Several attempts have been made to visualize themes and topics in documents, either by keeping track of the word distribution or by dimensionality reduction techniques e.g., (Fortuna et al., 2005; Havre et al., 2002; Spoerri, 1993; Thomas and Cook, 2005). Such studies tend to visualize a corpus of unrelated documents as opposed to ordered collections of revisions which we explore.

8 Summary and Discussion

The task of analyzing and visualizing version controlled document is an important one. It allows external control and monitoring of collaboratively authored resources such as Wikipedia, Google Wave, and CVS or SVN documents. Our framework is the first to develop analysis and visualization tools in this setting. It presents a new representation for version controlled documents that uses local smoothing to map a space-time domain Ω to the simplex of tf vectors $\mathbb{P}_V$. We demonstrate the applicability of the representation for four tasks: visualizing change, predicting edges, segmentation, and predicting future revision operations.

Visualizing changes may highlight significant structural changes for the benefit of users and help the collaborative authoring process. Improved edge prediction and text segmentation may assist in discovering structural or semantic changes and their evolution with the authoring process. Predicting future operation may assist authors as well as prevent abuse in coauthoring projects such as Wikipedia.

The experiments described in this paper were conducted on synthetic, Wikipedia and Google Wave articles. They show that the proposed formalism achieves good performance both qualitatively and quantitatively as compared to standard baseline algorithms.

It is intriguing to consider the similarity between our representation and image processing. Predicting segment boundaries are similar to edge detection in images. Segmenting version controlled documents may be reduced to image segmentation. Predicting future operations is similar to completing image parts based on the remaining pixels and a statistical model. Due to its long and successful history, image processing is a good candidate for providing useful tools for version controlled document analysis. Our framework facilitates this analogy and we believe is likely to result in novel models and analysis tools inspired by current image processing paradigms. A few potential examples are wavelet filtering, image compression, and statistical models such as Markov random fields.

Acknowledgements

The research described in this paper was funded in part by NSF grant IIS-0746853.

References

Beeferman, D., A. Berger, and J. D. Lafferty. 1999. Statistical models for text segmentation. *Machine Learning*, 34(1-3):177–210.

Blei, D. and J. Lafferty. 2006. Dynamic topic models. In *Proc. of the International Conference on Machine Learning*.

Blei, D., A. Ng, , and M. Jordan. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.

Fortuna, B., M. Grobelnik, and D. Mladenic. 2005. Visualization of text document corpus. *Informatica*, 29:497–502.

Havre, S., E. Hetzler, P. Whitney, and L. Nowell. 2002. Themeriver: Visualizing thematic changes in large document collections. *IEEE Transactions on Visualization and Computer Graphics*, 8(1).

Hearst, M. A. 1997. Texttiling: Segmenting text into multi-paragraph subtopic passages. *Computational Linguistics*, 23(1):33–64.

Lebanon, G., Y. Mao, and J. Dillon. 2007. The locally weighted bag of words framework for documents. *Journal of Machine Learning Research*, 8:2405–2441, October.

McCallum, A., D. Freitag, and F. Pereira. 2000. Maximum entropy Markov models for information extraction and segmentation. In *Proc. of the International Conference on Machine Learning*.

Spoerri, A. 1993. InfoCrystal: A visual tool for information retrieval. In *Proc. of IEEE Visualization*.

Thomas, J. J. and K. A. Cook, editors. 2005. *Illuminating the Path: The Research and Development Agenda for Visual Analytics*. IEEE Computer Society.

Wand, M. P. and M. C. Jones. 1995. *Kernel Smoothing*. Chapman and Hall/CRC.

Wang, C., D. Blei, and D. Heckerman. 2009. Continuous time dynamic topic models. In *Proc. of Uncertainty in Artificial Intelligence*.

A Logistic Regression Model of Determiner Omission in PPs

Tibor Kiss **Katja Keßelmeier** **Antje Müller** **Claudia Roch** **Tobias Stadtfeld** **Jan Strunk**
Sprachwissenschaftliches Institut,
Ruhr-Universität Bochum
{tibor, kesselmeier, mueller, roch, stadtfeld, strunk}@linguistics.rub.de

Abstract

The realization of singular count nouns without an accompanying determiner inside a PP (determinerless PP, bare PP, Preposition-Noun Combination) has recently attracted some interest in computational linguistics. Yet, the relevant factors for determiner omission remain unclear, and conditions for determiner omission vary from language to language. We present a logistic regression model of determiner omission in German based on data obtained by applying annotation mining to a large, automatically and manually annotated corpus.

1 The problem and how to deal with it

Preposition-Noun Combinations (PNCs, sometimes called determinerless PPs or bare PPs) minimally consist of a preposition and a count noun in the singular that – despite requirements formulated elsewhere in the grammar of the respective language – appears without a determiner. The noun in a PNC can be extended through prenominal modification (1) and postnominal complementation (2). Still, a determiner is missing. The following examples are given from German.

(1) *auf parlamentarische Anfrage* ('after being asked in parliament'), *mit beladenem Rucksack* ('with loaded backpack'), *unter sanfter Androhung* ('under gentle threat')

(2) *Er wehrt sich gegen die Forderung nach* he defies REFL against the demand for *Stilllegung einer Verbrennungsanlage*. closedown an incineration plant
'He defies the demand for closing an incineration plant.'

PNCs occur in a wide range of languages (Himmelmann, 1998); the conditions for determiner omission, however, have not been detected yet, and conditions applying to one language do not carry over to other languages. In addition, speakers only reluctantly judge the acceptability of newly coined PNCs, so that reliance to introspective judgments cannot be assumed.

For English, Stvan (1998) and Baldwin et al. (2006) have claimed that either the semantics of the preposition or of the noun play a major role in determining whether a singular count noun may appear without a determiner in a PNC. Stvan (1998) assumes that nouns determine the well-formedness of PNCs (3) if the denotation of the noun occurs in a particular semantic field, while Baldwin et al. (2006) assume that certain prepositions impose selection restrictions on their nominal complements that allow for determiner omission (4).

(3) *from school, at school, in jail, from jail, …*

(4) *by train, by plane, by bus, by pogo stick, by hydro-foil …*

Interestingly, Le Bruyn et al. (2009) have observed that basic assumptions of Stvan's analysis do not apply to Dutch, French, or Norwegian. With regard to German, we observe that neither the pattern in (3) nor in (4) is productive. Constructions like (4) cannot be realized as PNCs in German, but require full PPs.

In the following, we propose an analysis of PNCs that combines corpus annotation, annotation mining (Chiarcos et al., 2008), and logistic regression modeling (Harrell, 2001). Annotation mining assumes that linguistically relevant generalizations can be derived in a bottom-up fashion from a suitably annotated corpus. Relevant hits in the corpus are mapped into a feature vector that serves as input for logistic regression classification. In the present case, the input con-

Coling 2008: Poster Volume, pages 561–569,
Beijing, August 2010

sists of sentences containing either PNCs or PPs. Binary logistic regression suggests itself as a classification method since the problem of PNCs can be rephrased as the following question: Under which conditions can an otherwise obligatory determiner be omitted?

The majority of required annotations can be derived automatically, but there are no available systems for the automatic determination of preposition senses in German, so preposition sense annotation has to be carried out manually and requires a language-specific tagset for preposition senses.

While our initial analysis is based on German data, the general methodology can be applied to other languages, provided that corpora receive proper annotation.

2 Corpus annotation

2.1 General characteristics

The present analysis is based on a newspaper corpus of the Swiss-German newspaper *Neue Zürcher Zeitung* from 1993 to 1999, comprising approx. 230 million words. The annotation is based on an XML-stand-off format. MMAX2 (Müller and Strube, 2006) is used for manual annotation. Annotations are carried out both for PNCs and for full-fledged PPs. For each preposition, the following data is considered: PNCs, where N is a count noun; corresponding PPs with the same count noun; and PPs containing count nouns not appearing inside PNCs.

The following annotations are provided for each dataset in the corpus:

Lexical level: part-of-speech, inflectional morphology, derivational morphology of nouns, count/mass distinction of nouns, interpretation of nouns, interpretation of prepositions, noun compounding.

Syntactic level: mode of embedding of the phrase (adjunct or complement), syntactic dependents of the noun, modification of the noun.

Global level: Is the phrase contained in a headline, title, or quotation? Is the phrase idiomatic? Headlines, titles, and quotations are particularly prone to text truncation and PNCs occurring here might not be the result of syntactic operations. Similarly, idiomatic PNCs *and* PPs might follow combination rules that differ from the general modes of combination. Hence, the annotations may serve to exclude these cases from general classification.

2.2 Automatic annotation

The following tools are employed for automatic annotation: Regression Forest Tagger (Schmid and Laws, 2008) for POS tagging and morphological analysis (the tagger contains the SMOR component for morphological analysis, cf. Schmid, 2004), and Tree Tagger (Schmid, 1995) for chunk parsing.

To determine noun meanings, we make use of two resources. The first resource is GermaNet (Kunze and Lemnitzer, 2002), the German version of WordNet. We employ 23 top-level categories, and each noun is annotated with every top-level category it belongs to.[1] Secondly, we use the computer lexicon HaGenLex (Hartrumpf et al., 2003), which offers specific sortal information derived from a formal ontology for each noun. Finally, we employ a classifier for the count/mass distinction. The classifier combines lexical statistics, expressed in terms of a decision tree classifier, with contextual information, which is handled by Naïve Bayes classification (cf. Stadtfeld 2010). The classification is based on the fine-grained distinctions first introduced in Allan (1980), but we employ a reduced set of five instead of eight classes. The classifier is type-based as it makes use of the relation between singular and plural realizations of noun lemmas, but takes the immediate context of the lemma into account.

Nouns are only assigned to a particular class if both classifiers come to the same result w.r.t. this class assignment. While this leads to some nouns being excluded from the count/mass distinction, the resulting classes show a high degree of precision.

2.3 Manual annotation of preposition senses

Prepositions are highly polysemous. What is more, the relation between a preposition and its senses has to be determined in a language-

[1] Nouns that are assigned to more than one top-level category are presumably homonymous or polysemous. We do not disambiguate the nouns. The reason is that individual features will be evaluated for their effect in a logistic model, and an ambiguous noun will receive a value in each feature. Hence, we can be sure that a significant semantic feature will be included in the classification.

specific manner. While the *Preposition Project* forms a basis for preposition sense annotation in English (cf. Litkowski and Hargraves 2005, 2007), little attention has been paid to specialized annotation schemata for preposition senses in German, which form the first prerequisite for a classification of preposition senses.

Based on four usage-based grammars and dictionaries of German (Duden 2002, Helbig and Buscha 2001, Durrell and Brée 1993, Schröder 1986), we have developed an annotation schema with a hierarchical structure, allowing for subtrees of preposition senses that require a fine-grained classification (such as TEMPORAL, SPATIAL, CAUSAL, and PRESENCE). For temporal and spatial interpretations, the annotation is further facilitated by the use of decision trees.[2]

Altogether, the annotation schema includes the following list of top-level categories: MODAL, CAUSAL, PRESENCE, SPATIAL, TEMPORAL, STATE, COMITATIVE, AGENT, REDUCTION/EXTENSION, PARTICIPATION, SUBORDINATION, RECIPIENT, AFFILIATION, CORRELATION/INTERACTION, TRANSGRESSION, ORDER, THEME, SUBSTITUTE, EXCHANGE, COMPARISON, RESTRICTIVE, COPULATIVE, ADVERSATIVE, DISTRIBUTIVE, STATEMENT/OPINION, CENTRE OF REFERENCE, and REALISATION.

Based on an extension of the weighted kappa statistic we have reached an overall kappa value (κ_w) of 0.657 and values between 0.551 and 0.860 for individual features (cf. Müller et al. 2010a). Two properties of the annotation schema prohibit the application of a standard kappa statistic: First, the schema allows subsorts, and secondly, a preposition may receive more than one annotation if its sense cannot be fully disambiguated. The values reported in Müller et al. (2010) for maximal subtypes such as SPATIAL (κ_w = 0.709) and TEMPORAL (κ_w = 0.860) can be equated to aggregate values in standard kappa statistics.

In the models presented below, we employ top-level categories only and have aggregated more specific sense annotations.

[2] The schema does not directly distinguish between local and directional senses, but makes use of cross-classification to deal with the distinction. Cf. Müller et al. (2010b).

3 Preparing logistic regression models for *ohne* ('without') and *unter* ('under', 'below')

The problem of PNCs, i.e. why a determiner is omitted in a construction which otherwise requires the realization of the determiner, can be rephrased as a problem for binary logistic regression and classification.

While binary logistic regression does not prohibit monocausal explanations, typical models for binary logistic regression employ more than one factor, and the value of the coefficients models the relative influence of the individual factors. Logistic regression thus does not only help to identify factors for determiner omission, but also reveals the interplay of multiple licensing conditions – thus possibly accounting for the relative difficulty to distinguish acceptable from inacceptable PNCs.

We are aiming at a description of PNCs in German for the 22 prepositions listed in (5).

(5) *an, auf, bei, dank, durch, für, gegen, gemäß, hinter, in, mit, mittels, nach, neben, ohne, seit, über, um, unter, vor, während, wegen*

These prepositions have been chosen on the basis of the following two assumptions: a) they appear in PNCs and PPs, and b) their 'typical' object is an NP.

We present logistic regression models of determiner realization for two prepositions: *ohne* ('without') and *unter* ('under', 'below'). The first preposition, *ohne,* is the only preposition that appears more often in PNCs than in PPs. The second preposition, *unter,* belongs to the class of highly polysemous prepositions. In fact, it is the preposition with the second largest number of senses (10 senses), only surpassed by *mit* ('with') (11 senses), which however appears much more often than *unter* in the corpus and thus requires further annotation. The following table summarizes the distribution of PNCs and PPs for both prepositions, after tokens that had been annotated as belonging to *headlines, quotations, telegram style sentences,* or as being idiomatic were excluded from the data. With regard to the first group (headlines etc.), the elimination mostly applies to PNCs, but among the PPs we found many idiomatic expressions and fixed phrases, which have also been excluded from modeling.

Preposition	Σ	PP	PNC
ohne	3,750	591	3,159
unter	5,181	4,334	857

Table 1. Data Distribution of PNCs and PPs

The analysis has been carried out in R (R Development Core Team, 2010) and makes extensive use of Harrell's DESIGN package (Harrell, 2001).

The feature vector consists of the dependent variable – the factor DET with its levels *no* and *yes* – and of relevant classificatory features representing the interpretation of the preposition (in terms of the features presented in section 2), the internal syntactic structure of the nominal projection (prenominal modification of N, syntactic arguments of N, internal structure of N as a compound, derivational status of N), the external syntactic embedding of the PNC or PP, and the interpretation of the noun.

Features starting with DEP signify syntactic arguments of the noun (DEP-S a sentential complement, DEP-NP an NP complement, etc.); the feature ADJA signifies the presence of one or more modifying adjectives; the feature COMPOUND indicates whether the noun in question is a compound. The feature GOVERNED indicates whether a noun or a verb governs the phrase. The feature NOMINALIZATION provides information about the derivational structure of the noun, in particular it indicates whether a noun is derived from a verb by use of the suffix *–ung*.

Features starting with GN are GermaNet top-level categories, features starting with HL are HaGenLex ontological sorts; both describe the interpretation of the noun.

The statistical modeling started with the assumption that each feature is relevant, so that an initial feature set of 92 features was considered. Feature elimination took place through *fast backwards elimination* (Lawless and Singhal, 1978) and manual inspection. The results of fast backwards elimination were not followed blindly. Following Harrell's (2001:56) suggestion, we have kept factors despite their low significance levels. In most cases, however, manual inspection and fast backwards elimination suggested the same results. The resulting models were subjected to *bootstrap validation* to identify possible overfitting (cf. section 5.1).

The value DET = *no* is taken to be the default value in the following models. As a consequence, negative values for coefficients indicate rising probability for an omission of a determiner, while positive coefficients shift odds in favor of a realization of the determiner.

4 Logistic models for the omission of a determiner with *ohne* and *unter*

The logistic regression models developed for the prepositions *ohne* and *unter* make use of 13 and 22 features, respectively. In each case, we have started with a full model fit (Harrell, 2001:58f.), evaluated the full model and eliminated factors through manual inspection and fast backwards elimination. The coefficients for the models for *ohne* and *unter* are reported in tables 2 and 3.

	Coef.	**S.E.**	**Wald Z**	**p**
INTERCEPT	-2.4024	0.1109	-21.66	0.000
NOMINAL.	-1.3579	0.1870	-7.26	0.000
ADJA	1.1360	0.1188	9.57	0.000
CAUSAL	1.2063	0.1302	9.26	0.000
COMITAT.	2.2821	0.5201	4.39	0.000
PARTICIP.	3.4027	0.4895	6.95	0.000
PRESENCE	-0.7780	0.1463	-5.32	0.000
DEP-S	5.0797	1.0542	4.82	0.000
DEP-NP	2.9752	0.1718	17.32	0.000
DEP-PP	2.1978	0.1487	14.78	0.000
GN-RELAT.	-1.0292	0.4072	-2.53	0.011
GN-ATTR.	-1.3528	0.3038	-4.45	0.000
GN-EVENT	-0.8431	0.1431	-5.89	0.000
GN-ARTE.	-0.4117	0.1564	-2.63	0.008

Table 2. Coefficients for a logistic regression model of determiner omission with *ohne*.[3]

[3] In the following tables, S.E. stands for standard error Wald Z reports the Z-score of the Wald statistic, which is determined by divided the value of the coefficient through its standard error. The squared Wald Z statistic is χ^2-distributed and thus indicates the goodness of fit for the coefficients of the model.

	Coef.	S.E.	Wald Z	p
INTERCEPT	-0.4379	0.1657	-2.64	0.008
NOMINAL.	-0.8346	0.2259	-3.70	0.000
ADJA	-1.0177	0.1432	-7.11	0.000
COMPOUND	2.1719	0.2538	8.56	0.000
GOVERNED	1.9894	0.3017	6.59	0.000
SPATIAL	2.3237	0.2044	11.37	0.000
CAUSAL	1.3047	0.2272	5.74	0.000
SUBORD.	3.0529	0.2559	11.93	0.000
ORDER	3.4228	0.1861	18.40	0.000
TRANSGR.	4.4186	0.3677	12.02	0.000
DEP-S	8.4717	4.0734	2.08	0.037
DEP-NP	0.8551	0.1436	5.95	0.000
DEP-PP	0.3043	0.2170	1.40	0.161
GN-GROUP	0.5241	0.2563	2.04	0.041
GN-COMM.	-0.9149	0.1443	-6.34	0.000
GN-LOC.	2.2704	0.6208	3.66	0.000
GN-REL.	-2.1161	0.6022	-3.51	0.000
GN-POSS.	-0.8482	0.3665	-2.31	0.021
GN-ATTR.	-2.2847	0.2741	-8.33	0.000
GN-ARTE.	0.4169	0.1601	2.60	0.009
GN-HUM.	1.8870	0.4999	3.77	0.000
HL-AD	-1.0253	0.1888	-5.43	0.000
HL-AS	-1.4214	0.3804	-3.74	0.000

Table 3. Coefficients for a logistic model of determiner omission with *unter.*

General measures of the two models are reported in table 4. Somers' D_{xy} describes the proportion of observations, for which the model provides an appropriate class probability. D_{xy} can be derived from C, the corresponding receiver operating characteristic curve area, since $D_{xy} = 2 \times (C–0.5)$. Model L.R. (likelihood ratio) indicates the improvement reached by including the predictors. Degrees of freedom (d.f.) have been omitted from table 4, as they correspond to the number of predictors, i.e. 12 in the case of *ohne* and 23 in the case of *unter*. The high figures for Somers' D_{xy} are reassuring.

	Model L.R.	p	C	D_{xy}
ohne	1,063.5	0	0.876	0.753
unter	2,245.6	0	0.937	0.874

Table 4. Model Quality.

4.1 The model for *ohne*

Starting with the model in table 2, we can identify several groups of factors:

The first group comprises the interpretation of the preposition. The group discriminates between determiner omission and realization. The semantic features CAUSAL, COMITATIVE, and PARTICIPATION show positive coefficients, suggesting that prepositions receiving the aforementioned interpretations tend to favor an 'ordinary' NP including a determiner. The interpretation PRESENCE, on the other hand, shows a negative coefficient and thus suggests the omission of a determiner. There are further senses of *ohne,* which do not have a significant effect on determiner omission/realization.

Turning to the representation of syntactic argument structure of the noun, we find that the coefficients of DEP-S, DEP-NP, and DEP-PP receive positive values throughout. The presence of syntactic complements thus shifts odds in favor of determiner realization. There is a strong preference against determiner omission with DEP-S, and somewhat weaker values for Dep-NP and Dep-PP, respectively. A comparison of interpretation and complement realization offers a general assessment of PNCs. As *ohne* and *unter* share only a few senses, we do not necessarily expect that the discerning senses relevant for a realization of a PNC with *ohne* carry over to *unter;* but we do expect that features pertaining to the syntactic structure of the nominal complement play a role not only for *ohne,* but for *unter* (or for prepositions admitting PNCs in general) as well. And this prediction is actually borne out in the model for *unter.* The model thus already offers interesting insights not only w.r.t. the realization conditions of PNCs and PPs headed by *ohne,* but for broader analyses of PNCs as well.

We will return to the role and value of the features ADJA and NOMINALIZATION in section 4.3.

The last group comprises the semantic characteristics of nouns derived from GermaNet. If a noun is classified as belonging to the relevant GermaNet top-level categories, determiner omission is favored.

4.2 The model for *unter*

A first glance at the model for *unter* shows that it requires a larger set of predictors than the model for *ohne.* In part, this is due to the higher degree of polysemy of *unter:* with more senses, we expect more semantic predictors to enter the discrimination. In addition, a wider range of senses

also allows for a wider range of selection restrictions, and hence for a larger number of different sortal specifications for selected nouns. The higher complexity of the model, however, should not conceal a peculiarity of this model that casts serious doubt on the idea that PNCs are monocausally licensed by particular senses of a preposition: the model selects five senses from the ten top level interpretations of *unter,* but the coefficients are unsigned. Thus, the model indicates that the senses SPATIAL, CAUSAL, SUBORDINATION, ORDER, and TRANSGRESSION *block* the omission of a determiner. What we do not find are senses that favor the omission of a determiner.

The features DEP-S, DEP-NP, and DEP-PP again favor the realization of a determiner. A comparison of the coefficient of DEP-S to the coefficients of DEP-NP and DEP-PP shows, however, that the presence of a sentential complement has a strong influence on determiner realization, while NP- and PP-complements may still occur in PNCs, as their coefficients are relatively low (also in comparison to the coefficients of these values for *ohne).*[4]

In more general terms, we suspect a general mechanism relating sentential complementation to the realization of the determiner, a topic to be addressed in future research.

It should also be noted that the external syntactic realization of the phrase plays a role for *unter.* The feature GOVERNED did not play a role for *ohne,* but suggests the realization of a determiner for *unter.* The reason might be that few verbs or nouns govern the preposition *ohne*. Prepositional objects headed by *unter,* however, are more common. Prepositional objects headed by *ohne* make up only 1.2 % of the occurrences of *ohne* in the present corpus, while the share of prepositional objects headed by *unter* is three times larger: 3.6 %.

Finally, we note that a variety of sortal classifications for nouns suggest either an omission or realization of the determiner, supporting the assumption that in addition to the preposition's meaning, the meaning of the noun plays a role. GermaNet top-level categories were already discriminating in the model for *ohne;* but the model for *unter* also makes use of HaGenLex sortal categories (HL-AD and HL-AS). The predictors stand for dynamic and static concepts that both receive an abstract interpretation. Their inclusion is particularly interesting, as it is sometimes claimed (e.g. Bale and Barner, 2009) that 'abstract' nouns are never to be classified as count nouns.

4.3 General assessment of the models

Both models show that the realization of syntactic complements, of sentential complements in particular, seems to impede determiner omission. That syntactic complexity does not seem to play a role per se, can be deduced from the coefficients for the factor ADJA: While ADJA favors determiner realization with *ohne,* it prohibits determiner realization with *unter.*

The role of morphological derivation through *–ung,* as represented by the factor NOMINALIZATION, is the same in both models: derived nominals shift odds in favor of determiner omission. While the derivational structure might be considered a formal property of the construction, it might also reflect an underlying denotational distinction between events and objects, which has to be clarified in future work.

It is a striking feature of the model for *unter that* we do not find interpretational features of the preposition *unter* that favor determiner omission. Taken together with the other factors in the two models presented, the analysis suggests a picture rather different from the (more or less) monocausal analyses of Stvan (1998) and Baldwin et al. (2006). With regard to *unter* a model in the sense of Baldwin et al. (2006) could only provide negative rules of the form *"if P does not mean this, its nominal complement may be realized without a determiner",* but such a model would lead to less precision than the multicausal model presented here.

5 Validation of the models

5.1 Bootstrap validation

Logistic regression models may suffer from overfitting the data. We have thus carried out a

[4] One could argue against the inclusion of the coefficient for DEP-PP altogether, as it does not seem to be significant ($p > 0.05$) in the first place. However, we have followed Harrell's (2001) advice that blind exclusion of seemingly insignificant factors may not lead to model improvement. In fact, models for *unter* including Dep-PP outperform models excluding this feature.

bootstrap validation of both models and applied penalized maximum likelihood estimation (Harrell, 2001) to the models. The results of the initial (non-penalized) models are reported in Table 5 and Table 6, where we report values for D_{xy} and the average maximal error of the model. Bootstrap validation makes use of sampling with replacement. The training samples for evaluation thus may contain certain instances many times, but some original data will never be sampled and can thus be used for testing the models. Bootstrap validation is carried out 200 times, the results being averaged. The overfitting of the models is determined by the optimism derived from the bootstrap evaluation.

	D_{xy}	E_{max}
Original Index	0.7525	0.0000
Training	0.7578	0.0000
Test	0.7497	0.0123
Optimism	0.0080	0.0123
Corrected Index	0.7445	0.0123

Table 5. Bootstrap validation of model for *ohne.*[5]

	D_{xy}	E_{max}
Original Index	0.8737	0.0000
Training	0.8741	0.0000
Test	0.8690	0.0072
Optimism	0.0051	0.0072
Corrected Index	0.8685	0.0072

Table 6. Bootstrap validation of model for *unter.*

Penalized maximum likelihood estimation (Harrell, 2001:207) for both models resulted in penalties of 0.3 and 0.8, respectively, based on Akaike's AIC. The updated models have again been bootstrap validated, resulting in the improved values presented in table 7 and table 8.

	D_{xy}	E_{max}
Original Index	0.7526	0.0000
Training	0.7570	0.0000
Test	0.7500	0.0096
Optimism	0.0070	0.0096
Corrected Index	0.7456	0.0096

[5] E_{max} is the maximal error determined in average over the bootstrap runs.

Table 7. Bootstrap validation of penalized model for *ohne.*

	D_{xy}	E_{max}
Original Index	0.8736	0.0000
Training	0.8744	0.0000
Test	0.8692	0.0055
Optimism	0.0052	0.0055
Corrected Index	0.8684	0.0055

Table 8. Bootstrap validation of penalized model for *unter.*

5.2 Representing the influence of factors in a nomogram

The respective influence of individual factors can be read of a nomogram (Banks, 1985) derived from the models presented above (we make use of a tabular presentation for reasons of legibility). The nomogram for *ohne* consists of the tables 9 and 10. Table 9 lists the individual scores for the factors in the model for *ohne,* were 0 indicates that the pertinent property is not present and 1 indicates that the property is present. Table 10 maps the sum to probability of determiner omission.

Predictor	**0**	**1**
NOMINALIZATION	27	0
ADJA	0	22
CAUSAL	0	24
COMITATIVE	0	45
PARTICIPATION	0	67
PRESENCE	15	0
DEP-S	0	100
DEP-NP	0	59
DEP-PP	0	43
GN-RELATION	20	0
GN-ATTRIBUTE	27	0
GN-EVENT	17	0
GN-ARTEFACT	8	0

Table 9. Nomogram: individual scores of predictors for *ohne.*

Total Points	**Pr("Omission of Det")**
118	0.9
134	0.8
144	0.7
153	0.6

161	0.5
169	0.4
178	0.3
188	0.2
204	0.1

Table 10. Nomogram: mapping from total points to probability of determiner omission.

As an illustration, consider pairs of *ohne* and a noun with the values in (6) and (7).

(6) NOMINALIZATION = 1, ADJA = 1, COMITATIVE = 1, all other senses including PRESENCE = 0, all GN features = 0, DEP features = 0.

(7) NOMINALIZATION = 0, ADJA = 1, PRESENCE = 1, all other senses = 0, GN-ATTRIBUTE = 1, all other GN features = 0, DEP features = 0.

Given the individual scores for the factors in table 9, the total number of points for the combination in (6) is 144, leading to a probability of 0.7 that a determiner will be omitted in the construction. In other words, a determiner omission is likely with the feature set given in (6). In (7), we reach a total of 92 only, so that the likelihood of determiner omission rises above 0.9.

6 Summary and prospects

The models presented support the general assumption that the realization or omission of a determiner in a prepositional phrase should be analyzed as a multicausal phenomenon. The logistic regression analysis presents evidence for the assumption that the senses of the preposition and the interpretation of the noun (possibly governed by selection restrictions of the preposition) as well as the syntactic complexity of the embedded nominal projection are major factors in determining whether an article can be dropped or not.

With regard to the complexity of the nominal projection, the two models presented here indicate that it is not complexity per se, but that the realization of a complement of the noun, in particular of a sentential complement, clearly raises the probability of article realization. While this is a speculation, based on the models presented here, it might very well be that this dependency reflects a deeper referential requirement.

In developing further models for prepositions, we expect that the realization of a complement of the noun will establish itself as a common factor, but this has to await further research and model development.

Acknowledgement

We gratefully acknowledge the funding of our research by the *Deutsche Forschungsgemeinschaft* (DFG) under project grant KI 759/5-1.

References

Allan, Keith. 1980. Nouns and countability. *Language* 56(3):541-567.

Baldwin, Timothy, John Beavers, Leonoor van der Beek, Francis Bond, Dan Flickinger, and Ivan Sag. 2006. In search of a systematic treatment of determinerless PPs. In Patrick Saint-Dizier (eds.), *Syntax and Semantics of Prepositions*. Springer, Dordrecht, 163-179.

Bale, Alan, and David Barner. 2009. The interpretation of functional heads: Using comparatives to explore the mass/count distinction. *Journal of Semantics 26,* 217-252.

Banks, J. 1985. Nomograms. In S. Kotz and N.L. Johnson (eds.), *Encyclopedia of Statistical Sciences,* Vol. 6. Wiley, New York.

Chiarcos, Christian, Stefanie Dipper, Michael Götze, Ulf Leser, Anke Lüdeling, Julia Ritz, and Manfred Stede. 2008. A flexible framework for integrating annotations from different tools and tagsets. *Traitement Automatique des Langues.* Special Issue Platforms for Natural Language Processing. ATALA, 49 (2).

Duden. 2002. Duden. Deutsch als Fremdsprache. Bibliographisches Institut and F.A. Brockhaus AG, Mannheim.

Durell, Martin and David Brée. 1993. German temporal prepositions from an English perspective. In Cornelia Zelinsky-Wibbelt (ed.), The Semantics of Prepositions. From Mental Processing to atural Language Processing. De Gruyter, Berlin/New York, 295-325.

Harrell, Frank E. 2001. *Regression modeling strategies: with applications to linear models, logistic regression, and survival analysis*. Springer: New York.

Hartrumpf, Sven, Hermann Helbig, and Rainer Osswald. 2003. The Semantically Based Computer Lexicon HaGenLex - Structure and Technological Environment. *Traitement automatique des langues* 44(2):81-105.

Helbig, Gerhard and Joachim Buscha. 2001. Deutsche Grammatik. Ein Handbuch für den Ausländerunterricht. Leipzig, Langenscheidt.

Himmelmann, Nikolaus. 1998. Regularity in irregularity: Article use in adpositional phrases. *Linguistic Typology*, 2:315–353.

Kunze, Claudia, and Lothar Lemnitzer. 2002. GermaNet - representation, visualization, application. *Proc. LREC 2002*, main conference, Vol V., 1485-1491.

Lawless, J. and K. Singhal. 1978. Efficient screening on nonnormal regression models. *Biometrics 34*:318-327.

Le Bruyn, Bert, Henriëtte de Swart, and Joost Zwarts. 2009. *Bare PPs across languages*. Presented at the Workshop on Bare nouns, Paris.

Müller, Antje, Olaf Hülscher, Claudia Roch, Katja Keßelmeier, Tobias Stadtfeld, Jan Strunk, and Tibor Kiss. 2010. An Annotation Schema for Preposition Senses in German. Proceedings of ACL-LAW IV, Uppsala, Sweden.

Müller, Antje, Katja Keßelmeier, Claudia Roch, Jan Strunk, Tobias Stadtfeld and Tibor Kiss. 2010. Creating a Feature Space for the Annotation of Preposition Senses in German. Linguistic Evidence,TÜbingen 2010.

Müller, Christoph, and Michael Strube. 2006. Multi-level annotation of linguistic data with MMAX2. In Sabine Braun, Kurt Kohn, Joybrato Mukherjee, (eds.), *Corpus Technology and Language Pedagogy: New Resources, New Tools, New Methods.* Peter Lang, Frankfurt a.M., 197-214.

Nivre, Joakim. 2006. *Inductive Dependency Parsing (Text, Speech, and Language Technology)*. New York: Springer.

R Development Core Team. 2010. R: *A language and environment for statistical computing*. Foundation for Statistical Computing, Vienna, Austria. http://www.rproject.org.

Stadtfeld, Tobias. 2010. Determining the Countability of English and German Nouns. Ms. Ruhr-University Bochum.

Schmid, Helmut. 1995. Improvements in part-of-speech tagging with an application to German. In *Proceedings of the EACL SIGDAT Workshop*, Dublin, Ireland, March.

Schmid, Helmut, Arne Fitschen, and Ulrich Heid. 2004. SMOR: A German computational morphology covering derivation, composition, and inflection. In *Proceedings of LREC 2004*, 1263-1266, Lisbon, Portugal.

Schmid, Helmut, and Florian Laws. 2008. Estimation of conditional probabilities with decision trees and an application to fine-grained POS tagging. In *Proceedings of COLING 2008*, Manchester, UK.

Schröder, Jochen. 1986. Lexikon deutscher Präpositionen. Leipzig, VEB Verlag Enzyklopädie.

Stvan, Laurel S. 1998. *The Semantics and Pragmatics of Bare Singular Noun Phrases*. Ph.D. thesis, Northwestern University, Evanston/ Chicago, IL.

Using Syntactic and Semantic based Relations for Dialogue Act Recognition

Tina Klüwer, Hans Uszkoreit, Feiyu Xu
Deutsches Forschungszentrum für Künstliche Intelligenz (DFKI)
Projektbüro Berlin
{tina.kluewer,uszkoreit,feiyu}@dfki.de

Abstract

This paper presents a novel approach to dialogue act recognition employing multi-level information features. In addition to features such as context information and words in the utterances, the recognition task utilizes syntactic and semantic relations acquired by information extraction methods. These features are utilized by a Bayesian network classifier for our dialogue act recognition. The evaluation results show a clear improvement from the accuracy of the baseline (only with word features) with 61.9% to an accuracy of 67.4% achieved by the extended feature set.

1 Introduction

Dialogue act recognition is an essential task for dialogue systems. Automatic dialogue act classification has received much attention in the past years either as an independent task or as an embedded component in dialogue systems. Various methods have been tested on different corpora using several dialogue act classes and information coming from the user input.

The work presented in this paper is part of a dialogue system called *KomParse* (Klüwer et al., 2010), which is an application of a NL dialogue system combined with various question answering technologies in a three-dimensional virtual world named *Twinity*, a web-based online product of the Berlin startup company *Metaversum*[1]. The KomParse NPCs provide various services through conversation with game users such as selling pieces of furniture to users via text based conversation.

The main task of the input interpretation component of the agent is the detection of the dialogue acts contained in the user utterances. This classification is done via a cue-based method with various features from multi-level knowledge sources extracted from the incoming utterance considering a small context of the previous dialogue.

In contrast to existing systems using mainly lexical features, i.e. words, single markers such as punctuation (Verbree et al.,) or combinations of various features (Stolcke et al., 2000) for the dialogue act classification, the results of the interpretation component presented in this paper are based on syntactic and semantic relations. The system first gathers linguistic information coming from different levels of deep linguistic processing similar to (Allen et al., 2007). The retrieved information is used as input for an information extraction component that delivers the relations embedded in the actual utterance (Xu et al., 2007). These relations combined with additional features (a small dialogue context and mood of the sentence) are then utilized as features for the machine-learning based recognition.

The classifier is trained on a corpus originating from a Wizard-of-Oz experiment which was semi-automatically annotated. It contains automatically annotated syntactic relations namely, predicate argument structures, which were checked and corrected manually afterwards. Furthermore these relations are enriched by manual annotation with semantic frame information from VerbNet to gain an additional level of semantic richness. These two representations of relations, the syntax-based re-

[1] http://www.metaversum.com/

Coling 2008: Poster Volume, pages 570–578,
Beijing, August 2010

lations and the VerbNet semantic relations, were used in separate training steps to detect how much the classifier can benefit from either notations.

A systematic analysis of the data has been conducted. It turns out that a comparatively small set of syntactic relations cover most utterances, which can moreover be expressed by an even smaller set of semantic relations. Because of this observation as well as the overall performance of the classifier the interpretation is extended with an additional rule based approach to ensure the robustness of the system.

The paper is organized as follows: Section 2 provides an overview about existing dialogue act recognition systems and the features they use for classification.
Section 3 introduces the original data used as basis for the annotation and the classification task.
In Section 4 the annotation that provides the necessary information for the dialogue act classification and involves the relation extraction is described in detail. The annotation is split into three main steps: The annotation of dialogue information (section 4.1), the integration of syntactic information (section 4.2) and finally the manual annotation of VerbNet predicate and role information in section 4.3.
Section 5 presents the results of the actual classification task using different feature sets and in Section 6 the results and methods are summarized.
Finally, Section 7 provides a brief description of the rule-based interpretation and presents an outlook on future work.

2 Related Work

Dialogue Acts (DAs) represent the functional level of a speaker's utterance, such as a greeting, a request or a statement. Dialogue acts are verbal or nonverbal actions that incorporate participant's intentions originating from the theory of Speech Acts by Searle and Austin (Searle, 1969). They provide an abstraction from the original input by detecting the intended action of an utterance, which is not necessarily inferable from the surface input (see the two requests in the following example).

Can you show me a red car please?

Please show me a red car!

To detect the action included in an utterance, different approaches have been suggested in recent years which can be clustered into two main classes: The first class uses AI planning methods to detect the intention of the utterance based on belief states of the communicating agents and the world knowledge. These systems are often part of an entire dialogue system e.g. in a conversational agent which provides the necessary information about current beliefs and goals of the conversation participants at runtime. One example is the TRIPS system (Allen et al., 1996). Because of the huge amount of reasoning, systems in this class generally gather as much linguistic information as possible.

The second class uses cues derived from the actual utterance to detect the right dialogue act, mostly using machine learning methods. This class gained much attention due to less computational costs. The probabilistic classifications are carried out via training on labeled examples of dialogue acts described by different feature sets. Frequently used cues for dialogue acts are lexical features such as the words of the utterance or ngrams of words for example in (Verbree et al.,), (Zimmermann et al., 2005) or (Webb and Liu, 2008). Although the performance of the classification task is difficult to compare, because of the variety of different corpora, dialogue act sets and algorithms used, these approaches do provide considerably good results. For example (Verbree et al.,) achieve accuracy values of 89% on the ICSI Meeting Corpus containing 80.000 utterances with a dialogue act set of 5 distinct dialogue act classes and amongst others the features "ngrams of words" and "ngrams of POS information".

Another group of systems utilizes acoustic features derived from Automatic Speech Recognition for automatic dialogue act tagging (Surendran and Levow, 2006), context features like the preceding dialogue act or ngrams of previous dialogue acts (Keizer and Akker, 2006).

However grammatical and semantic information is not that often incorporated into feature sets, with the exception of single features such as the

Dialogue Act	Meaning	Frequency
REQUEST	The utterance contains a wish or demand	449
REQUEST_INFO	The utterance contains a wish or demand regarding information	154
PROPOSE	The utterance serves as suggestion or showing of an object	216
ACCEPT	The utterance contains an affirmation	167
REJECT	The utterance contains a rejection	88
PROVIDE_INFO	The utterance provides an information	156
ACKNOWLEDGE	The utterance is a backchannelling	9

Table 1: The used Dialogue Act Set

type of verbs or arguments or the presence or absence of special operators e.g. wh-phrases (Andernach, 1996). (Keizer et al., 2002) use among others linguistic features like sentence type for classification with Bayesian networks. Although (Jurafsky et al., 1998) already noticed a strong correlation between selected dialogue acts and special grammatical structures, approaches using grammatical structure were not very succesful.

While grammatical and semantic features are not often incorporated into dialogue act recognition, they are a commonly used in related fields like automatic classification of rhetorical relations. For example (Sporleder and Lascarides, 2008) and (Lapata and Lascarides, 2004) extract verbs as well as their temporal features derived from parsing to infer sentence internal temporal and rhetorical relations. Their best model for analysing temporal relations between two clauses achieves 70.7% accuracy. (Subba and Eugenio, 2009) also show a significant improvement of a discourse relation classifier incorporating compositional semantics compared to a model without semantic features. Their VerbNet based frame semantics yield in a better result of 4.5%.

3 The Data

The data serving as the basis for the relation identification as well as the training corpus for the dialogue act classifier is taken from a Wizard-of-Oz experiment (Bertomeu and Benz, 2009) in which 18 users furnish a virtual living room with the help of a furniture sales agent. Users buy pieces of furniture and room decoration from the agent by describing their demands and preferences in a text chat. During the dialogue with the agent, the preferred objects are then selected and directly put to the right location in the apartment. In the experiments, users spent one hour each on furnishing the living room by talking to a human wizard controlling the virtual sales agent. The final corpus consists of 18 dialogues containing 3,171 turns with 4,313 utterances and 23,015 alpha-numerical strings (words). The following example shows a typical part of such a conversation:

USR.1: And do we have a little side table for the TV?
NPC.1: I could offer you another small table or a sideboard.
USR.2: Then I'll take a sideboard that is similar to my shelf.
NPC.2: Let me check if we have something like that.

Table 2: Example Conversation from the Wizard-of-Oz Experiment

4 Annotation

The annotation of the corpus is carried out in several steps.

4.1 Pragmatic Annotation

The first annotation step consists of annotating discourse and pragmatic information including dialogue acts, projects according to (Clark, 1996), sentence mood, the topic of the conversation and an automatically retrieved information state for every turn of the conversations. From the annotated information the following elements were selected as features in the final recognition system:

- The dialogue acts which carry the intentions of the actual utterance as well as the last preceding dialogue act. The set used for annotation is a domain specific set containing the dialogue acts shown in table 1.
- The sentence mood. Sentence mood was annotated with one of the following values: declarative, imperative, interrogative.

- The topic of the utterance. The topic value is coreferent with the currently discussed object. Topic can consist of an object class (e.g. sofa) or an special object instance (sofa_1836). The topic of the directly preceding utterance was chosen as a feature too.

4.2 Annotation with Predicate Argument Structure

The second annotation step, applied to the utterance level of the input, automatically enriches the annotation with predicate argument structures. Each utterance is parsed with a predicate argument parser and annotated with syntactic relations organized according to PropBank (Palmer et al., 2005) containing the following features: Predicate, Subject, Objects, Negation, Modifiers, Copula Complements.

A single relation mainly consists of a predicate and the belonging arguments. Verb modifiers like attached PPs are classified as "argM" together with negation ("argM_neg") and modal verbs ("argM_modal"). Arguments are labeled with numbers according to the found information for the actual structure. PropBank is organized in two layers, the first one being an underspecified representation of a sentence with numbered arguments, the second one containing fine-grained information about the semantic frames for the predicate comparable to FrameNet (Baker et al., 1998). While the information in the second layer is stable for each verb, the values of the numbered arguments can change from verb to verb. While for one verb the "arg0" may refer to the subject of the verb, another verb may encapsulate a direct object behind the same notation "arg0". This is very complicated to handle in a computational setup, which needs continuous labeling for the successive components. Therefore the arguments were in general named as in PropBank but consistently numbered by syntactic structure. This means for example that the subject is always labeled as "arg1".

Consider the example "Can you put posters or pictures on the wall?". The syntactic relation will yield in the following representation:

```
<predicate: put>
<ArgM_modal: can>
<Arg1: you>
<Arg2: posters or pictures>
<ArgM: on the wall>
```

Predicate Argument Structure Parser The syntactic predicate argument structure that constitutes the syntactic relations and serves as basis for the VerbNet annotation, is automatically retrieved by a rule-based predicate argument parser. The rules utilized by the parser describe subtrees of dependency structures in XML by means of relevant grammatical functions. For detecting verbs with two arguments in the input, for instance, a rule can be written describing the dependency structure for a verb with a subject and an object. This rule would then detect every occurrence of the structure "Verb-Subj-Obj" in a dependency tree. This sample rule would express the following constraints: The matrix unit should be of the part of speech "Verb" , The structure belonging to this verb must contain a "nsubj" dependency and an "obj" dependency.

The rules deliver raw predicate argument structures, in which the detected arguments and the verb serve as hooks for further information lookup in the input. If a verb fulfills all requirements described by the rule, in a second step all modificational arguments existing in the structure are recursively acquired. The same is done for modal arguments as well as modifiers of the arguments such as determiners, adjectives or embedded prepositions. After the generation of the main predicate argument structure from the grammatical functions, the last step inserts the content values present in the actual input into the structure to get the syntactic relations for the utterance.

Before the input can be parsed with the predicate argument parser, some preprocessing steps of the corpus are needed. These include:

Input Cleaning The input data coming from the users contain many errors. Some string substitutions as well as the external Google spellchecker were applied to the input before any further processing.

Segmentation For clausal separation we apply a simple segmentation via heuristics based on punctuation.

POS Tagging Then the input is processed by

the external part-of-speech tagger TreeTagger (Schmid, 1994).

The embedded dependency parser is the Stanford Dependency Parser (de Marneffe and Manning, 2008), but other dependency parsers could be employed instead. The predicate argument parser is an standalone software and can be used either as a system component or for batch processing of a text corpus.

4.3 VerbNet Frame Annotation

The last step of annotation consists of the manual annotation of semantic predicate classes and semantic roles. Moreover, the automatically determined syntactic relations are checked and corrected if possible. VerbNet (Schuler, 2005) is utilized as a source for semantic information. The VerbNet role set consists of 21 general roles used in all VerbNet classes. Examples of roles in this general role set are “agent”, “patient” and “theme”.

For the manual addition of the semantic frame information a web-based annotation tool has been developed. The annotation tool shows the utterance which should be annotated in the context of the dialogue including the information from the preceding annotation steps. All VerbNet classes containing the current predicate are listed as possibilities for the predicate classification together with their syntactic frames. The annotators can select the appropriate predicate class and frame according to the arguments found in the utterance. If an argument is missing in the input that is required in the selected frame a null argument is added to the structure. If the right predicate class is existing, but the predicate is not yet a member of the class, it is added to the VerbNet files. In case the right predicate class is found but the fitting frame is missing, the frame is added to the VerbNet files. Thus during annotation 35 new members have been added to the existing VerbNet classes, 4 Frames and 4 new subclasses. Via these modifications, a version of VerbNet has been developed that can be regarded as a domain-specific VerbNet for the sales domain.

During the predicate classification, the annotators also assign the appropriate semantic roles to the arguments belonging to the selected predicate. The semantic roles are taken from the selected VerbNet frame.

From the annotated semantic structure, semantic relations are inferred such as the one in the following example:

```
<predicate: put-3.1>
<agent: you>
<theme: posters or pictures>
<destination: on the wall>
```

5 Dialogue Act Recognition

Two datasets are derived from the corpus: The dataset containing the utterances of the users (CST) and one dataset containing the utterances of the wizard (NPC), whereas the NPC corpus is cleaned from the “protocol sentences”. Protocol sentences are canned sentences the wizard used in every conversation, for example to initialize the dialogue. For the experiments, the two single datasets “NPC” and “CST” as well as a combined dataset called “ALL” are used. Unfortunately from the original 4,313 utterances in total, many utterances could not be used for the final experiments. First, fragments are removed and only the utterances found by the parser to contain a valid predicate argument structure are used. After protocol sentences are taken out too, a dataset of 1702 valid utterances remains. Moreover, 292 utterances are annotated to contain no valid dialogue act and are therefore not suitable for the recognition task. Of the remaining utterances, 171 predicate argument structures were annotated as wrong because of completely ungrammatical input. In this way we arrive at a dataset of 804 instances for the users and 435 for the wizard, summing up to 1239 instances in total.

The features used for dialogue act recognition exploit the information extracted from the different annotation steps:

- Context features: The last preceding dialogue act, equality between the last preceding topic and the actual topic, sentence mood
- Syntactic relation features: Syntactic predicate class, arguments, negation
- VerbNet semantic relation features: VerbNet predicate class, VerbNet frame arguments, negation

- Utterance features: The original utterances without any modifications

Different sets of features for training and evaluation are generated from these:

DATASET_Syn: All utterances of the specified dataset described via syntactic relation and context features.

DATASET_VNSem: All utterances of the specified dataset described via VerbNet semantic relations and context features.

DATASET_Syn_Only: All utterances of the specified dataset only described via the syntactic relations.

DATASET_VNSem_Only: All utterances of the specified dataset only described via the VerbNet semantic relations.

DATASET_Context_Only: All utterances of the specified dataset described via the context features and negation without any information regarding relations.

DATASET_Utterances_Context: The utterances of the specified dataset as strings combined with the whole set of context features without further relation extraction results.

DATASET_Utterances: Only the utterances of the specified dataset as strings. This and the last "Utterances"-set serve as baselines.

Dialogue Act Recognition is carried out via the Bayesian network classifier AOEDsr from the WEKA toolkit. AODEsr augments AODE, an algorithm averaging over all of a small space of alternative naive-Bayes-like models that have weaker independence assumptions than naive Bayes, with Subsumption Resolution (Zheng and Webb, 2006). Evaluation is performed using crossfolded evaluation.

All results of the experiments are given in terms of accuracy.

Results for the dataset "All" comparing the syntactic relations with VerbNet relations as well as the pure utterances and context are shown in table 4.

Dataset	Accuracy
All_Syn	67.4%
All_VNSem	66.8%
All_Utterances_Context	61.9%
All_Utterances	48.1%

Table 4: Dialogue Act Classification Results for the "ALL" Datasets

The best result is achieved with the syntactic information, although the VerbNet information provides an abstraction over the predicate classification. Both the set containing the VerbNet relations as well as the syntactic relations are much better than the set containing only the context and the original utterances. The dataset containing only the utterances could not reach 50%.

Although the experiments show much better results using the relations instead of the original utterance, the overall accuracy is not very satisfying. Several reasons for this phenomenon come into consideration. While it can to a certain extend be the fault of the classifying algorithm (see table 8 for some tests with a ROCCHIO based classifier), the main reason might as well lie in the imprecise boundaries of the dialogue act classes: Several categories are hard to distinguish even for a human annotator as you can see from the wrongly classified examples in table 3. Another possibility can be the comparatively small number of total training instances.

For the NPC dataset the results are slightly better and much better still for the set CST, which is due to a smaller number (6) of dialogue acts: The dialogue act "PROPOSE", which is the act for showing an object or proposing a possibility, was not used by any user, but only by the wizard.

Dataset	Accuracy
CST_Syn	73.1%
NPC_Syn	68.5%

Table 5: Dialogue Act Classification Results for Datasets "CST" and "NPC"

To find out if one sort of features is especially important for the classification we reorga-

Utterance	Right Classification	Classified As
What do you think about this one?	request_info	propose
Let see what you have and where we can put it	request_info	request

Table 3: Wrongly classified instances

nize the training sets to contain only the context features without the relations (All_Context_Only) on the one hand and only the relational information without the context features on the other hand (All_Syn_Only and All_VNSem_Only). Results are shown in table 6.

Dataset	Accuracy
All_Context_Only	56.6%
All_VNSem_Only	53.5%
All_Syn_Only	50.8%

Table 6: Dialogue Act Classification Results for Context and Relation sets

Table 6 shows that the results are considerably worse if only parts of the features are used. The set with context feature performs 3,1% better than the best set with the relations only. Furthermore the VerbNet semantic relation set leads to nearly 3% better accuracy, which may mean that the abstraction of semantic predicates provides a better mapping to dialogue acts after all if used without further features which may be ranked more important by the classifier.

Besides the experiments with the Bayesian networks, additional experiments are performed using a modified ROCCHIO algorithm similar to the one in (Neumann and Schmeier, 2002). Three different datasets were tested (see table 7).

Dataset	Accuracy
All_Utterances	70.1%
All_Utterances_Context	73.2%
All_Syn	74.4%

Table 8: Dialogue Act Classification Results using the ROCCHIO Algorithm

Table 8 shows that the baseline dataset containing only the utterances already provides much better results with the ROCCHIO algorithm, delivering 70.1% which is more than 10% more accuracy compared to the 48.1% of the Bayesian classifier. If tested together with the context features the accuracy of the utterance dataset raises to 73.2% and, after including the relational information, even to 74.4%. Thus, the results of this ROCCHIO experiment also prove that the employment of the relation information leads to improved accuracy of the classification.

6 Conclusion

This paper reports on a novel approach to automatic dialogue act recognition using syntactic and semantic relations as new features instead of the traditional features such as ngrams of words.

Different feature sets are constructed via an automatic annotation of syntactic predicate argument structures and a manual annotation of VerbNet frame information. On the basis of this information, both the syntactic relations as well as the semantic VerbNet-based relations included in the utterances can be extracted and added to the feature sets for the recognition task. Besides the relation information the employed features include information from the dialogue context (e.g. the last preceding dialogue act) and other features like sentence mood.

The feature sets have been evaluated with a Bayesian network classifier as well as a ROCCHIO algorithm. Both classifiers demonstrate the benefits gained from the relations by exploiting the additionally provided information. While the difference between the best baseline feature set and the best relation feature set in the Bayesian network classifier yields a 5,5% boost in accuracy (61.9% to 67.4%), the ROCCHIO setup exceeds the boosted accuracy by another 1,5% , starting from a higher baseline of 73.2%. Based on the observed complexity of the classification task we expect that the benefit of the relational informa-

Predicate	Instances	Example
see-30.1	59	I would like to see a table in front of the sofa
put-9.1	74	Can you put it in the corner?
reflexive_appearance-48.1.2	80	Show me the red one
own-100	137	Do you have wooden chairs?
want-32.1	153	I would like some plants over here

Table 7: The Main Semantic Relations Found in the Data Sorted by Predicate

tion may turn out to be even more significant on larger learning data.

7 Future Work

The results in section 5 show that the pure classification cannot be used as interpretation component in isolation, but additional methods have to be incorporated. In a preceding analysis of the data it was found that certain predicates are very frequently uttered by the users. In the syntactic predicate scenario the total number of different predicates is 80, whereas the semantic predicates build up a total number of 66. The class containing the predicates with one to ten occurrences constitutes 137 of 1239 instances. The remaining 1101 instances are covered by only 21 different predicate classes. These predicates together with their arguments constitute a set of common domain relations for the sales domain. The main domain relations found are shown in table 7.

The figures suggest that the interpretation at least for the domain relations can be established in a robust manner, wherefore the agent's interpretation component was extended to a hybrid module including a robust rule based method. To derive the necessary rules a rule generator was developed and the rules covering the used feature set (including the context features, sentence mood and the syntactic relations) were automatically generated from the given data.

Future work will focus on the evaluation of these automatically derived rules on a recently collected but not yet annotated dataset from a second Wizard-of-Oz experiment, carried out in the same furniture sales setting.

Additional experiments are planned for evaluating the relation-based features in dialogue act recognition on other corpora tagged with different dialogue acts in order to test the overall performance of our classification approach on more transparent dialogue act sets.

Acknowledgements

The work described in this paper was partially supported through the project "KomParse" funded by the ProFIT program of the Federal State of Berlin, co-funded by the EFRE program of the European Union. Additional support came from the project TAKE, funded by the German Ministry for Education and Research (BMBF, FKZ: 01IW08003).

References

Allen, James F., Bradford W. Miller, Eric K. Ringger, and Teresa Sikorski. 1996. A robust system for natural spoken dialogue. In *Proceedings of ACL 1996*.

Allen, James, Mehdi Manshadi, Myroslava Dzikovska, and Mary Swift. 2007. Deep linguistic processing for spoken dialogue systems. In *DeepLP '07: Proceedings of the Workshop on Deep Linguistic Processing*, Morristown, NJ, USA.

Andernach, Toine. 1996. A machine learning approach to the classification of dialogue utterances. *CoRR*, cmp-lg/9607022.

Baker, Collin F., Charles J. Fillmore, and John B. Lowe. 1998. The berkeley framenet project. In *Proceedings of COLING 1998*.

Bertomeu, Nuria and Anton Benz. 2009. Annotation of joint projects and information states in human-npc dialogues. In *Proceedings of CILC-09*, Murcia, Spain.

Clark, H.H. 1996. *Using Language*. Cambridge University Press.

de Marneffe, Marie C. and Christopher D. Manning. 2008. The Stanford typed dependencies representation. In *Coling 2008: Proceedings of the workshop on Cross-Framework and Cross-Domain Parser Evaluation*, Manchester, UK.

Jurafsky, Daniel, Elizabeth Shriberg, Barbara Fox, and Traci Curl. 1998. Lexical, prosodic, and syntactic cues for dialog acts.

Keizer, Simon and Rieks op den Akker. 2006. Dialogue act recognition under uncertainty using bayesian networks. *Nat. Lang. Eng.*, 13(4).

Keizer, Simon, Rieks op den Akker, and Anton Nijholt. 2002. Dialogue act recognition with bayesian networks for dutch dialogues. In *Proceedings of the 3rd SIGdial workshop on Discourse and dialogue*, Morristown, NJ, USA.

Klüwer, Tina, Peter Adolphs, Feiyu Xu, Hans Uszkoreit, and Xiwen Cheng. 2010. Talking npcs in a virtual game world. In *Proceedings of the System Demonstrations Section at ACL 2010*.

Lapata, Mirella and Alex Lascarides. 2004. Inferring sentence-internal temporal relations. In *Proceedings of the North American Chapter of the Association for Computational Linguistics*, pages 153–160.

Neumann, Günter and Sven Schmeier. 2002. Shallow natural language technology and text mining. *Künstliche Intelligenz. The German Artificial Intelligence Journal*.

Palmer, Martha, Daniel Gildea, and Paul Kingsbury. 2005. The proposition bank: An annotated corpus of semantic roles. *Comput. Linguist.*, 31(1).

Schmid, Helmut. 1994. In *Proceedings of the International Conference on New Methods in Language Processing*.

Schuler, Karin Kipper. 2005. *Verbnet: a broad-coverage, comprehensive verb lexicon*. Ph.D. thesis, Philadelphia, PA, USA.

Searle, John R. 1969. *Speech acts : an essay in the philosophy of language / John R. Searle*. Cambridge University Press, London.

Sporleder, Caroline and Alex Lascarides. 2008. Using automatically labelled examples to classify rhetorical relations: A critical assessment. *Natural Language Engineering*, 14(3).

Stolcke, Andreas, Klaus Ries, Noah Coccaro, Elizabeth Shriberg, Rebecca Bates, Daniel Jurafsky, Paul Taylor, Rachel Martin, Carol Van, and Ess dykema Marie Meteer. 2000. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational Linguistics*, 26.

Subba, Rajen and Barbara Di Eugenio. 2009. An effective discourse parser that uses rich linguistic information. In *NAACL '09*, Morristown, NJ, USA.

Surendran, Dinoj and Gina-Anne Levow. 2006. Dialog act tagging with support vector machines and hidden markov models. In *Interspeech*.

Verbree, A.T., R.J. Rienks, and D.K.J. Heylen. Dialogue-act tagging using smart feature selection: results on multiple corpora. In Raorke, B., editor, *First International IEEE Workshop on Spoken Language Technology SLT 2006*.

Webb, Nick and Ting Liu. 2008. Investigating the portability of corpus-derived cue phrases for dialogue act classification. In *Proceedings of COLING 2008*, Manchester, UK.

Xu, Feiyu, Hans Uszkoreit, and Hong Li. 2007. A seed-driven bottom-up machine learning framework for extracting relations of various complexity. In *Proceedings of ACl (07)*, Prague, Czech Republic.

Zheng, Fei and Geoffrey I. Webb. 2006. Efficient lazy elimination for averaged one-dependence estimators. In *ICML*, pages 1113–1120.

Zimmermann, Matthias, Yang Liu, Elizabeth Shriberg, and Andreas Stolcke. 2005. Toward joint segmentation and classification of dialog acts in multiparty meetings. In *Proc. Multimodal Interaction and Related Machine Learning Algorithms Workshop (MLMI05*, page 187.

Automatic Allocation of Training Data for Rapid Prototyping of Speech Understanding based on Multiple Model Combination

Kazunori Komatani[†] **Masaki Katsumaru**[†] **Mikio Nakano**[‡]
Kotaro Funakoshi[‡] **Tetsuya Ogata**[†] **Hiroshi G. Okuno**[†]

† Graduate School of Informatics, Kyoto University
{komatani,katumaru,ogata,okuno}@kuis.kyoto-u.ac.jp

‡ Honda Research Institute Japan Co., Ltd.
{nakano,funakoshi}@jp.honda-ri.com

Abstract

The optimal choice of speech understanding method depends on the amount of training data available in rapid prototyping. A statistical method is ultimately chosen, but it is not clear at which point in the increase in training data a statistical method become effective. Our framework combines multiple automatic speech recognition (ASR) and language understanding (LU) modules to provide a set of speech understanding results and selects the best result among them. The issue is how to allocate training data to statistical modules and the selection module in order to avoid overfitting in training and obtain better performance. This paper presents an automatic training data allocation method that is based on the change in the coefficients of the logistic regression functions used in the selection module. Experimental evaluation showed that our allocation method outperformed baseline methods that use a single ASR module and a single LU module at every point while training data increase.

1 Introduction

Speech understanding in spoken dialogue systems is the process of extracting a semantic representation from a user's speech. That is, it consists of automatic speech recognition (ASR) and language understanding (LU). Because vocabularies and language expressions depend on individual systems, it needs to be constructed for each system, and accordingly, training data are required for each. To collect more real training data, which will lead to higher performance, it is more desirable to use a prototype system than that based on the Wizard-of-Oz (WoZ) method where real ASR errors cannot be observed, and to use a more accurate speech understanding module. That is, in the bootstrapping phase, spoken dialogue systems need to operate before sufficient real data have been collected.

We have been addressing the issue of rapid prototyping on the basis of the "Multiple Language model for ASR and Multiple language Understanding (MLMU)" framework (Katsumaru et al., 2009). In MLMU, the most reliable speech understanding result is selected from candidates produced by various combinations of multiple ASR and LU modules using hand-crafted grammar and statistical models. A grammar-based method is still effective at an early stage of system development because it does not require training data; Schapire et al. (2005) also incorporated human-crafted prior knowledge into their boosting algorithm. By combining multiple understanding modules, complementary results can be obtained by different kinds of ASR and LU modules.

We propose a novel method to allocate available training data to statistical modules when the amount of training data increases. The training data need to be allocated adaptively because there are several modules to be trained, and they would cause overfitting without data allocation. There are speech understanding modules that have language models (LMs) for ASR and LU models

Coling 2008: Poster Volume, pages 579–587,
Beijing, August 2010

(LUMs), and a selection module that selects the most reliable speech understanding result from multiple candidates in the MLMU framework. When the amount of available training data is small, and an LUM and the selection module are trained on the same data set, they are trained under a closed-set condition, and thus the training data for the selection module include too many correct understanding results. In such cases, the data need to be divided into subdata sets to avoid overfitting. On the other hand, when the amount of available training data is large, so that overfitting does not occur, all available data should be used to train each statistical module to prepare as much training data as possible.

We therefore develop a method for switching data allocation policies. More specifically, two points are automatically determined at which statistical modules with more parameters start to be trained. As a result, better overall performance is achieved at every point while the amount of training data increases, compared with all combinations of a single ASR module and a single LU module.

2 Related Work

It is important to consider the amount of available training data when designing a speech understanding module. Many statistical LU methods have been studied, e.g., (Wang and Acero, 2006; Jeong and Lee, 2006; Raymond and Riccardi, 2007; Hahn et al., 2008; Dinarelli et al., 2009). They generally outperform grammar-based LU methods when a sufficient amount of training data is available; but sufficient training data are not necessarily available during rapid prototyping. Several LU methods were constructed using a small amount of training data (Fukubayashi et al., 2008; Dinarelli et al., 2009). Fukubayashi et al. (2008) constructed an LU method based on the weighted finite state transducer (WFST), in which filler transitions accepting arbitrary inputs and transition weights were added to a hand-crafted FST. This method is placed between a grammar-based method and a statistical method because a statistically selected weighting scheme is applied to a hand-crafted grammar model. Therefore, the amount of training data can be smaller compared with general statistical LU methods, but this method does not outperform them when plenty of training data are available. Dinarelli et al. (2009) used a generative model for which overfitting is less prone to occur than discriminative models when the amount of training data is small, but they did not use a grammar-based model, which is expected to achieve reasonable performance even when the amount of training data is very small.

Raymond et al. (2007) compared the performances of statistical LU methods for various amounts of training data. They used a statistical finite-state transducer (SFST) as a generative model and a support vector machine (SVM) and conditional random fields (CRF) as discriminative models. The generative model was more effective when the amount of data was small, and the discriminative models were more effective when it was large. This shows that the performance of an LU method depends on the amount of training data available, and therefore, LU methods need to be switched automatically. Wang et al. (2002) developed a two-stage speech understanding method by applying statistical methods first and then grammatical rules. They also examined the performance of the statistical methods at their first stage for various amounts of training data and confirmed that the performance is not very high when a small amount of data is used.

Schapire et al. (2005) showed that accuracy of call classification in spoken dialogue systems improved by incorporating hand-crafted prior knowledge into their boosting algorithm. Their idea is the same as ours in that they improve the system's performance by using hand-crafted human knowledge while only a small amount of training data is available. We furthermore solve the data allocation problem because there are multiple statistical models to be trained in speech understanding, while their call classification has only one statistical model.

3 MLMU Framework

MLMU is the framework for selecting the most reliable speech understanding result from multiple speech understanding modules (Katsumaru et al., 2009). In this paper, we furthermore adapt the selection module to the amount of available train-

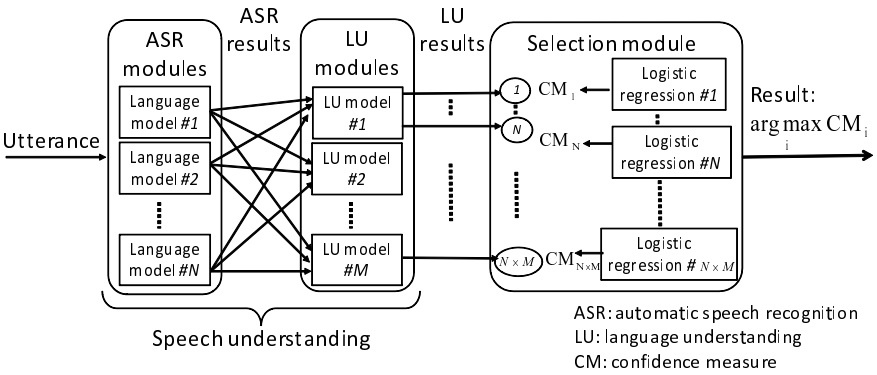


Figure 1: Overview of speech understanding framework MLMU

ing data. More specifically, the allocation policy of training data is changed and thus appropriate LMs and LUMs are selected as its result.

An overview of MLMU is shown in Figure 1. MLMU uses multiple LMs for ASR and multiple LUMs and selects the most reliable speech understanding result from all combinations of them. We denote a speech understanding module as SU_i ($i = 1, \ldots, n$). Its result is a semantic representation consisting of a set of concepts. The concept is either a semantic slot and its value or an utterance type. Note that $n = N \times M$, when N LMs and M LUMs are used. The confidence measure per utterance for a result of i-th speech understanding module SU_i is denoted as CM_i. The speech understanding result having the highest confidence measure is selected as the final result for the utterance. That is, the result is the output of SU_m where $m = \mathrm{argmax}_i\, CM_i$.

The confidence measure is calculated by logistic regression based on the features of each speech understanding result. A logistic regression function is constructed for each speech understanding module SU_i:

$$CM_i = \frac{1}{1 + e^{-(a_{i1}F_{i1} + \ldots + a_{i7}F_{i7} + b_i)}}. \qquad (1)$$

Parameters $a_{i1}, \ldots, a_{i7}$ and b_i are determined by using training data. In the training phase, teacher signal 1 is given when a speech understanding result is completely correct; that is, when no error is contained in the result. Otherwise, 0 is given. We use seven features, $F_{i1}, F_{i2}, \ldots, F_{i7}$, as independent variables. Each feature value is normalized so as to make its mean zero and its variance one.

Table 1: Features of speech understanding result obtained from SU_i

F_{i1}:	Acoustic score normalized by utterance length
F_{i2}:	Difference between F_{i1} and normalized acoustic scores of verification ASR
F_{i3}:	Average concept CM in understanding result
F_{i4}:	Minimum concept CM in understanding result
F_{i5}:	Number of concepts in understanding result
F_{i6}:	Whether any understanding result is obtained
F_{i7}:	Whether understanding result is yes/no

CM: confidence measure

The features used are listed in Table 1. Compared with those used in our previous paper (Katsumaru et al., 2009), we deleted ones that were highly correlated with other features and added ones regarding content of the speech understanding results. Features F_{i1} and F_{i2} are obtained from an ASR result. Another ASR with a general large vocabulary LM is executed for verifying the i-th ASR result. F_{i2} is the difference between its score and F_{i1} (Komatani et al., 2007). These two features represent the reliability of the ASR result. F_{i3} and F_{i4} are calculated for each concept in the LU result on the basis of the posterior probability of the 10-best ASR candidates (Komatani and Kawahara, 2000). F_{i5} is the number of concepts in the LU result. This feature is effective because the LU results of lengthy utterances tend to be erroneous in a grammar-based LU. F_{i6} represents the case when an ASR result is not accepted by the subsequent LU module. In such cases, no speech understanding result is obtained, which is

U1: It is June ninth.
ASR result:
- **grammar** "It is June ninth."
- **N-gram** "It is June noon and"
LU result:
- **grammar + FST** "month:6 day:9 type:refer-time"
- **N-gram + WFST** "month:6 type:refer-time"

U2: I will borrow it on twentieth.
(Underlined part is out-of-grammar.)
ASR result:
- **grammar** "Around two pm on twentieth."
- **N-gram** "Around two at ten on twentieth."
LU result:
- **grammar + FST** "day:20 hour:14 type:refer-time"
- **N-gram + WFST** "day:20 type:refer-time"

Combination of LM and LUM is denoted as "LM+LUM".

Figure 2: Example of speech understanding results in MLMU framework

regarded as an error. F_{i7} is added because affirmative and negative responses, typically "Yes" and "No", tend to be correctly recognized and understood.

Figure 2 depicts an example when multiple ASRs based on LMs and multiple LUs are used. In short, the correct speech understanding result is obtained from a different combination of LMs and LUMs.

4 Automatic Allocation of Training Data Using Change in Coefficients

The training data need to be allocated to the speech understanding modules (i.e., statistical LM and statistical LUM) and the selection module. If more data are allocated to the ASR and LU modules, the performances of these modules are improved, but the overall performance is degraded because of the low performance of the selection module. On the other hand, even if more training data are allocated to the selection module, the performance of each ASR and LU module remains low.

4.1 Allocation Policy

We focus on the convergence of the logistic regression functions when the amount of training data increases. The convergence is defined as the change in their coefficients, which will appear later as Equation 2, and determines two points

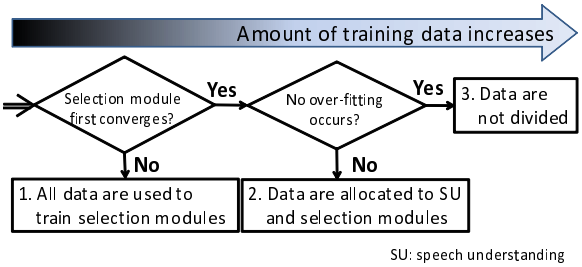


Figure 3: Flowchart of data allocation

during the increase in training data, and thus three phases are defined. The flowchart of data allocation is depicted in Figure 3. The three phases are explained below.

In the first phase, the first priority is given to the selection module. This is because the logistic regression functions used in the selection module converge with relatively less training data than those in the statistical ASR and LU modules for speech understanding; there are eight parameters for each logistic regression function as shown in Equation 1, far fewer than for other statistical models such as N-gram and CRF. The output from a speech understanding module that employs grammar-based LM and LUM would be the most reliable in many cases because its performance is better than that of other statistical modules when a very small amount of training data is available. As a result, equivalent or better performance would be achieved than methods using a single ASR module and a single LU module.

In the second phase, the training data are also allocated to the speech understanding modules after the selection module converges. This aims to improve the performance of the speech understanding modules by allocating as much training data to them as possible. The amount of training data is fixed in this phase to the amount allocated to the selection module determined in the first phase. The remaining data are used to train the speech understanding modules.

When the performances of all the speech understanding modules stabilize, the allocation phase proceeds to the third one. After this point, we hypothesize that overfitting does not occur in this phase because plenty of training data are available. All available data are used to train all mod-

ules without dividing the data in this phase.

4.2 Determining When to Switch Allocation Policies

Automatic switching from one phase to the next requires the determination of two points in the number of training utterances: when the selection module first converges ($k_{onlysel}$) and when the speech understanding modules all become stable (k_{nodiv}). These points are determined by focusing on the changes in the coefficients of the logistic regression functions when the number of utterances used as training data increases. We observe the sum of the changes in the coefficients of the functions and then identify the points at which the changes converge. The points are determined individually by the following algorithm.

Step 1 Construct two logistic regression functions for speech understanding module SU_i by using k and $(k + \delta k)$ utterances out of k_{max} utterances, where k_{max} is the amount of training data available.

Step 2 Calculate the change in coefficients from the two logistic regression functions by

$$\Delta_i(k) = \sum_j |a_{ij}(k + \delta k) - a_{ij}(k)| + |b_i(k + \delta k) - b_i(k)|, \quad (2)$$

where $a_{ij}(k)$ and $b_i(k)$ denote the parameters of the logistic regression functions, shown in Equation 1, for speech understanding module SU_i, when k utterances are used to train the functions.

Step 3 If $\Delta_i(k)$ becomes smaller than threshold θ, consider that the training of the functions has converged, and record this k as the point of convergence. If not, return to Step 1 after $k \leftarrow k + \delta k$.

The δk is the minimum unit of training data containing various utterances. We set it as the number of utterances in one dialogue session, whose average was 17. Threshold θ was set to 8, which corresponds to the number of parameters in the logistic regression functions. No experiments were conducted to determine if better performance could be achieved with other choices of θ[1].

The first point, $k_{onlysel}$, is determined using the speech understanding module that uses no training data. Specifically, we used "grammar+FST" as method SU_i. Here, "LM+LUM" denotes a combination of LM for ASR and LUM. If the function converges at k utterances, we set $k_{onlysel}$ to k and fix the k utterances as training data used by the selection module. The remaining $(k_{max} - k)$ utterances are allocated to the speech understanding modules, that is, the LMs and LUMs. Note that if k becomes equal to k_{max} before Δ_i converges, all training data are allocated to the selection module; that is, no data are allocated to the LMs and LUMs. In this case, no output is obtained from statistical speech understanding modules, and only outputs from the grammar-based modules are used.

The second point, k_{nodiv}, is determined on the basis of the speech understanding module that needs the largest amount of data for training. The amount of data needed depends on the number of parameters. Specifically, we used "N-gram+CRF" as SU_i in Equation 2. If the function converges, we hypothesize that the performance of all the speech understanding modules stabilize and thus overfitting does not occur. We then stop the division of training data, and use all available data to train the statistical modules.

5 Experimental Evaluation

5.1 Target Data and Implementation

We used a data set previously collected through actual dialogues with a rent-a-car reservation system (Nakano et al., 2007) with 39 participants. Each participant performed 8 dialogue sessions, and 5900 utterances were collected in total. Out of these utterances, we used 5240 for which the automatic voice activity detection (VAD) results agreed with manual annotation. We divided the utterances into two sets: 2121 with 16 participants as training data and 3119 with 23 participants as the test data.

[1]We do not think the value is very critical after seeing the results shown in Figure 4.

We constructed another rent-a-car reservation system to evaluate our allocation method. The system included two language models (LMs) and four language understanding models (LUMs). That is, eight speech understanding results in total were obtained. The two LMs were a grammar-based LM ("grammar", hereafter) and a domain-specific statistical LM ("N-gram"). The grammar model was described by hand to be equivalent to the FST model used in LU. The N-gram model was a class 3-gram and was trained on a transcription of the available training data. The vocabulary size was 281 for the grammar model and 420 for the N-gram model when all the training data were used. The ASR accuracies of the grammar and N-gram models were 67.8% and 90.5% for the training data and 66.3% and 85.0% for the test data when all the training data were used. We used Julius (ver. 4.1.2) as the speech recognizer and a gender-independent phonetic-tied mixture model as the acoustic model (Kawahara et al., 2004). We also used a domain-independent statistical LM with a vocabulary size of 60250, which was trained on Web documents (Kawahara et al., 2004), as the verification model.

The four LUMs were a finite-state transducer (FST) model, a weighted FST (WFST) model, a keyphrase-extractor (Extractor) model, and a conditional random fields (CRF) model. In the FST-based LUM, the FST was constructed by hand. The WFST-based LUM is based on the method developed by Fukubayashi et al. (2008). The WFSTs were constructed by using the MIT FST Toolkit (Hetherington, 2004). The weighting scheme used for the test data was selected by using training data (Fukubayashi et al., 2008). In the extractor-based LUM, as many parts as possible in the ASR result were simply transformed into concepts. As the CRF-based LUM, we used open-source software, CRF++[2], to construct the LUM. As its features, we use a word in the ASR result, its first character, its last character, and the ASR confidence of the word. Its parameters were estimated by using training data.

The metric used for speech understanding performance was concept understanding accuracy, defined as

$$1 - \frac{\text{SUB} + \text{INS} + \text{DEL}}{\text{no. of concepts in correct results}},$$

where SUB, INS, and DEL denote the numbers of substitution, insertion, and deletion errors.

[2]http://crfpp.sourceforge.net/

Table 2: Absolute degradation in oracle accuracy when each module was removed

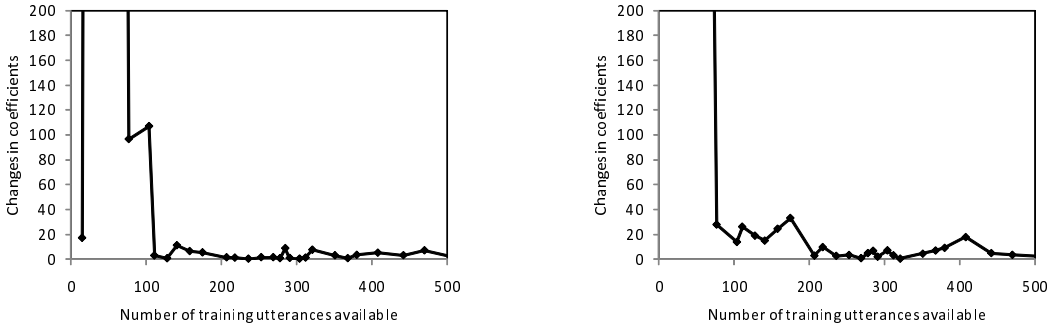
Case	(A)	(B)
With all modules (%)	86.6	90.1
w/o grammar ASR	-12.0	-1.1
w/o N-gram ASR	-6.1	-7.7
w/o FST LUM	-0.4	0.0
w/o WFST LUM	-1.2	-0.5
w/o Extractor LUM	-0.1	0.0
w/o CRF LUM	-0.6	-3.7
(w/o FST & Extractor LUMs)	-1.0	-0.1

(A): 141 utterances with 1 participant
(B): 2121 utterances with 16 participants

5.2 Effectiveness of Using Multiple LMs and LUMs

We investigated how much the performance of our framework degraded when one ASR or LU module was removed. We used the oracle accuracies, i.e., when the most appropriate result was selected by hand. The result reveals the contribution of each ASR and LU module to the performance of the framework. A module is regarded as more important when the accuracy is degraded more when it is removed than when another one is removed. Two cases (A) and (B) were defined: when the amount of available training data was (A) small and (B) large. We used 141 utterances with 1 participant for case (A) and 2121 utterances with 16 participants for case (B). The results are shown in Table 2.

When a small amount of training data was available (case (A)), the accuracy was degraded by 12.0 points when the grammar-based ASR module was removed and 6.1 points when the N-gram-based ASR module was removed. The accuracy was thus degraded substantially when either ASR module was removed. This indicates that the two ASR modules work complementarily.



(a) grammar+FST (b) N-gram+CRF

Figure 4: Change in the sum of coefficients Δ_i when amount of training data increases ("LM+LUM" denotes combination of LM and LUM)

On the other hand, when a large amount of training data was available (case (B)), the accuracy was degraded by 1.1 points when the grammar-based ASR was removed. This means that it became less important when there are plenty of training data because the coverage of the N-gram-based ASR became wider. In short, especially when the amount of training data is smaller, speech understanding modules based on a handcrafted grammar are more important because of the low performance of statistical modules.

Concerning the LUMs, the accuracy was degraded when any of the LUM modules was removed when a small amount of training data was available. When a large amount of training data was available, the module based on CRF in particular became more important.

5.3 Results and Evaluation of Automatic Allocation

Figure 4 shows the change in the sum of the coefficients, Δ_i, with the increase in the amount of training data. In Figure 4(a), the change was very large while the amount of training data was small, and decreased dramatically and converged around one hundred utterances. By applying θ (=8) to Δ_i, we set 111 utterances as the first point, $k_{onlysel}$, up to which all the training data are allocated to the selection module, as described in Section 4.1. Similarly, from the results shown in Figure 4(b), we set 207 utterances as the second point, k_{nodiv}, from which the training data are not divided.

To evaluate our method for allocating training data, we compared it with two baseline methods:

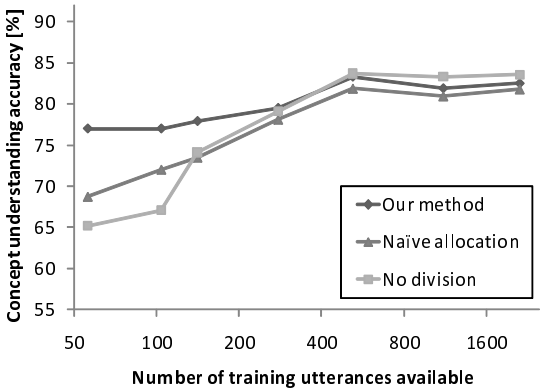


Figure 5: Results of allocation methods

- No-division method: All data available at each point were used to train both the speech understanding modules and the selection module. That is, the same data set was used to train them.
- Naive-allocation method: Training data available at each point were allocated equally to the speech understanding modules and the selection module.

As shown in Figure 5, our method had the best concept understanding accuracy when the amount of training data was small, that is, up to about 278 utterances. This indicates that our method for allocating the available training data is effective when the amount of training data is small.

This result is explained more specifically by us-

Table 3: Concept understanding accuracy for 141 utterances

	Accuracy (%)
Our method	77.9
Naive allocation	73.5
No division	74.1

ing the case in which 141 utterances were used as the training data. 111 ($= k_{onlysel}$) were secured to train the selection module and 30 utterances were allocated to train the speech understanding modules. As shown in Table 3, the accuracy with our method was 3.8 points higher than that with the no-division baseline method. This was achieved by avoiding the overfitting of the logistic regression functions; i.e., the data input to the functions became similar to the test data due to allocation, so the concept understanding accuracy for the test set was improved. The accuracy with our method was 4.4 points higher than that with the naive allocation baseline method. This was because the amount of training data allocated to the selection module was less than our method, and accordingly the selection module was not trained sufficiently.

5.4 Comparison with methods using a single ASR and a single LU

Figure 6 plots concept understanding accuracy with our method against baseline methods using a single ASR module and a single LU module for various amounts of training data. Each module for comparison was constructed by using all available training data at each point while training data increased; i.e., the same condition as our method. The accuracies of only three speech understanding modules are shown in the figure, out of the eight obtained by combining two LMs for ASR and four LUMs. These three are the ones with the highest accuracies while the amount of training data increased. Our method switched the allocation phase at 111 and 207 utterances, as described in Section 5.3.

Our method performed equivalently or better than all baseline methods even when only a small amount of training data was available. As a result, our method outperformed all the baseline methods

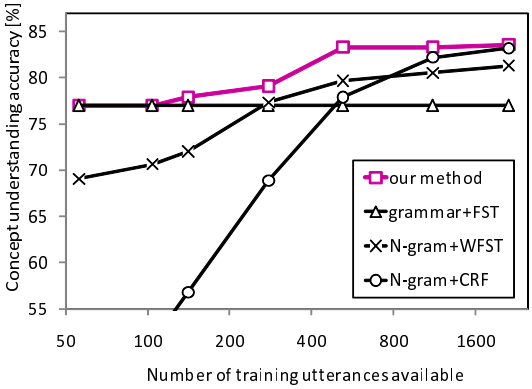


Figure 6: Comparison with baseline methods using single speech understanding

at every point while training data increase.

6 Conclusion

We developed a method to automatically allocate training data to statistical modules so as to avoid performance degradation caused by overfitting. Experimental evaluation showed that speech understanding accuracies achieved by our method were equivalent or better than the baseline methods based on all combinations of a single ASR module and a single LU module at every point while training data increase. This includes a case when a very small amount of training data is available. We also showed empirically that the training data should be allocated while an amount of training data is not sufficient. Our method allocated available training data on the basis of our allocation policy described in Section 4.1, and outperformed the two baselines where the training data were equivalently allocated and not allocated.

When plenty of training data were available, there was no difference between our method and the speech understanding method that requires the most training data, i.e., N-gram+CRF, as shown in Figure 6. It is possible that our method combining multiple speech understanding modules would outperform it as Schapire et al. (2005) reported. In their data, there were some examples that only a hand-crafted rules can parse. Including such a task as more complicated language understanding grammar is required, verification of our method in other tasks is one of the future works.

References

Dinarelli, Marco, Alessandro Moschitti, and Giuseppe Riccardi. 2009. Re-Ranking Models for Spoken Language Understanding. In *Proc. European Chapter of the Association for Computational Linguistics (EACL)*, pages 202–210.

Fukubayashi, Yuichiro, Kazunori Komatani, Mikio Nakano, Kotaro Funakoshi, Hiroshi Tsujino, Tetsuya Ogata, and Hiroshi G. Okuno. 2008. Rapid prototyping of robust language understanding modules for spoken dialogue systems. In *Proc. International Joint Conference on Natural Language Processing (IJCNLP)*, pages 210–216.

Hahn, Stefan, Patrick Lehnen, and Hermann Ney. 2008. System Combination for Spoken Language Understanding. In *Proc. Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pages 236–239.

Hetherington, Lee. 2004. The MIT Finite-State Transducer Toolkit for Speech and Language Processing. In *Proc. Int'l Conf. Spoken Language Processing (ICSLP)*, pages 2609–2612.

Jeong, Minwoo and Gary Geunbae Lee. 2006. Exploiting non-local features for spoken language understanding. In *Proc. COLING/ACL 2006 Main Conference Poster Sessions*, pages 412–419.

Katsumaru, Masaki, Mikio Nakano, Kazunori Komatani, Kotaro Funakoshi, Tetsuya Ogata, and Hiroshi G. Okuno. 2009. Improving speech understanding accuracy with limited training data using multiple language models and multiple understanding models. In *Proc. Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pages 2735–2738.

Kawahara, Tatsuya, Akinobu Lee, Kazuya Takeda, Katsunobu Itou, and Kiyohiro Shikano. 2004. Recent progress of open-source LVCSR engine Julius and Japanese model repository. In *Proc. Int'l Conf. Spoken Language Processing (ICSLP)*, pages 3069–3072.

Komatani, Kazunori and Tatsuya Kawahara. 2000. Flexible mixed-initiative dialogue management using concept-level confidence measures of speech recognizer output. In *Proc. Int'l Conf. Computational Linguistics (COLING)*, pages 467–473.

Komatani, Kazunori, Yuichiro Fukubayashi, Tetsuya Ogata, and Hiroshi G. Okuno. 2007. Introducing utterance verification in spoken dialogue system to improve dynamic help generation for novice users. In *Proc. 8th SIGdial Workshop on Discourse and Dialogue*, pages 202–205.

Nakano, Mikio, Yuka Nagano, Kotaro Funakoshi, Toshihiko Ito, Kenji Araki, Yuji Hasegawa, and Hiroshi Tsujino. 2007. Analysis of user reactions to turn-taking failures in spoken dialogue systems. In *Proc. 8th SIGdial Workshop on Discourse and Dialogue*, pages 120–123.

Raymond, Christian and Giuseppe Riccardi. 2007. Generative and Discriminative Algorithms for Spoken Language Understanding. In *Proc. Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pages 1605–1608.

Shapire, Robert E., Marie Rochery, Mazin Rahim, and Narendra Gupta. 2005. Boosting with prior knowledge for call classification. *IEEE Trans. on Speech and Audio Processing*, 13(2):174–181.

Wang, Ye-Yi and Alex Acero. 2006. Discriminative models for spoken language understanding. In *Proc. Int'l Conf. Spoken Language Processing (INTERSPEECH)*, pages 2426–2429.

Wang, Ye-Yi, Alex Acero, Ciprian Chelba, Brendan Frey, and Leon Wong. 2002. Combination of Statistical and Rule-based Approaches for Spoken Language Understanding. In *Proc. Int'l Conf. Spoken Language Processing (ICSLP)*, pages 609–612.

DL Meet FL: A Bidirectional Mapping between Ontologies and Linguistic Knowledge*

Hans-Ulrich Krieger and **Ulrich Schäfer**
Language Technology Lab
German Research Center for Artificial Intelligence (DFKI)
{krieger|ulrich.schaefer}@dfki.de

Abstract

We present a transformation scheme that mediates between description logics (DL) or RDF-encoded ontologies and type hierarchies in feature logics (FL). The *DL-to-FL* direction is illustrated by an implemented offline procedure that maps ontologies with large, dynamically maintained instance data to named entity (NE) and information extraction (IE) resources encoded in typed feature structures. The *FL-to-DL* translation is exemplified by a (currently manual) translation of so-called MRS (Minimal Recursion Semantics) representations into OWL instances that are based on OWL classes, generated from the the type hierarchy of a deep linguistic grammar. The paper will identify parts of knowledge which can be translated from one formalism into the other without loosing information and parts which can only be approximated. The work described here is important for the Semantic Web to become a reality, since semantic annotations of natural language documents (DL) can be automatically generated by shallow and deep natural language parsing systems (FL).

1 Introduction and motivation

Ontologies on the one hand and resources for natural language processing (lingware) on the other hand, though closely related, are often maintained independently, thus constituting a duplication of work.

In the *first* part of this paper, we describe an implemented offline procedure that can be used to map concepts and instance information from ontologies to lingware resources for named entity recognition and information extraction systems. The approach (i) improves NE/IE precision and recall in closed domains, (ii) exploits linguistic knowledge for identifying ontology instances in texts more robustly, (iii) gives full access to ontology instances and concepts in natural language processing results, and (iv) avoids duplication of work in development and maintenance of ontologies and lingware. The advantages of this approach for Semantic Web and natural language (NL) processing-based applications come from a *cross-fertilization* effect. While ontology instance data can improve precision and recall of, e.g., named entity recognition (NER) and information extraction (IE) in closed domains, linguistic knowledge contained in NER and IE components can help to recognize ontology instances (or concepts) occurring in text, e.g., by taking into account inflection, anaphora, and context. Furthermore, (Haghighi and Klein, 2009) and others have shown that incorporating finer-grained semantic information on entities occurring in text (e.g., for antecedent filtering) helps to improve performance of coreference resolution systems.

If both resources would be managed jointly at a single place (in the ontology), they could be easily kept up-to-date and in sync, and their maintenance would be less time-consuming. When ontology concepts and instances are recognized in text, their name or ID can be used by applications to support subsequent queries, navigation, or inference in the ontology using an ontology query language (e.g., SPARQL). The procedure we describe here, preserves hierarchical concept information and links to ontology concepts and instances. Applications are, e.g., hybrid deep-shallow question answering (Frank et al., 2007), automatic typed hyperlinking (Busemann et al., 2003) of instances and concepts occurring in documents, or other innovative applications that combine Semantic Web and NL processing technologies, e.g., for semantic search (Schäfer et al., 2008).

The *second* part of this paper outlines the inverse transformation from feature logics (FL) into description logics (DL). Walking along this direction has the big advantage of potentially applying subsequent description logic reasoners to the lexical semantics of natural language input text in order to infer new knowledge, e.g., in interactive natural language ques-

*The work described in this paper has been carried out in the TAKE project (Technologies for Advanced Knowledge Extraction), funded by the German Federal Ministry of Education and Research under contract number 01IW08003.

Coling 2008: Poster Volume, pages 588–596,
Beijing, August 2010

tion answering. As an example, we will carefully develop the (approximate) translation of so-called robust minimal recursion semantic (RMRS) structures (Copestake, 2003) into OWL descriptions (McGuinness and van Harmelen, 2004). RMRS structures are the semantic output of various NL processing engines, encoded in typed feature structures (TFS). Since NL processors (e.g., taggers, chunkers, deep parsers) only build up structure, subsequent processing steps are either not realized or implemented in ad hoc way,

- dealing with merging & normalization of RMRS,
- infering new knowledge (e.g., w.r.t. the foregoing dialog),
- taking into account extralinguistic knowledge for reasoning.

Now, by moving from a specialized "designer language" (RMRS) to OWL, we can take advantage of years of solid theoretical and practical work in logic, especially in description logics. Since OWL is an instance of the description logics family and the defacto language for the Semantic Web, we can utilize the built-in reasoning capabilities of OWL and (rule-based) description logic reasoners.

The structure of this paper is as follows. In the next section, we outline the relationship between description logics and feature logics, trying to make clear what they have in common, but at the same time explaining their differences. Section 3 describes the syntactic mapping process from the ontology to feature structure descriptions. In Section 4, we present an example where recognized named entities enriched with ontology information are used in hybrid NL processing and subsequent applications. After that, Section 5 explains the mapping of RMRS structures into OWL descriptions. Finally, Section 6 shows that a subsequent description logic reasoner can utilize these descriptions to infer new knowledge.

2 The relationship between description and feature logics

Description logics (DL) (Baader et al., 2003) and *feature logics* (FL) (Carpenter, 1992) have been pursued independently for quite a while. Their close relationship was recognized by (Nebel and Smolka, 1990). Instances of both families of knowledge representation formalisms are usually decidable two-variable fragments of first-order predicate logic. Even though DL dialects usually have an intractable worst-case complexity, average-case reasoning is usually fast, due to the availability of highly-optimized tableaux reasoners. When adding seemingly easy constructs such as "role-value maps" (the analog to reentrancies), the underlying logical calculus becomes undecidable.

From an abstract viewpoint, both DL and FL employ unary and binary predicates for which the two communities invented different names (we only list some of them):

arity	description logic	feature logic
unary	concept, class	type, category
binary	role, property	feature, attribute

Though these names are different, both representation families (usually) vary in further, not so subtle details:

description logic	feature logic
open world assumption	closed world assumption
full Boolean concept logic	only conjunctions
relational properties	functional properties
role-value maps forbidden	reentrancies allowed

Let us be more verbose here to see the descriptional consequences of both approaches in terms of a mutual translation. We note here that we take OWL (McGuinness and van Harmelen, 2004) as an instance of DL and $\mathcal{TDL}$ (type description language) (Krieger and Schäfer, 1994) as an example of FL. OWL, the outcome of the DAML+OIL standardization, is regarded to be the de-facto language for the Semantic Web. OWL still makes use of constructs from RDF and RDFS, but restricts the expressive power of RDFS, thereby ensuring decidability of the standard inference problems. Compared to RDF(S), OWL provides more fine-grained modelling constructs, such as `intersectionOf` or `unionOf`.

Within the Head-Driven Phrase Structure Grammar (HPSG) (Pollard and Sag, 1994) paradigm in modern computational linguistics (CL), $\mathcal{TDL}$ is a language that has been employed in various implemented systems, such as PAGE, LKB, PET, or *SProUT*.

Before going into the details of our approximate transformation schema, let us quickly explain how to *atomize* a typed feature structure (TFS) in terms of description logic primitives, using OWL. Consider the following TFS which is a gross simplification of the *Head-Feature Principle* in HPSG. In terms of the "one-dimensional" line-based $\mathcal{TDL}$ notation, we write

```
phrase1 := phrase &
  [HEAD #h1, HEAD-DTR|HEAD #h1],
```

or as a two-dimensional AVM (attribute-value matrix) notation, we have

$$\text{phrase1} \equiv \left[\begin{array}{l} \textit{phrase} \\ \text{HEAD } \boxed{\text{h1}} \\ \text{HEAD-DTR|HEAD } \boxed{\text{h1}} \end{array}\right]$$

Assuming that this is an individual of class *phrase*, we can obtain a meaning-preserving OWL representation (we assume that HEAD and HEAD-DTR are functional OWL object properties):

```
<owl:Thing rdf:ID="h1"/>

<rdf:Description rdf:about="hdtr1">
  <rdf:type rdf:resource="owl:Thing"/>
  <HEAD rdf:resource="h1"/>
</rdf:Description>

<rdf:Description rdf:about="phr1">
  <rdf:type rdf:resource="phrase"/>
  <HEAD rdf:resource="h1"/>
  <HEAD-DTR rdf:resource="hdtr1"/>
</rdf:Description>
```

Note that only the top-level structure is explicitly typed (*phrase*); every other substructure thus is assigned the most general type, which translates into the OWL class `owl:Thing`. Note also the sharing of information under paths `HEAD` and `HEAD-DTR|HEAD`—this is realized by refering to the the name `h1` in the above RDF/OWL description for `phr1` and `hdtr1`.

Given a set of OWL descriptions, obtaining the inverse direction from DL to FL should now be clear. It is important here to group statements that are related to a specific class, viz., inheritance information (e.g., `intersectionOf`) together with property information about roles that are "introduced" on a given class (as given by the value of `rdfs:domain`). In Section 3, we focus on this inverse direction (DL-to-FL), whereas Section 5 exemplifies the FL-to-DL direction.

Let us finally elaborate fundamental differences between the DL and FL families that can only be approximated in terms of "less expressive" constructs.

Open vs. closed world assumption. Typed feature logics usually "live" in a closed world, meaning that if two types t_1 and t_2 do not share a common subtype (having a greatest lower bound), the unification (conjuntion) is assumed to be the bottom type (OWL: `owl:Nothing`), meaning that no individual exists which is of both t_1 and t_2 at the same time. This is totally different to the DL point of view: *what can not proven to be true* (whether the conjunction of t_1 and t_2 denotes the empty set) *is not believed to be false*. Thus we either have to introduce a new type t on the FL side, abbreviating the conjunction of t_1 and t_2 ($\mathcal{TDL}$: t := t_1 & t_2.), or to close the subclass hierarchy on the DL side: $\bot \equiv t_1 \sqcap t_2$ (OWL: `disjointWith`). This decision clearly depends on the direction of the transformation.

Boolean vs. conjunctive description logic. Typed feature logics rarely provide more than *conjunctions* of feature-value constraints. This is due to the fact that disjunctive descriptions render almost linear (conjunctive) unification exponential. A full Boolean calculus, such as OWL DL, even has an NEXPTIME complexity. Thus it is clear that the direction from DL to FL can only be approximated. The inverse direction is clearly trivial with the notable exception of *reentrancies* (see below).

To flesh out our point, consider the DL axiom *human* $\equiv$ *man* $\sqcup$ *woman* that fully determines ($\equiv$) *human* in terms of the union of the concepts *man* and *woman*. Given the syntax of $\mathcal{TDL}$, we can approximate parts of the intended meaning of the description by `man :< human` and `woman :< human`, since the above DL axiom entails that *man* $\sqsubseteq$ *human* and *woman* $\sqsubseteq$ *human* is the case. This is exactly specified by the above two $\mathcal{TDL}$ type definitions. Further, not so trivial approximations can be found in (Flickinger, 2002). The idea here is that foreseeable disjunctions of DL concepts can be emulated by introducing additional FL types (in the worst case, exponentially-many new types, however). Even negated concepts can be simulated this way, since FL lives in a closed world (see above).

Relational vs. functional properties. By default, roles in DL are relational properties, meaning that for a fixed individual in the domain of a given role, the number of individuals in the range needs not to be 0 or 1. DL further allows to impose *cardinality (or number) restrictions* on roles, so that we might write ≥ 0 *livingParents* $\sqcap \leq 2$ *livingParents* which says that one can have at least 0 and at most 2 living parents. This is in sharp contrast to FL which usually assume functional roles (so-called features), making such roles essentially partial functions. A partial workaround has been proposed in CL systems by using (ordered) difference lists to collect information. Other systems, such as *SProUT* (Krieger et al., 2004), come up with bags (or multisets) that even violate the foundational axiom (a set must not contain itself) in order to achieve runtime efficiency.

Summarizing, the FL-to-DL direction of translating features into roles is easy, since features in FL can be easily defined as functional roles in DL (OWL even provides the `owl:FunctionalProperty` characteristics). The inverse direction is only a gross approximation in that cardinality constraints can not be

stated on the FL side.

Role-value maps & reentrancies. The above Head-Feature Principle example seems to indicate that role-value maps can be easily represented in DL, simply by using the name of an individual to specify identity. In fact, this is true, but only for the ABox of a knowledge base, i.e., only for the set of individuals (or instances). However, the notion of role-value maps in DL or reentrencies in FL refers to the TBox and the set of concept definitions, resp. Thus, one can not intensionally specify identity of information for a potentially infinite number of individuals via a class axiom in DL, but needs to extensionally specify identity of information for each individual in the ABox.

3 OntoNERdIE: from OWL to $\mathcal{TDL}$

In this section, we describe an instantiation of the DL-to-FL mapping. OntoNERdIE is an offline procedure that maps ontology concept and instance information to lingware resources (Schäfer, 2006). The approach has been implemented for the language technology ontology that backs up the LT World web portal (http://www.lt-world.org), but can be easily adapted to other domains and ontologies, since it is fully automated, except for the choice of relevant main concepts and properties that are going to be mapped which is a matter of configuration.

The target named entity recognition and information extraction tool we employ here is *SProUT* (Drożdżyński et al., 2004), a shallow multilingual, multi-purpose NL processor. The advantage of *SProUT* in the described approach for named entity recognition and information extraction is that it comes with (1) a type system and typed feature structures as the basic data type, (2) a powerful, declarative rule mechanism with regular expressions over typed feature structures, and (3) a highly efficient gazetteer module with fine-grained, customizable classification of recognized entities.

SProUT provides additional modules such as morphology or a reference resolver that can be exploited in the rule system, e.g., to use context or morphological variation for improved NER. Through automatically generated mappings, *SProUT* output enriched with ontology information can be used for robust, hybrid deep-shallow parsing, and semantic analysis.

In this section, we describe the offline processing steps of the OntoNERdIE approach. The online part in applications is described in Section 4. The approach heavily relies on XSLT transformations (Clark, 1999) of the XML representation formats, both in the offline mapping and in the online application.

3.1 RDF preprocessing

Input to the mapping procedure is an OWL ontology file, containing both concept and instance descriptions. The RDF file is pre-processed with a generic XSLT stylesheet sorting and merging `rdf:Description`s that are distributed over the file but which belong together. We use XSLT's `key` and `generate-id` functions. Depending on the application, the next two processing stages take a list of concepts as filter because it will typically not be desirable to extract all concepts or instances available in the ontology. In both cases, resource files are generated as output that can be used to extend existing named entity recognition resources. E.g., while general rules can recognize domain-independent named entities (e.g., any person name), the extended resource contains specific, and potentially more detailed information for domain-specific entities.

3.2 Extracting inheritance

The second stylesheet converts RDFS `subClassOf` statements from output step 1 (Section 3.1) into a set of $\mathcal{TDL}$ type definitions that can be immediately imported by the *SProUT* named entity recognition grammar. Currently 1,260 type definitions for the same number of `subClassOf` statements in the LT World ontology are generated, e.g.,

```
NL_Parsing := Written_Language &
  Language_Analysis.
```

This is of course a lossy conversion because not all relations supported in an OWL ontology (such as `unionOf`, `disjointWith`, `intersectionOf`) are mapped. However, we think that for NE classifications, the `subClassOf` taxonomy mappings will be sufficient. Other relations could be formulated as direct (though slower) ontology queries using the `OBJID` mechanism described in the next step. If the target of OntoNERdIE is a NER system different from *SProUT* and without a type hierarchy, this step can be omitted. The `subClassOf` information can always be gained by querying the ontology appropriately on the basis of the concept name.

3.3 Generating gazetteer entries

The next stylesheet selects statements about instances of relevant concepts via the `rdf:type` information and converts them to structured gazetteer source files for the *SProUT* gazetteer compiler (or into a different format in case of another NER system). In the following example, one of the approximately 20,000 converted entries for LT World is shown.

```
Bernd Kiefer | GTYPE: lt_person |
  SNAME: "Kiefer" | GNAME: "Bernd" |
  CONCEPT: Active_Person |
  OBJID: "obj_62893"
```

The attribute CONCEPT contains a $\mathcal{TDL}$ type generated in step 2 (described in Section 3.2). For convenience, several ontology concepts are mapped (defined manually as part of the configuration of the stylesheet) to only a few named entity classes (under attribute GTYPE). For the LT World ontology, these classes are person, organization, event, project, product, and technology. The advantage of this simplification is that NER context rules from existing *SProUT* named entity grammars can be re-used for improved robustness and disambiguation.

The rules, e.g., recognize name variants with title like Prof. Kiefer, Dr. Kiefer, or Mr. Kiefer with or without a first name. Moreover, context (e.g., prepositions with location names, verbs), morphology and reference resolution information can be exploited in these rules.

The following *SProUT* rule lt-event (extended $\mathcal{TDL}$ syntax) simply copies the slots of a matched gazetteer entry for events (e.g., a conference) to the output as a recognized named entity.

```
lt-event :> gazetteer &
  [GTYPE lt_event, SURFACE #name,
   CONCEPT #concept, OBJID #objid,
   GABBID #abbrev]
  ->
  ne-event & [EVENTNAME #name,
    CONCEPT #concept, OBJID #objid,
    GABBID #abbrev].
```

OBJID contains the object identifier of the instance in the ontology. It can be used as a link back to the full knowledge stored in the ontology, e.g., for subsequent queries, like *Who else participated in project* [*with OBJID obj_4789*]*?*.

In case multiple instances with same names but different object IDs occur in the ontology (which actually happens to be the case in LT World), multiple alternatives are generated as output which is probably the expected and desired behavior (e.g., for frequent names such as John Smith). On the other hand, if product or event names with an abbreviated variant exist in the ontology, they both point to the same object ID (provided they are stored appropriately in the ontology).

4 Application to hybrid deep-shallow parsing

We now describe and exemplify how the named entities enriched with ontology information are employed in a robust, hybrid deep-shallow architecture, combining domain-specific shallow named entity recognition with deep, broad-coverage, domain-independent, unification-based parsing for generating a semantic representation of the meaning of parsed sentences. An application of this scenario is deep question analysis for question answering of structured knowledge sources, encoded as an OWL ontology (Frank et al., 2007).

The output of *SProUT* for a recognized named entity is a typed feature structure in XML containing the instantiated RHS of the recognition rule as shown in step 3 (Section 3.3) with the copied structured gazetteer data, plus some additional information like character span, named entity type, etc. The mapping of recognized named entities to generic lexicon entries of the deep grammar, in this case the English Resource Grammar (Flickinger, 2002), for hybrid processing are performed through an XSLT stylesheet, automatically generated from the *SProUT* type hierarchy. Analogous mappings are supported for other grammars available in the DELPH-IN repository (see http://www.delph-in.net). The mapping basically transports the surface string, a character span, and a generic lexicon type of the deep grammar for a chart item to be generated in an XML format, readable by the deep parser. A sample output of the semantic representation generated by the deep parser is shown in Figure 1. The semantic representation format, called RMRS, is described in (Copestake, 2003) and in Section 5.3 below.

In addition to the basic named entity type mapping for default lexicon entries, the recognized concepts are also useful for constraining the semantic sort in the deep grammar in a more fine-grained way (e.g., for disambiguation). The deep parser's XML input chart format foresees "injection" of such types into deep structures. Here, OBJID and other structured information, like given name and surname, can be preserved in the representation. The advantage of the RMRS format is that it can also be combined *ex post* with analyses from other deep or shallow NLP components, e.g., with partial analyses when a full parse fails.

5 (R)MRS2OWL: from $\mathcal{TDL}$ to OWL

This section is devoted to the translation of MRSs which are encoded as TFSs into a set of OWL expressions. An example of a variant of MRS, a so-called robust MRS (RMRS) has already been depicted in Figure 1. RMRS will be explained in more detail in Section 5.3.

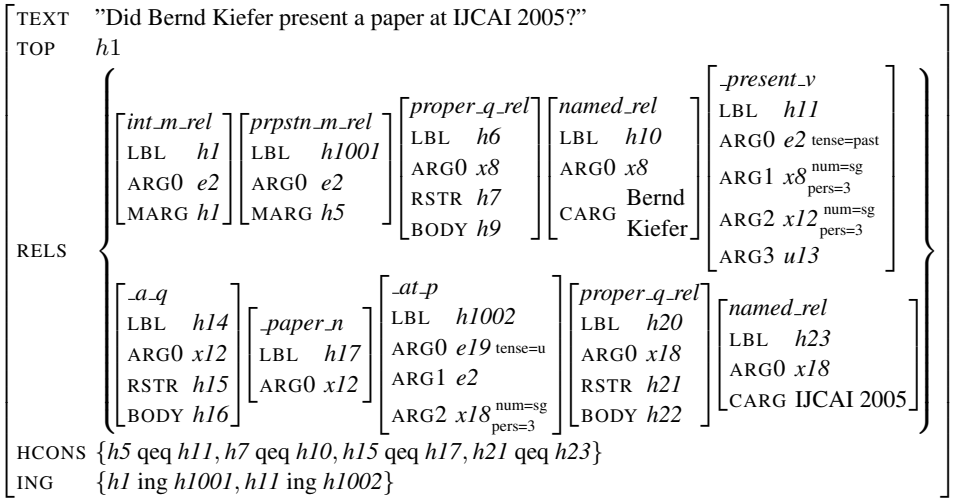


Figure 1: RMRS generated through hybrid parsing.

5.1 Some words on MRSs

There exist good linguistic reasons for assuming that the semantics of a sentence like *Kim ate a cookie* is not $past(eat(kim', cookie'))$, but instead something like $\exists e \ . \ eating(e) \wedge subject(e, kim') \wedge object(e, cookie') \wedge before(e, now)$. This approach to NL semantics is often called *Event* or *Davidsonian* semantics (named after the American philosopher Donald Davidson). HPSG has incorporated ideas from event semantics by defining so-called Minimal Recursion Semantics (MRS) structures (Copestake et al., 2005) that are constructed in parallel with the syntactic structure. MRS as such provides a flat compositional semantics and maximizes splitting using equality constraints. Structural ambiguities, as can be found in the famous sentence *Every farmer who owns a donkey beats it*, are not spelled out, but instead quantifier scope is underspecified. By imposing constraints on the scope, specific analysis trees can be reconstructed. Robust MRS (RMRS) (Copestake, 2003), derived from MRS, was designed as an abstract language that supports the integration of partial and total analysis results from deep and shallow processors and provides a good tradeoff between robustness and accuracy (see (Frank et al., 2004) for an example).

5.2 Why the translation is useful

NL processors (e.g., tokenizer, POS tagger, shallow chunk parser, deep parser, etc.) that are geared towards (R)MRSs (or another common language) have the potential of combining their output on the level of semantics. However, these engines do *not* provide any form of reasoning, i.e., they only build up structure.

Consider, for instance, a deep unification-based parser that might return analyses represented as typed feature structures, where both syntax and semantics (the MRS) has been constructed with the help of unification. Now, to bring structures together and to perform deductive and abductive forms of reasoning, subsequent computational steps are necessary, but these steps strictly go beyond the power of ordinary parsing.

In order to perform these subsequent steps, we need a concrete implemented (and hopefully standardized) representation language for which editing, displaying, and reasoning tools are available. Exactly OWL accomplishes these requirements. Hence we think that the described below translation process from (R)MRSs into OWL is worthwhile, especially when one is interested in interfacing linguistic knowledge (the (R)MRSs) with extralinguistic ontologies for specific domains.

5.3 The translation process

In order to explain the translation process, we will analyze the RMRS depicted in Figure 1. The RMRS was derived from the MRS of the deep unification-based parser. We see that an RMRS contains four distinguished attributes (the `TEXT` attribute is only added for illustration):

1. `TOP`: a handle (pointer) to the top-level structure.

2. `RELS` (relations): a set of so-called *elementary predications* (EP), encoded as TFSs, each expressing an atomic semantic unit that can not be

further decomposed; due to the lack of sets, TFS grammars use a list here.

3. HCONS (handle constraints): a set of so-called *qeq constraints* (equality modulo quantifiers); the left side of a qeq constraints (a handle h in an argument position) is always related to a label l of an EP, (i) either directly ($h = l$) or (ii) indirectly, in case h dominates a quantifier q, such that $\texttt{BODY}(q) = l$ or again another quantifier, where condition (ii) is recursively applied again.

4. ING (in group): a set of relations used to express a conjunction of EPs from the set RELS.

Giving this information, it should now be clear that the TFS from Figure 1 must be realized as an instance of the OWL class RMRS and that the features TOP and RELS must be implemented as roles in OWL, all defined on RMRS through the use of rdfs:domain:

```
<owl:Class rdf:ID="RMRS"/>
<owl:ObjectProperty rdf:ID="TOP">
  <rdf:type rdf:resource=
     "&owl;FunctionalProperty"/>
  <rdfs:domain rdf:resource="#RMRS"/>
  <rdfs:range rdf:resource=
     "#HandleVar"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="RELS">
  <rdfs:domain rdf:resource="#RMRS"/>
  <rdfs:range rdf:resource="#EP"/>
</owl:ObjectProperty>
```

TOP takes exactly one argument, hence we use OWL's FunctionalProperty characteristics mechanism here. Since RELS (as well as HCONS and ING, see below) might take more than one argument, we do not impose a property restriction here, so they are relational by default. TOP maps to a special variable class (see below), and RELS to EPs.

TOP. The TOP property always takes a handle variable; other variable classes, such as label vars are used for restricting properties:

```
<owl:Class rdf:ID="Var"/>
<owl:Class rdf:ID="HandleVar">
  <rdfs:subClassOf rdf:resource=
     "#Var"/>
</owl:Class>
<owl:Class rdf:ID="LabelVar">
  <rdfs:subClassOf rdf:resource=
     "#Var"/>
</owl:Class>
```

Actually, this modelling is mere window-dressing and clearly verbose, since an OWL instance of class RMRS is always assigned a name (<RMRS rdf:ID="...">), and in fact, this name can be taken to be the TOP handle. This means that we can in principle forbear from the TOP property. However, if we want to utilize morpho-syntactical information in subsequent inference steps, we have to enrich the above variable classes with further properties/roles, such as tense, pers, or num (see, e.g., the "structured" variables in the structure for _present_v in Figure 1).

RELS. Elements of RELS, i.e., concrete EPs are essentially "slimed" instances of feature structure types. Overall, this means that we have to represent the relevant types of the linguistic type hierarchy and their subsumption relationship as OWL classes. As shown in Section 3, this process can be automated and only some guidance from a knowledge engineer is necessary to mark the features that should *not* be taken over to the DL side.

HCONS and ING. HCONS essentially specifies a ternary relation, but since OWL (and DL in general) are restricted to unary and binary relations, one way to model a qeq constraint is to define a binary property, consisting of a left-hand and a right-hand side. From what has been said above, the left-hand side is a handle and the right-hand side a label, hence we have the following declaration for qeq:

```
<owl:ObjectProperty rdf:ID="qeq">
  <rdfs:domain rdf:resource=
     "#HandleVar"/>
  <rdfs:range rdf:resource=
     "#LabelVar"/>
</owl:ObjectProperty>
```

Given this way of modelling, it is now *impossible* to define a property HCONS (as well as ING) on class RMRS, since properties can only take instances of classes, but not instances of other *properties*. However, since we assume that our variables (instances of class Var) are always unique at runtime, it is in principle not necessary to group the qeq constraint *inside* an (R)MRS—note that there is still a connection between EPs and qeq constraints through the use of variables. However, if we want to talk about/want to access the qeq constraints of a specific (R)MRS instance directly, this kind of modelling is somewhat unhandy.

To overcome this seemingly wrong representation (we are neutral about this), we have to "reify" or "wrap" qeq property instances. This would mean that qeq would no longer be a property, but instead becomes a class, say QEQ, consisting of a right-hand and a left-hand side. With this in mind, we can easily model, e.g., the first qeq constraint qeq1 from the above figure:

```
<RMRS rdf:ID="rmrs1">
  <TOP  rdf:resource="#h1"/>
  <RELS rdf:resource="#ep1"/>
  <HCONS rdf:resource="#qeq1"/>
```

```
  ...
</RMRS>
<QEQ rdf:ID="qeq1">
  <LHS rdf:resource="#h5"/>
  <RHS rdf:resource="#h11"/>
</QEQ>
<HandleVar rdf:ID="h1"/>
<HandleVar rdf:ID="h5"/>
<LabelVar rdf:ID="h11"/>
<int_m_rel rdf:ID="ep1">
  <LBL rdf:resource="#h1"/>
  <ARG0 rdf:resource="#e2"/>
  <MARG rdf:resource="#h1"/>
</int_m_rel>
```

What we have said about qeq constraints so far do hold for in-group constraints as well.

6 DL reasoning: a small example

We have already said that the OWL representation of RMRS structures are a good starting point to implement some useful forms of reasoning. Consider the sentence *Did Bernd Kiefer present a paper at IJCAI 2005?* from Figure 1. From the resulting EPs and with the help of an in-group constraint, we can infer the fact that Bernd Kiefer was (physically) at IJCAI 2005, assuming he has presented a paper (which he did). The inference rule achieving this can be stated informally as *presenting a paper at a conference entails being at the conference*. A more formal representation in terms of feature structures is given in Figure 2.

Clearly this rule can be rewritten to operatate on OWL expressions (as is proposed in SWRL (Horrocks et al., 2004)) or on the underlying RDF triple notation (which, for instance, OWLIM (Kiryakov, 2006) assumes). Note the use of logical variables in the above rule in order to formulate the transport of information from the LHS to the RHS. The above rule abstract away from concrete persons and locations through the use of logic variables $?p$ and $?l$. Note further that the resulting RHS output structure is no longer a RMRS but a domain-specific representation (somewhat simplified in this example) that can be queried for or can be employed in subsequent reasoning tasks.

In (Frank et al., 2007), an implemented approach is described that utilizes an *additional* frame representation layer (Ruppenhofer et al., 2006) in which rules of the above kind are applied, using the term rewriting system of (Crouch, 2005).

7 Summary

Our paper returned to mind that there exists a close relationship between feature logics as used in computational linguistics and description logics employed in the Semantic Web community. This relationship can be utilized to obtain more and better semantic annotations through information extraction and deep parsing of text documents. We have indicated that specific language constructs in FL and DL can be mutally transformed without losing any meaning, whereas others can only be approximated (esp., role-value maps/reentrencies and functional features/relational roles).

We have described an implemented procedure that maps ontology instances and concepts to named entity recognition and information extraction resources. As argued in the paper, the benefits for minimized domain-specific and linguistic knowledge engineering are manifold. An application using hybrid shallow and deep NL processing on the basis of the mapped ontology data has been successfully implemented for question answering. This application (Frank et al., 2007) employs an additional frame semantics layer (cf. Section 6) on which light forms of reasoning take place. In order to make this additional layer superflous, we have described a transformation scheme that maps (R)MRS into OWL descriptions. Given these descriptions, rules of the above kind (Section 6) can directly operate on OWL, and no additional translation is necessary to query the instance data, encoded in RDF/OWL.

References

Baader, Franz, Diego Calvanese, Deborah McGuinness, Daniele Nardi, and Peter Patel-Schneider. 2003. *The Description Logic Handbook*. Cambridge University Press, Cambridge.

Busemann, Stephan, Witold Drożdżyński, Hans-Ulrich Krieger, Jakub Piskorski, Ulrich Schäfer, Hans Uszkoreit, and Feiyu Xu. 2003. Integrating Information Extraction and Automatic Hyperlinking. In *Proceedings of the Interactive Posters/Demonstration at ACL-03*, pages 117–120.

Carpenter, Bob. 1992. *The Logic of Typed Feature Structures*. Tracts in Theoretical Computer Science. Cambridge University Press, Cambridge.

Clark, James, 1999. *XSL Transformations (XSLT)*. W3C, http://w3c.org/TR/xslt.

Copestake, Ann, Dan Flickinger, Ivan A. Sag, and Carl Pollard. 2005. Minimal recursion semantics: An introduction. *Research on Language and Computation*, 3(4):281–332, 12. DOI 10.1007/s11168-006-6327-9.

Copestake, Ann. 2003. Report on the Design of RMRS. Technical Report D1.1b, University of Cambridge, Cambridge, UK.

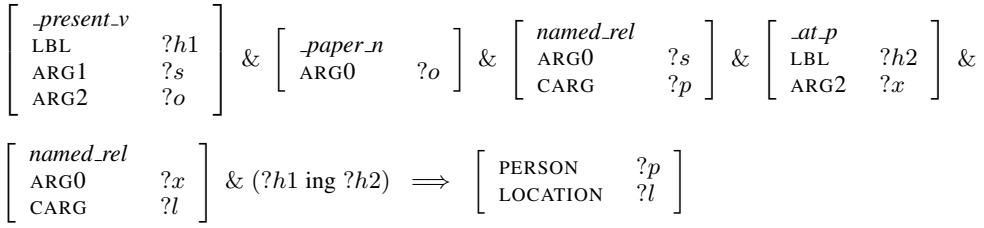


Figure 2: RMRS rule over EPs and in-group constraint.

Crouch, Richard. 2005. Packed rewriting for mapping semantics to KR. In *Proceedings of the International Workshop on Computational Semantics (IWCS) 6, Tilburg*.

Drożdżyński, Witold, Hans-Ulrich Krieger, Jakub Piskorski, Ulrich Schäfer, and Feiyu Xu. 2004. Shallow Processing with Unification and Typed Feature Structures—Foundations and Applications. *KI*, 04(1):17–23.

Flickinger, Dan. 2002. On building a more efficient grammar by exploiting types. In Oepen, S. D. Flickinger, J. Tsuji, and H. Uszkoreit, editors, *Collaborative Language Engineering. A Case Study in Efficient Grammar-based Processing*, pages 1–17. CSLI Publications.

Frank, Anette, Kathrin Spreyer, Witold Drożdżyński, Hans-Ulrich Krieger, and Ulrich Schäfer. 2004. Constraint-Based RMRS Construction from Shallow Grammars. In Müller, Stefan, editor, *Proceedings of the HPSG04 Conference Workshop on Semantics in Grammar Engineering*, pages 393–413. CSLI Publications, Stanford, CA.

Frank, Anette, Hans-Ulrich Krieger, Feiyu Xu, Hans Uszkoreit, Berthold Crysmann, and Ulrich Schäfer. 2007. Question answering from structured knowledge sources. *Journal of Applied Logics, Special Issue on Questions and Answers: Theoretical and Applied Perspectives*, 5(1):20–48.

Haghighi, Aria and Dan Klein. 2009. Simple coreference resolution with rich syntactic and semantic features. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1152–1161.

Horrocks, Ian, Peter F. Patel-Schneider, Harold Boley, Said Tabet, Benjamin Grosof, and Mike Dean. 2004. SWRL: A semantic web rule language combining OWL and RuleML. W3C Member Submission.

Kiryakov, Atanas. 2006. OWLIM: balancing between scalable repository and light-weight reasoner. Presentation of the Developer's Track of WWW2006.

Krieger, Hans-Ulrich and Ulrich Schäfer. 1994. *TDL*—A Type Description Language for Constraint-Based Grammars. In *Proceedings of the 15th International Conference on Computational Linguistics, COLING-94*, pages 893–899.

Krieger, Hans-Ulrich, Witold Drożdżyński, Jakub Piskorski, Ulrich Schäfer, and Feiyu Xu. 2004. A Bag of Useful Techniques for Unification-Based Finite-State Transducers. In *Proceedings of KONVENS 2004*, pages 105–112.

McGuinness, Deborah L. and Frank van Harmelen. 2004. OWL Web Ontology Language Overview. Technical report, W3C. 10 February.

Nebel, Bernhard and Gert Smolka. 1990. Representation and reasoning with attributive descriptions. In Bläsius, K.-H., U. Hedtstück, and C.-R. Rollinger, editors, *Sorts and Types in Artificial Intelligence*, pages 112–139. Springer, Berlin. Also available as IWBS Report 81, IBM Germany, September 1989.

Pollard, Carl and Ivan A. Sag. 1994. *Head-Driven Phrase Structure Grammar*. Studies in Contemporary Linguistics. University of Chicago Press, Chicago.

Ruppenhofer, Josef, Michael Ellsworth, Miriam R.L. Petruck, Christopher R. Johnson, and Jan Scheffczyk. 2006. FrameNet II: extended theory and practice. Technical report, International Computer Science Institute (ICSI), University of California, Berkley. http://framenet.icsi.berkley.edu/book/book.pdf.

Schäfer, Ulrich, Hans Uszkoreit, Christian Federmann, Torsten Marek, and Yajing Zhang. 2008. Extracting and querying relations in scientific papers on language technology. In *Proceedings of LREC-2008*.

Schäfer, Ulrich. 2006. OntoNERdIE – mapping and linking ontologies to named entity recognition and information extraction resources. In *Proceedings of the 5th International Conference on Language Resources and Evaluation LREC-2006*, pages 1756–1761, Genoa, Italy.

Generating Simulated Relevance Feedback: A Prognostic Search approach

Nithin Kumar M and **Vasudeva Varma**
Search and Information Extraction Lab,
International Institute of Information Technology Hyderabad,
nithin_m@research.iiit.ac.in and vv@iiit.ac.in

Abstract

Implicit relevance feedback has proved to be a important resource in improving search accuracy and personalization. However, researchers who rely on feedback data for testing their algorithms or other personalization related problems are loomed with problems like unavailability of data, staling up of data and so on. Given these problems, we are motivated towards creating a synthetic user relevance feedback data, based on insights from query log analysis. We call this simulated feedback. We believe that simulated feedback can be immensely beneficial to web search engine and personalization research communities by greatly reducing efforts involved in collecting user feedback. The benefits from "Simulated feedback" are - it is easy to obtain and also the process of obtaining the feedback data is repeatable, customizable and does not need the interactions of the user. In this paper, we describe a simple yet effective approach for creating simulated feedback. We have evaluated our system using the clickthrough data of the users and achieved 77% accuracy in generating click-through data.

1 Introduction

Implicit relevance feedback serves as a great source of information about user behaviour and search context. A lot of research went through in the recent past in making use of this great pool of information. Relevance feedback is proven to significantly improve retrieval performance (Harman, 1992; Salton and Buckley, 1990). It has also been successfully used to improve searching ranking, query expansion, personalization, user profiling et cetera (Steve Fox et al.,, 2005; Rocchio, 1999; Xuehua et al., 2005).

Clickthrough data is the most prevalent form of implicit feedback used by researchers for personalization purposes. Click log data provides valuable information about the interests, preferences and semantic search intent of the user (Daniel and Levinson, 2004; Kelly and Belkin, 2001). Unlike explicit feedback, clicks logs do not require any special effort from the user (Rocchio, 1999). It is collected in the background while the user interacts with the search engine to quench his information need. Hence, it is easy and feasible to collect large amounts of clickthrough data.

However, using clickthrough data has its own share of problems. Firstly, it is not available for public or even research communities at large for reasons like being a potential threat to privacy of web users. Secondly, it only contains the URLs of the results that the user clicked and does not contain the documents that the user has chosen. Given the dynamic nature of the web, content of many of the urls is prone to change and in some cases it might not exist. In other cases, even if the old expected results remain good resources, search engines might not retrieve them in response to queries. It will return near-duplicate pages that have equivalent content but different URLs. Thus feedback data may rapidly become stale with new pages replacing old ones as more approporiate resources. And also, given the rapidly changing ranking algorithms of web search engines, feed-

Coling 2008: Poster Volume, pages 597–604,
Beijing, August 2010

back data collected from the users becomes outdated. Hence researchers who rely on feedback data either for testing their algorithms or other personalization related problems are faced with the problems of non-availability of user feedback data.

In this paper, we strive to address the above problems by generating simulated relevance feedback using prognostic search techniques. *Prognostic search* is a process of simulating user's search process and emulating their actions, through preferences captured in their profile. Such generated feedback can be used for research in personalization techniques and analyzing personalization algorithms and search ranking functions(Harman, 1988). The main advantage with this system is that we can create data on the fly and hence not fear of it becoming stale. Since it does not involve user's actions, it is feasible to generate large amounts of data in this way.

2 Contributions and Organization

In this paper, we propose a novel way of creating simulated feedback. The data thus produced can be used for evaluating/training personalization systems. Using our proposed method, given a user's training data, we can produce synthetic implicit feedback data - simulated feedback data on the fly. We also propose a novel user browsing model which extends the high performing cascade model of (Craswell et al., 2008). Our *Patience* parameter can be used to build more complex user browsing models to bring the whole process of generating implicit feedback data a step nearer to the real world mechanisms.

In section 3, we describe our approach to generate simulated feedback data. In sections 3.2.3 and 3.2.4, we describe the process of browsing results and generating clicks which form the crux of our approach. We evaluate our system and prove the usefulness of it in section 4. Section 5 and 6 give an account of our experiments and the study of works related to ours already present in the literature. We conclude that our proposed approach can be highly useful in personalization research and give an account of our future directions in section 7.

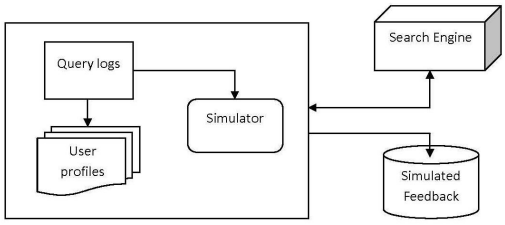


Figure 1: System architecture

3 Proposed Approach

Simulated feedback is a new type of feedback similar to implicit and explicit relevance feedback. Simulated feedback is created by observing and analyzing real world search log data. We propose a two phase process to create simulated relevance feedback as follows: In phase 1, we process real world click-through data of a search engine and build user profiles using the data. In phase 2, we simulate a user's search process and emulate their actions based on their profile. We call this process as "*Prognostic Search*".

3.1 Creating Profiles

After closely examining and analyzing the semantics of the query log, we have chosen the following parameters to characterize a user: an anonymous user-id, *perceived relevance threshold*, *patience*, previous queries issued and search history of the user.

A user-id is used to distinguish and uniquely identify each and every user. *Perceived relevance* is the relevance estimate of the result according to the user on examining the title, snippet and the url of the result. And *Perceived relevance threshold* is the threshold limit of *perceived relevance* of a result for the user to click it. *Patience* of the user is the trait which determines the number of clicks and the depth to which the user examines the results. We explain the process of computing a user's patience parameter in detail in section 3.2.3. We stored the previous queries and clicks of the user to capture the preferences of the user.

To make use of the search history, we used the previous queries issued and previous results clicked by the user. We store the titles and snippets of those results to capture the interests of the

user. Here, our aim is to generate implicit relevance feedback which is very close to the real world data. To generate synthetic relevance feedback, we instantiate these parameters with appropriate values using real world data.

3.2 Prognostic Search

Prognostic search is simulation of a user's search process and emulating their actions based on their interests and preferences captured in their user profile. Simulating search process involves four steps viz., i)Query formulation, ii)Searching, iii)Browsing results and iv)Generating Clicks. Each of these processes are explained below.

3.2.1 Query Formulation

Query formulation involves cognitive process of the user and requires background knowledge about the user like their interests, preferences and their knowledge base. It is highly impossible to capture the cognitive thought process of a user and emulate their method of generating a query. To solve this problem, we randomly select a search session from a user's history and send all the queries in it sequentially to the search engine. This helps us to preserve the inter query relations that naturally exist between the subsequent queries in a session.

3.2.2 Searching

This step involves retrieving documents relevant to the query generated in the previous step. We used yahoo search engine which is very much similar to the search engine from which the training data is collected.

3.2.3 Browsing results

In this step, we simulate the manner in which a user browses the results in the real world. Based on the observations in (Granka et al., 2004; Filip and Joachims, 2005), we assume that the user in the real world follows the browsing model explained in Algorithm 1. In real world, a user may follow more complex browsing models, but presently we have considered this browsing model to simplify things.

Accordingly, to simulate the browsing process of the user explained in algorithm 1, we followed the Algorithm 2.

Algorithm 1 User browsing model in real world

Step1: Start browsing with the top-most result.
Step2: Examine title, snippet and URL of the result.
Step3: Click if the result looks promising.
Step4: If(user has patience) go to step 5, else go to step 6.
Step5: Select next result and go to step 2.
Step6: Start examining the clicked results.
Step7: If(information need satisfied) end the process, else go to step 8.
Step8: Reformulate the query and go to step 1.

Algorithm 2 Simulated User browsing model

Step 1: Determine the number of results to be browsed based on *patience* parameter.
Step 2: Browse the results in increasing order of their ranks and examine them.
Step 3: Compute the perceived relevance score of the results.
Step 4: In the same order, generate clicks based on the perceived relevance scores of the results.
Step 5: If(session has more queries) go to step 6, else end the process.
Step 6: Select next query in the session and go to step 1.

Thus based on the *patience* parameter, we determine the number of results that the user browses. In our analysis of query log parameters, we learned that the *patience* value of a user can be characterized by the following parameters: number of clicks per session, maximum rank of the result clicked in a session, time spent in a session, the number of queries issued per second and the average semantic relevance of the top ten results of that session to the user. We found out that the patience of the user is directly proportional to the maximum rank of the result he has clicked in a session. We also found out that the number of clicks a user generates is inversely proportional to the number of queries he issues per second and directly proportional to the amount of time he spends per session. Thus, a user with

more *patience* tends to examine more search results and thus generate more clicks based on their relevance. We explain these dependencies in detail in the experiments section. So in order to learn the Patience parameter of the user, we devised the following formula:

$$\text{Patience} = \alpha \times \frac{(MR \times T \times C \times S_{q_i})}{Q} \quad (1)$$

Here MR denotes the average of maximum rank of the results clicked by the user in a session, T denotes the average time spent in a session, C is the average number of clicks in a session and Q denotes the average number of queries issued per session and S_{q_i} is the average semantic distance of the top ten results of the query 'q_i'. Here, "α'' is an equalization constant.

3.2.4 Generating clicks

This is the most important step in our simulation process. Typically, a user observes the visual information viz., title, snippet and the URL of a result(Joachims et al., 2005). Then based on their interests, they choose the results relevant to them. Similarly, we closely examine the results selected in the previous step and then score them according to their relevance to the user. We consider the title, snippet and the page-rank of the result and determine its relevance to the user known as *perceived relevance* score.

We first compute the semantic distance between the title and snippet of the present result from the titles and snippets of previously clicked results of the user. The results already clicked by the user serve as a knowledge base of the interests and preferences of the user. Thus, the semantic distance between the present result and the previous result gives us an account of the relevance that the present result carries to the user.

We used latent semantic analysis (LSA) to compute the semantic distance between the results. LSA does not take the dictionary meaning of the words as input; it rather extracts the contextual meaning of the word with respect to all other words in semantic space(Landauer et al., 2007). This property of LSA is very much useful in the present context. A particular word may have a lot of meanings but we are concerned about only those meanings of the word which the user interprets, which are captured in the sentences present in the user's click history. Hence, we used LSA to compute the semantic distance between the results.

We also consider the page-rank of the result, which has proven to be an important factor in making the decision of a click. In our study, we found that for about 89% of the queries with clicks, the top ranked document has been clicked and for 56% of the queries second ranked document has been clicked. In Figure 3, we show the click ratio for each of the top ten ranked documents[1]. Thereby, we derive that the rank of the result is also a very important factor in deciding whether a result has to be clicked or not. We also consider the distance of the present result from the previous click of the user. In (Joachims et al., 2005), it is shown that the user is more biased to click the result that immediately follows the result he previously clicked. In our simulation process, if this distance for any result exceeds 10, then we terminate the browsing process and reformulate the query. We believe that when this distance exceeds 10, it signifies that the quality of the results is low and hence can be ignored.

We used the bayesian probabilistic techniques to calculate the probability of the user clicking a result based on the above discussed factors. Hence *Click* being a Bernoulli variable, we have

$$P(c/R, q, u) = \alpha_{R,q,u}^{c} \left(1 - \alpha_{R,q,u}\right)^{1-c} \quad (2)$$

Where $\alpha_{R,q,u}$ is the probability that user 'u' clicks the result 'R' for a query 'q'. We model the probability of a click, $P\left(c/R, q, u\right)$ as a joint probability of P(c,r,Rel,D) where 'r' denotes the rank of the result, 'Rel' denotes the semantic relevance score of the result to the user – precisely to his previous clicks – and 'D' denotes the distance of the previous click of the user. We use this probability of the result as the *Perceived relevance* score of the result. Thus, we have:

[1] In figure 3, we have normalized the clicks statistics with the number of clicks for top ranked document. So, the click-ratio for the top ranked document will be 1.

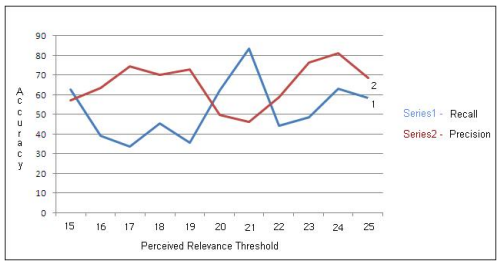


Figure 2: Graph showing Precision and Recall of generating clicks for a particular user

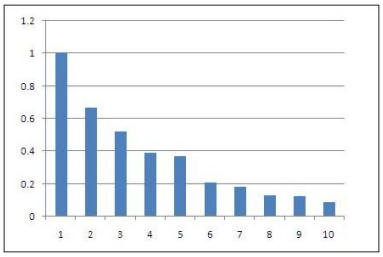


Figure 3: Ranks Vs Clicks-ratio

$$\text{Perceived relevance} = P\left(c/R, q, u\right) = P\left(c/r, Rel, D\right) \propto ln\left[P\left(r/c\right)\right] + ln\left[P\left(Rel/c\right)\right] + ln\left[P\left(D/c\right)\right] + ln\left[P\left(c_{i+1}\right)\right] \quad (3)$$

Here, 'r' denotes the rank of the result, 'Rel' denotes the *perceived relevance* of the result to the user and 'D' denotes the distance of the result from the user's previous clicked result. Prior probablities of each of these factors are calculated from the data stored in the user profile. We used Laplace smoothing techniques to deal with zero probability entries. $P(c_{i+1})$ is the probability that the user may click a result after clicking 'i' results. We also believe that the behaviour of the user changes with each click he generates in a session. Hence we used the factor $P(c_{i+1})$ in determining the probability of the click[2]. Then, we compare this score with the *Perceived relevance threshold* of the user and generate the clicks accordingly.

Computing *Perceived Relevance Threshold*: Using the above formula, we generated clicks for different values of *Perceived Relevance Threshold* for a user. Figure 2 show the precision and recall values of generating clicks for different values of Perceived Relevance Threshold of a user. Thus, we plot the accuracy of our system for different values of *Patience Relevance Threshold* and accordingly set the threshold selecting the best values for precision and recall of the system.

4 Experiments

Clickthrough data is a valuable source of user information. In our statistical analysis of clickthrough data, we have found that the page-rank of a result can highly influence the user to make a click which can be seen in figure 3.

In our definition of *Patience*, we termed it as parameter to denote the depth to which the user examines the results and the number of clicks he generates. In equation 1, we show that the patience value is inversely proportinal to the number of queries the user issues in a session. To prove this fact, we made a statistical analysis on the real world querlogs[3]. From the graphs shown in figure 4, it can be clearly seen that the *Patience* of the user is inversely proportional to the user's number of Queries/sec. These graphs show the influence of the factor Queries/sec on the number of clicks the user generates for a query and the maximum rank clicked by the user in a session. We drew the graphs averaging the different queries/sec value of a user in a session for each value of MR and number of clicks respectively. It is evident that both the graphs are weakly decreasing functions. Since maximum rank clicked and the number of clicks per session directly affect the *Patience* parameter, we can say that Queries/sec is inversely proportional to the *Patience* of the user.

Both the graphs show occasional phases of increasing behaviour which can be attributed to a variety of reasons. While plotting the graphs, for a given value of MR/number of clicks, we take observations from numerous sessions of the user and average the queries/sec value. Thus, presence of some outlier values may affect the overall out-

[2]We used laplace smoothing technique to negate the effect of zero probability instances.

[3]We performed these experiments on the query log data of a popular commerical search engine. The data consists of 21 million web queries collected from 650,000 users. The query log data consists of anonymous id given to the user, query, the time at which the query was posed, rank of the clicked URL (if any) and the URL of the document clicked by the user (if any).

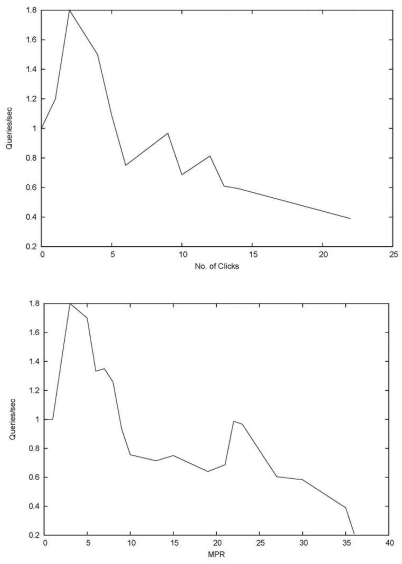


Figure 4: Clicks Vs Queries/sec and MR Vs Queries/sec

put of the graph. It can also be attributed to the low quality of results that the search engine might have returned due to various reasons.

5 Related Work

Although simulation-based methods have been used to test query modification techniques (Harman, 1988) or to detect shifts in the interests of computer users (Mostafa et al., 2003), to our knowledge not much research went into creating relevance feedback for web search based on search simulations.

Searcher simulations were created by White et al (Mostafa et al., 2003; White et al., 2005), for evaluating implicit feedback models. The simulations assume the role of a searcher, browsing the results of a retrieval. It is assumed that the actual relevant and irrelevant documents for a query are given. The system creates simulations of searchers by simulating relevance paths i.e., how the user would traverse results of retrieval. Different strategies were experimented like, the users only view relevant/non-relevant information, i.e., follow relevant paths from only relevant or only non-relevant documents, or they view all relevant or all non-relevant information, i.e., follow all relevance paths from top-ranked relevant documents or top-ranked non-relevant documents etc. Their research tries to model only certain phases of the search process like clicking the results and to some extent the process of looking and identifying the results to click. It also does not consider modeling the nature of the searcher in context and also does not calculate the relevance of a document for a user. The search process is not complete without discussing or characterizing the user that participates in the search and computing the relevance of a document for a user.

In (Agichtein et al.,, 2006), they show that clickthrough data and other implicit data of a user can be used to build user models to effectively personalize the search results. Craswell et al (Craswell et al., 2008) have also done some good work in this area. They try to model the results browsing pattern of the user. (Craswell et al., 2008) brings out the position bias in the user's click-decision making process. It provides some interesting browsing models which can be used in our prognostic search process. We used the cascade model – best performing model – proposed by them to compare the effectiveness of our approach.

In our approach, we address some of these issues to improve the reliability of the simulated feedback and the scalability of the simulations. We first identify certain parameters that are natural to the search process on the whole and are generic to hold well across search engines and users. Wherever applicable we try to characterize these parameters as probabilistic distributions, using large volumes of data from existing search engine clickthrough logs. We then instantiate these parameters by drawing values from these probabilistic distributions. This ensures that the simulated feedback resembles as closely as possible to the real world scenario and thus is of high quality. We can easily run the simulations on large sets of documents to create large amounts of simulated feedback, as there are no interventions of a human to provide any kind of extra information or relevance information on the document set.

6 Evaluation

In this section, we present the evaluation procedure of our approach. We first collected query

Table 1: System Configurations

System	Patience	Clicks
System1	Random	Random
System2	Random	Proposed method
System3	Proposed method	Proposed method

log data of 60 users using a browser plug-in for two months. Our query log data consists user-id, queries and the time at which they are entered, list of search results – rank, title, snippet and url of the result –[4] and the results clicked by the user. We used 70% of this query log data to build profiles of the searchers and the rest of the data is used for evaluation purpose. Using the rest of the query log data, we initiated the prognostic search process giving the queries sequentially in the order given by the user. We compared the simulated clicks with the clicks already generated by the user. We found that the data generated by us is 77% accurate and its recall[5] value is 68%. We measured the accuracy of our system as follows.

$$\text{Accuracy} = \frac{\text{No. of simulated clicks clicked by the user}}{\text{Total no. of results clicked by the user}} \quad (4)$$

We also built two more systems which we considered as the baseline systems. The first system gives a random value for the patience value of the user – random value is used to determine the number of documents to be browsed during the prognostic search process – and random value is given for the user's *Perceived relevance threshold* parameter. The second system generates the patience value of the user according to the process described by us in section 3.2.3 and gives a random value for the *Perceived relevance threshold* value of the user. Systems built by us can be summarized as shown in table 1:

Figure 5 shows a comparision of the accuracies of the three systems. Here, we can see that the

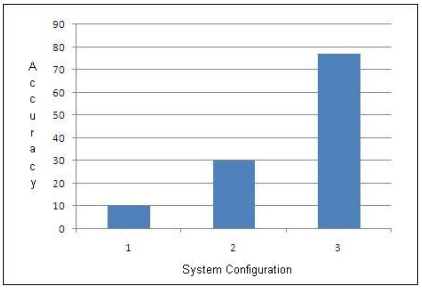


Figure 5: Results comparision

baseline 1 which uses random values for *patience* and *generating clicks* is only 10% accurate in generating clickthrough data. However, with the addition of our *generating clicks* approach to the baseline 1, the performance increased by 200%. And the system 3 which uses our proposed models for both *patience* and *generating clicks* generates 77% accurate data which is a 670% improvement over the baseline 1.

We also performed manual evaluation of our system. Since manual evaluation requires a lot of effort, we performed it using 25 judges. We randomly selected 25 users from our query log data and used their data to build profiles. Then we showed the clicks generated by our system to these users. Based on their judgements, we found our system to be 79.5% accurate[6]. Figure 6 shows the accuracy levels of our system according to different judges. We also studied the reason behind the increase in accuracy of our system during human evaluation. We re-examined the clicks generated by the users and found that the users selected the results which they have not selected during their regular search. And the reasons behind these extra clicks are: they have missed examining these results or they have already reached their desired document. Thus it certifies that our system is able to personalize the results and the perceived relevance technique can be used to re-rank the results to personalize them.

As the *cascade* model is the best performing model in (Craswell et al., 2008), we evaluated our system on that model for comparision. We found our system to be 96% accurate. We used the data collected in our clickthrough logs for evaluating

[4]A typical search engine query log does not contain the snippets of the results and the whole list of search results. It only contains the link clicked by the user and the rank of that result.

[5]Recall is the fraction of results clicked for this query and simulated successfully

[6]We took the average of the accuracies of our system for each of these judges/users.

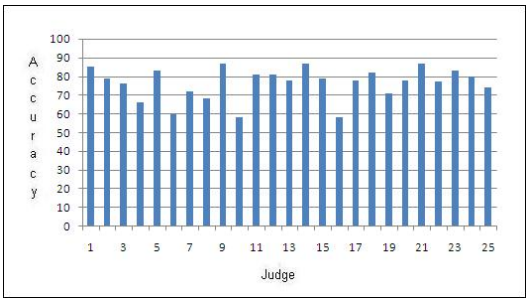


Figure 6: Accuracy based on human judge evaluation

our system using this model.

7 Conclusion and Future work

In this paper, we proposed Simulated Feedback based on insights from clickthrough data and using prognostic search methods to generate feedback. There is a lot of scope for interesting future directions to the current work. It would be an interesting experiment to see the use of the simulated feedback in evaluation of personalized search algorithms. Consider a personalized search algorithm, and use it to learn a user model from existing explicit/implicit feedback data. Learn a user model using the same algorithm from simulated feedback and compare the results. We plan to pursue the same in future.

As an extension to the current work, we aim to improve the web search process especially the query formulation step with insights from a user study. We are working towards incorporating much richer and complex models for query formulation like HMMs etc. Ability of the system to automatically create query reformulations of the original when no clicks are found is another interesting future work. We also plan to dig more information about the user by analysing the query log data. For example, the difference in the time between the clicks and the distance between the clicks can be used to analyze the browsing behaviour of the user. These observations can inturn be used in generation of simulated feedback thus reducing its gap with real world implicit feedback.

References

Mark Claypool, Phong Lee, Makoto Wased and David Brown. 2001. *Implicit interest indicators*. In Intelligent User Interfaces.

Granka L., Joachims J., and Gay G. 2004. *Eyetracking analysis of user behavior in www search*. Conference on Research and Development in Information Retrieval, SIGIR.

Harman D. 1988. *Towards interactive query expansion*. The 11th Annual ACM SIGIR Conference on Research and Development in Information Retrieval, 321-331.

Thorsten Joachims. 2002. *Optimizing search engines using clickthrough data*. Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, 133-142.

Kelly D., and Belkin N.J. 2001. *Reading time, scrolling and interaction: Exploring implicit sources of user preferences for relevance feedback during interactive information retrieval*. In Proceedings of the 24th Annual International Conference on Research and Development in Information Retrieval, SIGIR, 408-409.

Mostafa J., Mukhopadhyay S., and Palakal M. 2003. *Simulation studies of different dimensions of users' interests and their impact on user modelling and information filtering*. Information Retrieval, 199-223.

Filip Radlinski and Thorsten Joachims. 2005. *Evaluating the robustness of learning from implicit feedback*. In ICML Workshop on Learning In Web Search.

Rocchio J.J. 1999. *The SMART Retrieval System Experiments in Automatic Document Processing*. Relevance Feedback in Information Retrieval.

Sugiyama K., Hatano K., and Yoshikawa M. 2004. *Adaptive web search based on user profile constructed without any effort from users*. In Proceedings of WWW, 675-684.

Ryen W. White, Ian Ruthven, Joemon M. Jose and C.J van Rijsbergen. 2005. *Evaluating implicit feedback models using searcher simulations*. ACM Transactions on Information Systems,ACM TOIS, 325-361.

Xuehua Shen, Bin Tane and Bin Tan. 2005. *Implicit user modeling for personalized search*. ACM Transactions on Information Systems.

Feng Qiu and Junghoo Cho. 2006. *Automatic Identification of User interest for personalized search*. In proceedings of WWW.

Steve Fox, Kuldeep Karnawat, Mark Mydland, Susan Dumais and Thomas White. 2005. *Evaluating implicit measures to improve web search*. ACM Transactions on Information Systems, 147-168.

Eugene Agichtein, Eric Brill, Susan Dumais and Robert Ragno. 2006. *Learning user interaction models for predicting web search result preferences*. In proceedings of 29th conference on research and development in information retrieval, SIGIR, 3-10.

Thorsten Joachims, Laura Granka and Bing Pan. 2005. *Accurately interpreting clickthrough data as implicit feedback*. In proceedings of 28th conference on research and development in information retrieval, SIGIR.

Thomas K. Landauer, Danielle S. Mc Namara and Simon Dennis. 2007. *Handbook of Latent Semantic Analysis*. Lawrence Erlbaum Associates.

Craswell N., Zoeter O., Taylor M. and Ramsey B. 2008. *An experimental comparision of click position-bias models*. In First ACM International Conference on Web Search and Data Mining WSDM.

Olivier Chapelle and Ya Zhang. 2009. *A Dynamic Bayesian Network Click Model for Web Search Ranking*. In proceedings of International World Wide Web Conference(WWW).

Fan Guo, Chao Liu and Yi-Min Wang. 2009. *Efficient Multipl-Click Models in Web Search*. In Second ACM International Conference on Web Search and Data Mining WSDM.

Harman D. 1992. *Relevance feedback revisited*. In proceedings of 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 1-10.

Salton G., and Buckley C. 1990. *Improving retrieval performance by relevance feedback*. Journal of the American Society for Information Science.

Daniel E. Rose, and Danny Levinson. 2004. *Understanding user goals in Web Search*. In proceedings of International World Wide Web Conference(WWW).

Best Topic Word Selection for Topic Labelling

Jey Han Lau,[♠♡] **David Newman,**[♠♢] **Sarvnaz Karimi**[♠] and **Timothy Baldwin**[♠♡]
♠ NICTA Victoria Research Laboratory
♡ Dept of Computer Science and Software Engineering, University of Melbourne
♢ Dept of Computer Science, University of California
jhlau@csse.unimelb.edu.au, newman@uci.edu, skarimi@unimelb.edu.au, tb@ldwin.net

Abstract

This paper presents the novel task of best topic word selection, that is the selection of the topic word that is the best label for a given topic, as a means of enhancing the interpretation and visualisation of topic models. We propose a number of features intended to capture the best topic word, and show that, in combination as inputs to a reranking model, we are able to consistently achieve results above the baseline of simply selecting the highest-ranked topic word. This is the case both when training in-domain over other labelled topics for that topic model, and cross-domain, using only labellings from independent topic models learned over document collections from different domains and genres.

1 Introduction

In the short time since its inception, topic modelling (Blei et al., 2003) has become a mainstream technique for tasks as diverse as multi-document summarisation (Haghighi and Vanderwende, 2009), word sense discrimination (Brody and Lapata, 2009), sentiment analysis (Titov and McDonald, 2008) and information retrieval (Wei and Croft, 2006). For many of these tasks, the multinomial topics learned by the topic model can be interpreted natively as probabilities, or mapped onto a pre-defined discrete class set. However, for tasks where the learned topics are provided to humans as a first-order output, e.g. for use in document collection analysis/navigation, it can be difficult for the end-user to interpret the rich statistical information encoded in the topics. This research is concerned with making topics more readily human interpretable, by selecting a single term with which to label the topic.

Although topics are formally a multinomial distribution over terms, with every term having finite probability in every topic, topics are usually displayed by printing the top-10 terms (i.e. the 10 most probable terms) in the topic. These top-10 terms typically account for about 30% of the topic mass for reasonable setting of number of topics, and usually provide sufficient information to determine the subject area and interpretation of a topic, and distinguish one topic from another.

Our research task can be illustrated via the top-10 terms in the following topic, learned from a book collection. Terms w_i are presented in descending order of $P(w_i|t_j)$ for the topic t_j:

trout fish fly fishing water angler stream rod flies salmon

Clearly the topic relates to fishing, and indeed, the fourth term *fishing* is an excellent label for the topic. The task is thus termed *best word* or *most representative word* selection, as we are selecting the label from the closed set of the top-N topic words in that topic.

Naturally, not all topics are equally coherent, however, and the lower the topic coherence, the more difficult the label selection task becomes. For example:

oct sept nov aug dec july sun lite adv globe

appears to conflate months with newspaper names, and no one of these topic words is able to capture the topic accurately. As such, our methodology presupposes an automatic means of rating topics for coherence. Fortunately, recent research by Newman et al. (2010) has shown that this is achievable at levels approaching human performance, meaning that this is not an unreasonable assumption.

Labelling topics has applications across a diverse range of tasks. Our original interest in the

Coling 2008: Poster Volume, pages 605–613,
Beijing, August 2010

problem stems from work in document collection visualisation/navigation, and the realisation that presenting users with topics natively (e.g. as represented by the top-N terms) is ineffective, and would be significantly enhanced if we could automatically predict succinct labels for each topic. Another application area where labelling has been shown to enhance the utility of topic models is selectional preference learning via topic modelling (Ritter et al., to appear). Here, topic labelling via taxonomic classes (e.g. WordNet synsets) can lead to better topic generalisation, in addition to better human readability.

This paper is based around the assumption that an appropriate label for a topic can be found among the high-ranking (high probability) terms in that topic. We assess the suitability of each term by way of comparison with other high-ranking terms in that same topic, using simple pointwise mutual information and conditional probabilities. We first experiment with a simple ranking method based on the component scores, and then move on to using those scores, along with features from WordNet and from the original topic model, in a ranking support vector regression (SVR) framework. Our experiments demonstrate that we are able to perform the task significantly better than the baseline of selecting the topic word of highest marginal probability, including when training the ranking model on labelled topics from other document collections.

2 Related Work

Predictably, there has been significant work on interpreting topics in the context of topic modelling. Topic are conventionally interpreted via the top-N words in each topic (Blei et al., 2003; Griffiths and Steyvers, 2004), or alternatively by post-hoc manual labelling of each topic based on domain knowledge and subjective interpretation of each topic (Wang and McCallum, 2006; Mei et al., 2006).

Mei et al. (2007) proposed various approaches for automatically suggesting phrasal labels for topics, based on first extracting phrases from the document collection, and subsequently ranking the phrases based on KL divergence with a given topic.

Magatti et al. (2009) proposed a method for labelling topics induced by hierarchical topic modelling, based on ontological alignment with the Google Directory (gDir) hierarchy, and optionally expanding topics based on a thesaurus or WordNet. Preliminary experiments suggest the method has promise, but the method crucially relies on both a hierarchical topic model and a pre-existing ontology, so has limited applicability.

Over the general task of labelling a learned semantic class, Pantel and Ravichandran (2004) proposed the use of lexico-semantic patterns involving each member of that class to learn a (usually hypernym) label. The proposed method was shown to perform well over the semantically homogeneous, fine-grained clusters learned by CBC (Pantel and Lin, 2002), but for the coarse-grained, heterogeneous topics learned by topic modelling, it is questionable whether it would work as well.

The first works to report on human scoring of topics were Chang et al. (2009) and Newman et al. (2010). The first study used a novel but synthetic intruder detection task where humans evaluate both topics (that had an intruder word), and assignment of topics to documents (that had an intruder topic). The second study had humans directly score topics learned by a topic model. This latter work introduced the pointwise mutual information (PMI) score to model human scoring. Following this work, we use PMI as features in the ranking SVR model.

3 Methodology

Our task is to predict which words annotators tend to select as most representative or best words when presented with a list of ten words. Since annotators are not generally unanimous in their choice of best word, we formulate this as a ranking task, and treat the top-1, 2 and 3 system-ranked items as the best words, and compare that to the top-1, 2 and 3 words chosen most frequently by annotators. In this section, we describe the features that may be useful for this ranking task. We start with features motivated by word association.

An obvious idea is that the most representative word should be readily evoked by other words in the topic. For example, given a list of words $\langle$*space, earth, moon, nasa, mission*$\rangle$, which is a

Space Exploration topic, *space* could arguably be the most representative word. This is because it is natural to think about the word *space* after seeing the words *earth, moon* and *nasa* individually. A good candidate for best word could be the word that has high average conditional probability given each of the other words. To calculate conditional probability, we use word counts from the entire collection of English Wikipedia articles. Conditional probability is defined as:

$$P(w_i|w_j) = \frac{P(w_i, w_j)}{P(w_j)},$$

where $i \neq j$ and $P(w_i, w_j)$ is the probability of observing both w_i and w_j in the same sliding window, and $P(w_i)$ is the overall probability of word w_i in the corpus. In the above example, *evoked by* means that *space* would fill the slot of w_i. The average conditional probability for word w_i is given by:

$$\text{avg-CP1}(w_i) = \frac{1}{9}\sum_j P(w_i|w_j),$$

for $j = 1 \ldots 10, j \neq i$ (this range of indices applies to all following average quantities).

In other cases, we have the flip situation, where the most representative word may evoke (rather than be evoked by) other words in the list of ten words. Imagine a *NASCAR Racing* topic, which has a list of words ⟨*race, car, nascar, driver, racing*⟩. Given the word *nascar*, words from the list such as *race, car, racing* and *driver* might come to mind because *nascar* is heavily associated with these words. Therefore, a good candidate, w_i, might also correlate with high $P(w_j|w_i)$. As before, the average conditional probability (here denoted with CP2) for word w_i is given by:

$$\text{avg-CP2}(w_i) = \frac{1}{9}\sum_j P(w_j|w_i).$$

Another approach to measuring word association is by calculating pointwise mutual information (PMI) between word pairs. Unlike conditional probability, PMI is symmetric and thus the order of words in a pair does not matter. We calculate PMI using word counts from English Wikipedia as follows:

$$\text{PMI}(w_i, w_j) \quad = \quad log\frac{P(w_i, w_j)}{P(w_i)P(w_j)}.$$

The average PMI for word w_i is given by:

$$\text{avg-PMI}(w_i) = \frac{1}{9}\sum_j \text{PMI}(w_i, w_j).$$

The topic model produces an ordered list of words for each topic, and the ordering is given by the marginal probability of each word given that topic, $P(w_i|t_j)$. The ranking of words based on these probabilities indicates the importance of a word in a topic, and it is also a feature that we use for predicting the most representative word.

We also observe that sometimes the most representative words are generalized concepts of other words. As such, hypernym relations could be another feature that may be relevant to predicting the best word. To this end, we use WordNet to find hypernym relations between pairs of words in a topic and obtain a set of boolean-valued relationships for each topic word.

Our last feature is the distributional similarity scores of Pantel et al. (2009), as trained over Wikipedia.[1] This takes the form of representing the distributional similarity between each pairing of terms $\text{sim}(w_i|w_j)$; if w_i is not in the top-200 most similar terms for a given w_j, we assume it to have a similarity of 0.

While the above features can be used alone to get a ranking on the ten topic words, we can also use various combinations of features in a reranking model such as support vector regression (SVM^{rank}: Joachims (2006)). Applying the features described above — conditional probabilities, PMI, WordNet hypernym relations, the topic model word rank, and Pantel's distributional similarity score — as features for SVM^{rank}, a ranking of words is produced and candidates for the most representative word are selected by choosing the top-ranked words.

NEWS	stock market investor fund trading investment firm exchange ...
	police gun officer crime shooting death killed street victim ...
	food restaurant chef recipe cooking meat meal kitchen eat...
	patient doctor medical cancer hospital heart blood surgery ...
BOOKS	loom cloth thread warp weaving machine wool cotton yarn ...
	god worship religion sacred ancient image temple sun earth ...
	crop land wheat corn cattle acre grain farmer manure plough ...
	sentence verb noun adjective grammar speech pronoun ...

Figure 1: Selected topics from the two collections (each line is one topic, with fewer than ten topic words displayed because of limited space)

Features	Description
PMI	Pointwise mutual information
CP1	Conditional probability $P(w_i\|*)$
CP2	Conditional probability $P(*\|w_i)$
TM Rank	Original topic model word rank
Hypernym	WordNet hypernym relationships
PDS	Pantel distributional similarity score

Table 1: Description of feature sets

4 Datasets

We used two collections of text documents from different genres for our experiments. The first collection (NEWS) was created by selecting 55,000 news articles from the LDC Gigaword corpus. The second collection (BOOKS) was 12,000 English language books selected from the Internet Archive American Libraries collection. The NEWS and BOOKS collections provide a diverse range of content for topic modeling. In the first case – news articles from the past decade written by journalists — each article usually attempts to clearly and concisely convey information to the reader, and hence the learned topics tend to be fairly interpretable. For BOOKS (with publication dates spanning more than a century), the writing style often uses lengthy and descriptive prose, so one sees a different style to the learned topics.

The input to the topic model is a bag-of-words representation of the collection of text documents, where word counts are preserved, but word order is lost. After performing fairly standard tokenization and limited lemmatisation, and creating a vocabulary of terms that occurred at least ten times, each corpus was converted into its bag-of-words representation. We learned topic models for the two collections, choosing a setting of $T = 200$ topics for NEWS and $T = 400$ topics for BOOKS. After computing the PMI-score for each topic (according to Newman et al. (2010)), we selected 60 topics with high PMI-score, and 60 topics with low PMI-score, from both corpora, resulting in a total of 240 topics for human evaluation.

The 240 topics selected for human scoring were each evaluated by between 10 and 20 users. For the two topic models, we used the conventional approach of displaying each topic with its top-10 terms. In a typical survey, a user was asked to evaluate anywhere from 60 to 120 topics. The instructions asked the user to perform the following tasks, for each topic in the survey: (a) score the topic for "usefulness" or "coherence" on a scale of 1 to 3; and (b) select the single best word that exemplifies the topic (when score=3).

From both NEWS and BOOKS, the 40 topics with the highest average human scores had relatively complete data for the 'best word' selection task (i.e. every time a user gave a topics score=3, they also selected a 'best word'). The remainder of this paper is concerned with the 40 NEWS topics and 40 BOOKS topics where we had 'best word' data from the annotators. Sample topics from these two sets are given in Figure 1.

To measure presentational bias (i.e. the extent to which annotators tend to choose a word seen earlier rather than later, particularly when armed with the knowledge that words are presented in order of probability), we reissued a survey using the 40 NEWS topics to ten additional annotators, but this time the top-10 topic words were presented in random order. Again, these ten new annotators were asked to select the best word.

5 Experiments

We used average PMI and conditional probabilities, CP1 and CP2, to rank the ten words in each topic. Candidates for the best words were selected by choosing the top-1, 2 and 3 ranked words.

We used the following weighted scoring function for evaluation:

$$\text{Best-N score} = \frac{\sum_{i=1}^{N} n(w_{\text{rev}_i})}{\sum_{i=1}^{N} n(w_i)}$$

[1]Accessed from `http://demo.patrickpantel.com/Content/LexSem/thesaurus.htm`.

Features	Best-1	Best-2	Best-3
Baseline	0.35	0.50	0.59
PMI	0.25	0.38	0.49
CP1	0.30	0.42	0.51
CP2	0.15	0.27	0.45
Upper bound	0.48	—	—

Table 2: Best-1,2,3 scores for ranking with single feature sets (PMI and both conditional probabilities) for NEWS

Features	Best-1	Best-2	Best-3
Baseline	0.38	0.48	0.60
PMI	0.25	0.38	0.49
CP1	0.30	0.38	0.47
CP2	0.15	0.30	0.49
Upper bound	0.64	—	—

Table 3: Best-1,2,3 scores for ranking with single feature sets (PMI and both conditional probabilities) for BOOKS

where w_{rev_i} is the i^{th} term ranked by the system and w_i is the i^{th} most popular term selected by annotators; rev_i gives the index of the word w_i in the annotator's list; and $n(w)$ is the number of votes given by annotators for word w.

The baseline is obtained using the original word rank produced by the topic model based on topic word probabilities $P(w_i|t_j)$. An upperbound is calculated by evaluating the decision of an annotator against others for each topic. This upperbound signifies the maximum accuracy for human annotators on average; since the annotators were asked to pick a single best word in the survey, only the Best-1 upperbound can be obtained.

The Best-1/2/3 results are summarized in Table 2 for NEWS and Table 3 for BOOKS. These Best-N scores are computed just using the single feature of PMI, CP1 and CP2 (each in turn) to rank the words in each topic. None of these features alone produces a result that exceeds baseline performance.

To make better use of all the features described in Section 3, namely the PMI score, conditional probabilities (both directions), topic model word rank, WordNet Hypernym relationships and Pantel's distributional similarity score, we build a ranking classifier using SVM^{rank} and evaluating

Feature Set	Best-1	Best-2	Best-3
Baseline	0.35	0.50	0.59
All Features	0.43	0.56	0.62
−PMI	0.45 (+0.02)	0.52 (−0.04)	0.62 (±0.00)
−CP1	0.35 (−0.08)	0.49 (−0.07)	0.57 (−0.05)
−CP2	0.40 (−0.03)	0.50 (−0.06)	0.61 (−0.01)
−TM Rank	0.40 (−0.03)	0.52 (−0.04)	0.57 (−0.05)
−Hypernym	0.43 (±0.00)	0.57 (+0.01)	0.62 (±0.00)
−PDS	0.43 (±0.00)	0.53 (−0.03)	0.62 (±0.00)
Upper bound	0.48	—	—

Table 4: SVR-based best topic word results for NEWS for all six feature types, and feature ablation over each (numbers in brackets show the relative change over the full feature set)

Feature Set	Best-1	Best-2	Best-3
Baseline	0.38	0.48	0.60
All Features	0.40	0.51	0.62
−PMI	0.38 (−0.02)	0.51 (±0.00)	0.63 (+0.01)
−CP1	0.33 (−0.07)	0.47 (−0.04)	0.56 (−0.06)
−CP2	0.40 (±0.00)	0.50 (−0.01)	0.64 (+0.02)
−TM Rank	0.35 (−0.05)	0.49 (−0.02)	0.63 (+0.01)
−Hypernym	0.40 (±0.00)	0.50 (−0.01)	0.61 (−0.01)
−PDS	0.45 (+0.05)	0.48 (−0.03)	0.67 (+0.05)
Upper bound	0.64	—	—

Table 5: SVR-based best topic word results for BOOKS for all six feature types, and feature ablation over each (numbers in brackets show the relative change over the full feature set)

using 10-fold cross validation. Our first approach is to use the entire set of features to train the classifier. Following this, we also measure the effect of each feature by ablating (removing) one feature at a time. The drop in Best-N score indicates which features are the strongest predictors of the best words (a larger drop in score indicates that feature is more important). The results for Best-1, Best-2 and Best-3 scores are summarized in Table 4 for NEWS, and Table 5 for BOOKS (averaged across the 10 iterations of cross validation).

We then produced a condensed set of features, consisting of the conditional probabilities, the original topic model word rank and the WordNet hypernym relationships. This "best" set of features is used to make predictions of best words. Results are improved in most cases, and are summarized in Table 6 for both NEWS and BOOKS.

Dataset		Best-1	Best-2	Best-3
NEWS	Baseline	0.35	0.50	0.59
	Best Feat. Set	0.45	0.50	0.65
	Upper bound	0.48	—	—
BOOKS	Baseline	0.38	0.48	0.60
	Best Feat. Set	0.48	0.56	0.66
	Upper bound	0.64	—	—

Table 6: Results with the best feature set compared to the baseline

Dataset	Best-1	Best-2	Best-3
NEWS baseline	0.35	0.50	0.59
BOOKS → NEWS	0.38	0.56	0.62
NEWS upper bound	0.48	—	—
BOOKS baseline	0.38	0.48	0.60
NEWS → BOOKS	0.48	0.56	0.65
BOOKS upper bound	0.64	—	—

Table 7: Results for cross-domain learning

We also tested whether the SVM classifier could be trained using data from one domain, and run on data from another domain. Using our two datasets as these different domains, we trained a model using BOOKS data and made predictions for NEWS, and then we trained a model using NEWS data and made predictions for BOOKS.

The results, shown in Table 7, indicate that we are still able to outperform the baseline, even when the ranking classifier is trained on a different domain. In fact, when we trained a model using NEWS, we saw almost no drop in performance for predicting best words for BOOKS, and improvement is seen for Best-2 score from NEWS. This implies that the SVM classifier generalizes well across domains and suggests the possibility of having a fixed training model to predict best words for any data.

In these experiments, topic words are presented in the original order that the topic model produces, i.e. in descending order of probability of a word under a topic $P(w_i|t_j)$. We noticed that the first words of the topics are frequently selected as the best words by annotators, and suspected that this was introducing a bias towards the first word. As our baseline scores are derived from this topic word ordering, such a bias could give rise to an artificially high baseline.

To investigate this effect, we ran a second annotation exercise over the same set of topics (but different annotators), to obtain a new set of best word annotations for NEWS, with the topic words presented in random order. In Figure 2, we plot the cumulative proportion of words selected as best word by the annotators across the topics, in the case of the random topic word order, mapping the topic words back onto their original ranks in the topic model. A slight drop can be observed in the proportion of first- and second-ranked topic words being selected when we randomise the topic word order. When we recalculate the baseline accuracy for NEWS on the basis of the new set of annotations, we observe an appreciable drop in the scores (see Table 8).

Word Order	Best-1	Best-2	Best-3
Original	0.35	0.50	0.59
Randomized	0.23	0.33	0.46

Table 8: Reduction of baseline scores for NEWS when words are presented in random order to annotators.

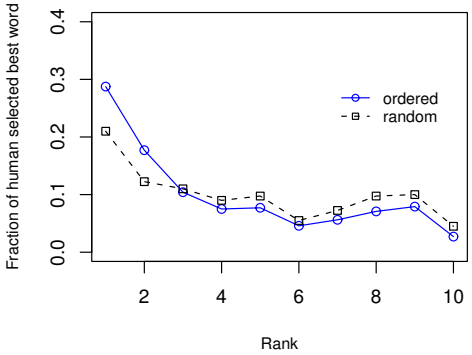


Figure 2: Bias for humans selecting the best word, when the topic words are presented in their original ordering (*ordered*) or randomised (*random*)

6 Discussion

From the experiments in Section 5, perhaps the first thing to observe is: (a) the high performance of the baseline, and (b) the relatively low (Best-1) upper bound accuracy for the task. The first is perhaps unsurprising, given that it represents the

topic model's own interpretation of the word(s) which are most representative of that topic. In this sense, we have set our sights high in attempting to better the baseline. The upper bound accuracy is a reflection of both the inter-annotator agreement, and the best that we can meaningfully expect to do for the task. That is, any result higher than this would paradoxically suggest that we are able to do better at a task than humans, where we are evaluating ourselves relative to the labellings of those humans. The upper bound for NEWS was slightly less than 0.5, indicating that humans agree on the best topic word only 50% of the time. To better understand what is happening here, consider the following topic from Figure 1:

health drug patient medical doctor hospital care cancer treatment disease

This is clearly a coherent topic, but at least two topic words suggest themselves as labels: *health* and *medical*. By way of having between 10 and 20 annotators (uniquely) label a given topic, and interpreting the multiple labellings probabilistically, we are side-stepping the inter-annotator agreement issue, but ultimately, for the Best-1 evaluation, we are forced to select one term only, and consider any alternative to be wrong. Because annotators selected only one best topic word, we unfortunately have no way of performing Best-2 or Best-3 upper bound evaluation and deal with topics such as this, but would expect the numbers to rise appreciably.

Looking at the original feature rankings in Tables 2 and 3, no clear picture emerges as to which of the three methods (PMI, CP1 and CP2) was most successful, but there were certainly clear differences in the relative numbers for each, pointing to possible complementarity in the scoring. This expectation was born out in the results for the reranking model in Tables 4 and 5, where the combined feature set surpassed the baseline in all cases, and feature ablation tended to lead to a drop in results, with the single most effective feature set being CP1 ($P(w_i|*)$), followed by CP2 ($P(*|w_i)$) and topic model rank. The lexical semantic features of WordNet hypernymy and PDS (Pantel's distributional similarity) were the worst performers, often having no or negative impact on the results.

Comparing the best results for the SVR-based reranking model and the upper bound Best-1 score, we approach the upper bound performance for NEWS, but are still quite a way off with BOOKS when training in-domain. This is encouraging, but a slightly artificial result in terms of the broader applicability of this research, as what it means in practical terms is that if we can access multi-annotator best word labelling for the majority of topics in a given topic model, we can use those annotations to predict the best word for the remainder of the topics with reasonably success. When we look to the cross-domain results, however, we see that we almost perfectly replicate the best-achieved Best-1, Best-2 and Best-3 in-domain results for BOOKS by training on NEWS (making no use of the annotations for BOOKS). Applying the annotations for BOOKS to NEWS is less successful in terms of Best-1 accuracy, but we actually achieve higher Best-2, and largely mirror the Best-3 results as compared to the best of the in-domain results in Table 6. This leads to the much more compelling conclusion that we can take annotations from an independent topic model (based on a completely unrelated document collection), and apply them to successfully model the best topic word for a new topic model, without requiring any additional annotation. As we now have two sets of topics multiply-annotated for best words, this result suggests that we can perform the best topic word selection task with high success over novel topic models.

We carried out manual analysis of topics where the model did particularly poorly, to get a sense for how and where our model is being led astray. One such example is the topic:

race car nascar driver racing cup winston team gordon season

where the following topic words were selected by our annotators: *nascar* (8 people), *race* (2 people), and *racing* (2 people). First, we observe the split between *race* and *racing*, where more judicious lemmatisation/stemming would make both the annotation easier and the evaluation cleaner. The SVR model tends to select more common, general terms, so in this case chose *race* as the best word, and ranked *nascar* third. This is one

instance were *nascar* evokes all of the other words effectively, but not conversely (*racing* is associated with many events/sports beyond *nascar*, e.g.).

Another topic where our model had difficulty was:

window nave aisle transept chapel tower arch pointed arches roof

where our best model selected *nave*, while the human annotators selected *chapel* (6 people), *arch* (2 people), *nave*, *roof*, *tower* and *transept* (1 person each). Clearly, our annotators struggled to come up with a best word here, despite the topic again being coherent. This is an obvious candidate for labelling with a hypernym/holonym of the topic words (e.g. *church* or *church architecture*), and points to the limitations of best word labelling — there are certainly many topics where best word labelling works, as our upper bound analysis demonstrated, but there are equally many topics where the most natural label is not found in the top-ranked topic words. While this points to slight naivety in the current task set up — we are forcing annotators to label words with topic words, where we know that this is sub-optimal for a significant number of topics — we contend that our numbers suggest that: (a) consistent best topic word labelling is possible at least 50% of the time; and (b) we have developed a method which is highly adept at labelling these topics. As a way forward, we intend to relax the constraint on the topic label needing to be based on a topic word, and explore the possibility of predicting which topics are best labelled with topic words, and which require independent labels. For topics which can be labelled with topic words, we can use the methodology developed here, and for topics where this is predicted to be sub-optimal, we intend to build on the work of Mei et al. (2007), Pantel and Ravichandran (2004) and others in selecting phrasal/hypernym labels for topics. We are also interested in applying the methodology proposed herein to the closely-related task of intruder word, or *worst* topic word, detection, as proposed by Chang et al. (2009).

Finally, looking to the question of the impact of the presentation order of the topic words on best word selection, it would appear that our baseline is possibly an over-estimate (based on Table 8). Having said that, the flipside of the bias is that it leads to more consistency in the annotations, and tends to help in tie-breaking of examples such as *race* and *racing* from above, for example. In support of this claim, the upper bound Best-1 accuracy of the randomised annotations, relative to the original gold-standard is 0.44, slightly below the original upper bound for NEWS. More work is needed to determine the real impact of this bias on the overall task setup and evaluation.

7 Conclusion

This paper has presented the novel task of best topic word selection, that is the selection of the topic word that is the best label for a given topic. We proposed a number of features intended to capture the best topic word, and demonstrated that, while they were relatively unsuccessful in isolation, in combination as inputs to a reranking model, we were able to consistently achieve results above the baseline of simply selecting the highest-ranked topic word, both when training in-domain over other labelled topics for that topic model, and cross-domain, using only labellings from independent topic models learned over document collections from different domains and genres.

Acknowledgements

NICTA is funded by the Australian government as represented by Department of Broadband, Communication and Digital Economy, and the Australian Research Council through the ICT centre of Excellence programme. DN has also been supported by a grant from the Institute of Museum and Library Services, and a Google Research Award.

References

Blei, D.M., A.Y. Ng, and M.I. Jordan. 2003. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.

Brody, S. and M. Lapata. 2009. Bayesian word sense induction. In *Proceedings of the 12th Conference of the EACL (EACL 2009)*, pages 103–111, Athens, Greece.

Chang, J., J. Boyd-Graber, S. Gerrish, C. Wang, and D. Blei. 2009. Reading tea leaves: How humans interpret topic models. In *Proceedings of the 23rd*

Annual Conference on Neural Information Processing Systems (NIPS 2009), pages 288–296, Vancouver, Canada.

Griffiths, T. and M. Steyvers. 2004. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101:5228–5235.

Haghighi, A. and L. Vanderwende. 2009. Exploring content models for multi-document summarization. In *Proceedings of the North American Chapter of the Association for Computational Linguistics – Human Language Technologies 2009 (NAACL HLT 2009)*, pages 362–370, Boulder, USA.

Joachims, T. 2006. Training linear SVMs in linear time. In *Proceedings of the ACM Conference on Knowledge Discovery and Data Mining (KDD)*, pages 217–226, Philadelphia, USA.

Magatti, D., S. Calegari, D. Ciucci, and F. Stella. 2009. Automatic labeling of topics. In *Proceedings of the International Conference on Intelligent Systems Design and Applications*, pages 1227–1232, Pisa, Italy.

Mei, Q., C. Liu, H. Su, and C. Zhai. 2006. A probabilistic approach to spatiotemporal theme pattern mining on weblogs. In *Proceedings of the 15th International World Wide Web Conference (WWW 2006)*, pages 533–542.

Mei, Q., X. Shen, and C. Zhai. 2007. Automatic labeling of multinomial topic models. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2007)*, pages 490–499, San Jose, USA.

Newman, D., J.H. Lau, K. Grieser, and T. Baldwin. 2010. Automatic evaluation of topic coherence. In *Proceedings of Human Language Technologies: The 11th Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL HLT 2010)*, pages 100–108, Los Angeles, USA.

Pantel, P. and D. Lin. 2002. Discovering word senses from text. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 613–619, Edmonton, Canada.

Pantel, P. and D. Ravichandran. 2004. Automatically labeling semantic classes. In *Proceedings of the 4th International Conference on Human Language Technology Research and 5th Annual Meeting of the NAACL (HLT-NAACL 2004)*, pages 321–328, Boston, USA.

Pantel, P., E. Crestan, A. Borkovsky, A-M. Popescu, and V. Vyas. 2009. Web-scale distributional similarity and entity set expansion. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP 2009)*, pages 938–947, Singapore.

Ritter, A, Mausam, and O Etzioni. to appear. A latent Dirichlet allocation method for selectional preferences. In *Proceedings of the 48th Annual Meeting of the ACL (ACL 2010)*, Uppsala, Sweden.

Titov, I. and R. McDonald. 2008. Modeling online reviews with multi-grain topic models. In *Proceedings of the 17th International World Wide Web Conference (WWW 2008)*, pages 111–120, Beijing, China.

Wang, X. and A. McCallum. 2006. Topics over time: A non-Markov continuous-time model of topical trends. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2006)*, pages 424–433, Philadelphia, USA.

Wei, S. and W.B. Croft. 2006. LDA-based document models for ad-hoc retrieval. In *Proceedings of 29th International ACM-SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2006)*, pages 178–185, Seattle, USA.

A Linguistically Grounded Graph Model for Bilingual Lexicon Extraction

Florian Laws, Lukas Michelbacher, Beate Dorow, Christian Scheible, Ulrich Heid, Hinrich Schütze
Institute for Natural Language Processing
Universität Stuttgart
{lawsfn,michells,dorowbe}@ims.uni-stuttgart.de

Abstract

We present a new method, based on graph theory, for bilingual lexicon extraction without relying on resources with limited availability like parallel corpora. The graphs we use represent linguistic relations between words such as adjectival modification. We experiment with a number of ways of combining different linguistic relations and present a novel method, multi-edge extraction (MEE), that is both modular and scalable. We evaluate MEE on adjectives, verbs and nouns and show that it is superior to cooccurrence-based extraction (which does not use linguistic analysis). Finally, we publish a reproducible baseline to establish an evaluation benchmark for bilingual lexicon extraction.

1 Introduction

Machine-readable translation dictionaries are an important resource for bilingual tasks like machine translation and cross-language information retrieval. A common approach to obtaining bilingual translation dictionaries is *bilingual lexicon extraction* from corpora. Most work has used *parallel text* for this task. However, parallel corpora are only available for few language pairs and for a small selection of domains (e.g., politics). For other language pairs and domains, monolingual comparable corpora and monolingual language processing tools may be more easily available. This has prompted researchers to investigate bilingual lexicon extraction based on monolingual corpora (see Section 2) .

In this paper, we present a new graph-theoretic method for bilingual lexicon extraction. Two monolingual graphs are constructed based on syntactic analysis, with words as nodes and relations (such as adjectival modification) as edges. Each relation acts as a similarity source for the node types involved. All available similarity sources interact to produce one final similarity value for each pair of nodes. Using a seed lexicon, nodes from the two graphs can be compared to find a translation.

Our main contributions in this paper are: (i) we present a new method, based on graph theory, for bilingual lexicon extraction without relying on resources with limited availability like parallel corpora; (ii) we show that with this graph-theoretic framework, information obtained by linguistic analysis is superior to cooccurrence data obtained without linguistic analysis; (iii) we experiment with a number of ways of combining different linguistic relations in extraction and present a novel method, multi-edge extraction, which is both modular and scalable; (iv) progress in bilingual lexicon extraction has been hampered by the lack of a common benchmark; we therefore publish a benchmark and the performance of MEE as a baseline for future research.

The paper discusses related work in Section 2. We then describe our translation model (Section 3) and multi-edge extraction (Section 4). The benchmark we publish as part of this paper is described in Section 5. Section 6 presents our experimental results and Section 7 analyzes and discusses them. Section 8 summarizes.

2 Related Work

Rapp (1999) uses word cooccurrence in a vector space model for bilingual lexicon extraction. Details are given in Section 5.

Fung and Yee (1998) also use a vector space approach, but use TF/IDF values in the vector components and experiment with different vector similarity measures for ranking the translation candidates. Koehn and Knight (2002) combine

Coling 2008: Poster Volume, pages 614–622,
Beijing, August 2010

a vector-space approach with other clues such as orthographic similarity and frequency. They report an accuracy of .39 on the 1000 most frequent English-German noun translation pairs.

Garera et al. (2009) use a vector space model with dependency links as dimensions instead of cooccurring words. They report outperforming a cooccurrence vector model by 16 percentage points accuracy on English-Spanish.

Haghighi et al. (2008) use a probabilistic model over word feature vectors containing cooccurrence and orthographic features. They then use canonical correlation analysis to find matchings between words in a common latent space. They evaluate on multiple languages and report high precision even without a seed lexicon.

Most previous work has used vector spaces and (except for Garera et al. (2009)) cooccurrence data. Our approach uses linguistic relations like subcategorization, modification and coordination in a graph-based model. Further, we evaluate our approach on different parts of speech, whereas some previous work only evaluates on nouns.

3 Translation Model

Our model has two components: (i) a graph representing words and the relationships between them and (ii) a measure of similarity between words based on these relationships. Translation is regarded as cross-lingual word similarity. We rank words according to their similarity and choose the top word as the translation.

We employ undirected graphs with typed nodes and edges. Node types represent parts of speech (POS); edge types represent different kinds of relations. We use a modified version of SimRank (Jeh and Widom, 2002) as a similarity measure for our experiments (see Section 4 for details).

SimRank is based on the idea that two nodes are similar if their neighbors are similar. We apply this notion of similarity across two graphs. We think of two words as translations if they appear in the same relations with other words that are translations of each other. Figure 1 illustrates this idea with verbs and nouns in the direct object relation. Double lines indicate *seed* translations, i.e., known translations from a dictionary (see Section 5). The nodes *buy* and *kaufen* have the same

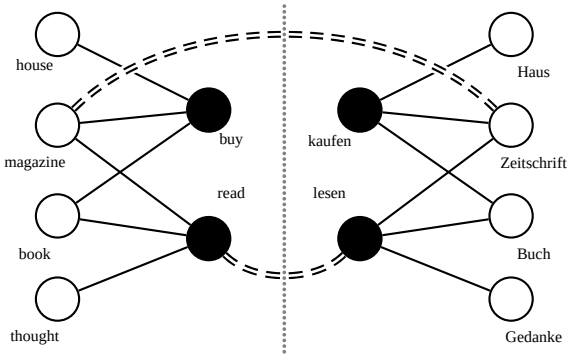


Figure 1: Similarity through seed translations

objects in the two languages; one of these (*magazine – Zeitschrift*) is a seed translation. This relationship contributes to the similarity of *buy – kaufen*. Furthermore, *book* and *Buch* are similar (because of *read – lesen*) and this similarity will be added to *buy – kaufen* in a later iteration. By repeatedly applying the algorithm, the initial similarity introduced by seeds spreads to all nodes.

To incorporate more detailed linguistic information, we introduce typed edges in addition to typed nodes. Each edge type represents a linguistic relation such as verb subcategorization or adjectival modification. By designing a model that combines multiple edge types, we can compute the similarity between two words based on *multiple sources* of similarity. We superimpose different sets of edges on a fixed set of nodes; a node is not necessarily part of every relation.

The graph model can accommodate any kind of nodes and relations. In this paper we use nodes to represent content words (i.e., non-function words): adjectives (a), nouns (n) and verbs (v). We extracted three types of syntactic relations from a corpus: see Table 1.

Nouns participate in two bipartite relations (amod, dobj) and one unipartite relation (ncrd). This means that the computation of noun similarities will benefit from three different sources.

Figure 2 depicts a sample graph with all node and edge types. For the sake of simplicity, a monolingual example is shown. There are four nouns in the sample graph all of which are (i) modified by the adjectives *interesting* and *political* and (ii) direct objects of the verbs *like* and

relation	entities	description	example
used in this paper			
amod	a, n	adjectival modification	a *fast car*
dobj	v, n	object subcategorization	*drive* a *car*
ncrd	n, n	noun coordination	*car*s and *buss*es
other possible relations			
vsub	v, n	subject subcategorization	a *man sleep*s
poss	n, n	possessive	the *child's toy*
acrd	a, a	adjective coordination	*red* or *blue* car

Table 1: Relations used in this paper (top) and possible extensions (bottom).

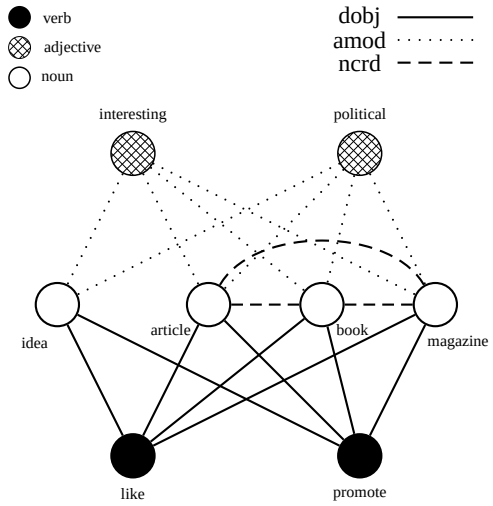


Figure 2: Graph snippet with typed edges

promote. Based on amod and dobj, the four nouns are equally similar to each other. However, the greater similarity of *article*, *book*, and *magazine* to each other can be deduced from the fact that these three nouns also occur in the relation ncrd. We exploit this information in the MEE method.

Data and Preprocessing. Our corpus in this paper is the Wikipedia. We parse all German and English articles with BitPar (Schmid, 2004) to extract verb-argument relations. We extract adjective-noun modification and noun coordinations with part-of-speech patterns based on a version of the corpus tagged with TreeTagger (Schmid, 1994). We use lemmas instead of surface forms. Because we perform the SimRank matrix multiplications in memory, we need to filter out rare words and relations; otherwise, running SimRank to convergence would not be feasible. For adjective-noun pairs, we apply a filter on pair frequency (≥ 3). We process noun pairs by applying a frequency threshold on words (≥ 100) and pairs (≥ 3). Verb-object pairs (the smallest data set) were not frequency-filtered. Based on the resulting frequency counts, we calculate association scores for all relationships using the log-likelihood measure (Dunning, 1993). For noun pairs, we discard all pairs with an association score < 3.84 (significance at $\alpha = .05$). For all three relations, we discard pairs whose observed frequency was smaller than their expected frequency (Evert, 2004, p. 76). As a last step, we further reduce noise by removing nodes of degree 1. Key statistics for the resulting graphs are given in Table 2.

We have found that accuracy of extraction is poor if unweighted edges are used. Using the log-likelihood score directly as edge weight gives too much weight to "semantically weak" high-frequency words like *put* and *take*. We therefore use the logarithms of the log-likelihood score as edge weights in all SimRank computations reported in this paper.

nodes		**n**	**a**	**v**
	de	34,545	10,067	2,828
	en	22,257	12,878	4,866
edges		**ncrd**	**amod**	**dobj**
	de	65,299	417,151	143,906
	en	288,889	686,073	510,351

Table 2: Node and edge statistics

4 SimRank

Our work is based on the SimRank graph similarity algorithm (Jeh and Widom, 2002). In (Dorow et al., 2009), we proposed a formulation of SimRank in terms of matrix operations, which can be applied to (i) weighted graphs and (ii) bilingual problems. We now briefly review SimRank and its bilingual extension. For more details we refer to (Dorow et al., 2009).

The basic idea of SimRank is to consider two nodes as similar if they have similar neighborhoods. Node similarity scores are recursively computed from the scores of neighboring nodes: the similarity S_{ij} of two nodes i and j is computed

as the normalized sum of the pairwise similarities of their neighbors:

$$S_{ij} = \frac{c}{|N(i)|\ |N(j)|} \sum_{k \in N(i), l \in N(j)} S_{kl}.$$

where $N(i)$ and $N(j)$ are the sets of i's and j's neighbors. As the basis of the recursion, S_{ij} is set to 1 if i and j are identical (self-similarity). The constant c ($0 < c < 1$) dampens the contribution of nodes further away. Following Jeh and Widom (2002), we use $c = 0.8$. This calculation is repeated until, after a few iterations, the similarity values converge.

For bilingual problems, we adapt SimRank for comparison of nodes across two graphs A and B. In this case, i is a node in A and j is a node in B, and the recursion basis is changed to $S(i, j) = 1$ if i and j are a pair in a predefined set of node-node equivalences (seed translation pairs).

$$S_{ij} = \frac{c}{|N_A(i)|\ |N_B(j)|} \sum_{k \in N_A(i), l \in N_B(j)} S_{kl}.$$

Multi-edge Extraction (MEE) Algorithm To combine different information sources, corresponding to edges of different types, in one SimRank computation, we use multi-edge extraction (MEE), a variant of SimRank (Dorow et al., 2009). It computes an aggregate similarity matrix after each iteration by taking the average similarity value over all edge types $\mathcal{T}$:

$$S_{ij} = \frac{c}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} \frac{1}{f(|N_{A,t}(i)|) f(|N_{B,t}(j)|)} \sum_{\substack{k \in N_{A,t}(i), \\ l \in N_{B,t}(j)}} S_{kl}.$$

f is a normalization function (either $f = g$, $g(n) = n$ as before or the normalization discussed in the next section).

While we have only reviewed the case of unweighted graphs, the extended SimRank can also be applied to weighted graphs. (See (Dorow et al., 2009) for details.) In what follows, all graph computations are weighted.

Square Root Normalization Preliminary experiments showed that SimRank gave too much influence to words with few neighbors. We therefore modified the normalization function $g(n) = n$. To favor words with more neighbors, we want f to grow sublinearly with the number of neighbors. On the other hand, it is important that, even for nodes with a large number of neighbors, the normalization term is not much smaller than $|N(i)|$, otherwise the similarity computation does not converge. We use the function $h(n) = \sqrt{n} * \sqrt{\max_k(|N(k)|)}$. h grows quickly for small node degrees, while returning values close to the linear term for large node degrees. This guarantees that nodes with small degrees have less influence on final similarity scores. In all experiments reported in this paper, the matrices $\tilde{A}$, $\tilde{B}$ are normalized with $f = h$ (rather than using the standard normalization $f = g$). In one experiment, accuracy of the top-ranked candidate (acc@1) was .52 for h and .03 for g, demonstrating that the standard normalization does not work in our application.

Threshold Sieving For larger experiments, there is a limit to scalability, as the similarity matrix fills up with many small entries, which take up a large amount of memory. Since these small entries contribute little to the final result, Lizorkin et al. (2008) proposed *threshold sieving*: an approximation of SimRank using less space by deleting all similarity values that are below a threshold. The quality of the approximation is set by a parameter δ that specifies maximum acceptable difference of threshold-sieved similarity and the exact solution. We adapted this to the matrix formulation by integrating the thresholding step into a standard sparse matrix multiplication algorithm.

We verified that this approximation yields useful results by comparing the ranks of exact and approximate solutions. We found that for the high-ranked words that are of interest in our task, sieving with a suitable threshold does not negatively affect results.

5 Benchmark Data Set

Rapp's (1999) original experiment was carried out on newswire corpora and a proprietary Collins dictionary. We use the free German (280M tokens) and English (850M tokens) Wikipedias as source and target corpora. Reinhard Rapp has generously provided us with his 100 word test set

	n	a	v
training set	.61	.31	.08
TS100	.65	.28	.07
TS1000	.66	.14	.20

Table 3: Percentages of POS in test and training

(TS100) and given us permission to redistribute it. Additionally, we constructed a larger test set (TS1000) consisting of the 1000 most frequent words from the English Wikipedia. Unlike the noun-only test sets used in other studies, (e.g., Koehn and Knight (2002), Haghighi et al. (2008)), TS1000 also contains adjectives and verbs. As seed translations, we use a subset of the dict.cc online dictionary. For the creation of the subset we took raw word frequencies from Wikipedia as a basis. We extracted all verb, noun and adjective translation pairs from the original dictionary and kept the pairs whose components were among the $5{,}000$ most frequent nouns, the $3{,}500$ most frequent adjectives and the 500 most frequent verbs for each language. These numbers are based on percentages of the different node types in the graphs. The resulting dictionary contains $12{,}630$ pairs: $7{,}767$ noun, $3{,}913$ adjective and 950 verb pairs. Table 3 shows the POS composition of the training set and the two test sets. For experiments evaluated on TS100 (resp. TS1000), the set of 100 (resp. 1000) English words it contains and all their German translations are removed from the seed dictionary.

Baseline. Our baseline is a reimplementation of the vector-space method of Rapp (1999). Each word in the source corpus is represented as a word vector, the dimensions of which are words of seed translation pairs. The same is done for corpus words in the target language, using the translated seed words as dimensions. The value of each dimension is determined by association statistics of word cooccurrence. For a test word, a vector is constructed in the same way. The labels on the dimensions are then translated, yielding an input vector in the target language vector space. We then find the closest corpus word vector in the target language vector space using the city block distance measure. This word is taken as the translation of the test word.

We went to great lengths to implement Rapp's method, but omit the details for reasons of space. Using the Wikipedia/dict.cc-based data set, we achieve 50% acc@1 when translating words from English to German. While this is somewhat lower than the performance reported by Rapp, we believe this is due to Wikipedia being more heterogeneous and less comparable than news corpora from identical time periods used by Rapp.

Publication. In conjunction with this paper we publish the benchmark for bilingual lexicon extraction described. It consists of (i) two Wikipedia dumps from October 2008 and the linguistic relations extracted from them, (ii) scripts to recreate the training and test sets from the dict.cc data base, (iii) the TS100 and TS1000 test sets, and (iv) performance numbers of Rapp's system and MEE. These can serve as baselines for future work. Note that (ii)–(iv) can be used independently of (i) – but in that case the effect of the corpus on performance would not be controlled. The data and scripts are available at `http://ifnlp.org/wiki/extern/WordGraph`

6 Results

In addition to the vector space baseline experiment described above, we conducted experiments with the SimRank model. Because TS100 only contains one translation per word, but words can have more than one valid translation, we manually extended the test set with other translations, which we verified using dict.cc and leo.org. We report the results separately for the original test set ("strict") and the extended test set in Table 4. We also experimented with *single*-edge models consisting of three separate runs on each relation.

The accuracy columns report the percentage of test cases where the correct translation was found among the top 1 (acc@1) or top 10 (acc@10) candidate words found by the translation models. Some test words are not present in the data at all; we count these as 0s when computing acc@1 and acc@10. The acc@10 measure is more useful for indicating topical similarity while acc@1 measures translation accuracy.

MRR is Mean Reciprocal Rank of correct translations: $\frac{1}{n}\sum_i^n \frac{1}{\text{rank}_i}$ (Voorhees and Tice, 1999). MRR is a more fine-grained measure than acc@n,

	TS100, strict			TS100, extended			TS1000		
	acc@1	acc@10	MRR	acc@1	acc@10	MRR	acc@1	acc@10	MRR
baseline	.50	.67	.56	.54	.70	.60	.33	.56	.41
single	.44	.67	.52	.49	.68	.56	.40‡	.70‡	.50
MEE	.52	.79†	.62	.58	.82†	.68	**.48**‡	.76‡	.58

Table 4: Results compared to baseline*

e.g., it will distinguish ranks 2 and 10. All MRR numbers reported in this paper are consistent with acc@1/acc@10 and support our conclusions.

The results for acc@1, the measure that most directly corresponds to utility in lexicon extraction, show that the SimRank-based models outperform the vector space baseline – only slightly on TS100, but significantly on TS1000. Using the various relations separately (single) already yields a significant improvement compared to the baseline. Using all relations in the integrated MEE model further improves accuracy. With an acc@1 score of 0.48, MEE outperforms the baseline by .15 compared to TS1000. This shows that a combination of several sources of information is very valuable for finding the correct translation.

MEE outperforms the baseline on TS1000 for all parts of speech, but performs especially well compared to the baseline for adjectives and verbs (see Table 5). It has been suggested that vector space models perform best for nouns and poorly for other parts of speech. Our experiments seem to confirm this. In contrast, MEE exhibits good performance for nouns and adjectives and a marked improvement for verbs.

On acc@10, MEE is consistently better than the baseline, on both TS100 and TS1000. All three differences are statistically significant.

6.1 Relation Comparison

Table 5 compares baseline, single-edge and MEE accuracy for the three parts of speech covered. Each single-edge experiment can compute noun similarity; for adjectives and verbs, only amod, dobj and MEE can be used.

Performance for nouns varies greatly depending on the relation used in the model. ncrd performs best, while dobj shows the worst performance. We hypothesize that dobj performs badly because (i) many verbs are semantically non-restrictive with respect to their arguments, (e.g., *use*, *contain* or *include*) and as a result semantically unrelated nouns become similar because they share the same verb as a neighbor; (ii) light verb constructions (e.g., *take a walk* or *give an account*) dilute the extracted relations; and (iii) dobj is the only relation we extracted with a syntactic parser. The parser was trained on newswire text, a genre that is very different from Wikipedia. Hence, parsing is less robust than the relatively straightforward POS patterns used for the other relations.

Similarly, many semantically non-restrictive adjectives such as *first* and *new* can modify virtually any noun, diluting the quality of the amod source. We conjecture that ncrd exhibits the best performance because there are fewer semantically non-restrictive nouns than non-restrictive adjectives and verbs.

MEE performance for nouns (.45) is significantly better than that of the single-edge models. The information about nouns that is contained in the verb-object and adjective-noun data is integrated in the model and helps select better translations. This, however, is only true for the noun

		noun	adj	verb	all
TS100	baseline	.55	.43	.29	.50
	amod	.15	.71	-	.30
	ncrd	.34	-	-	.22
	dobj	.02	-	.43	.04
	MEE	.45	.71	.43	.52
TS1000	baseline	.42	.26	.18	.33
	MEE	.53	.55	.27	.48

Table 5: Relation comparison, acc@1

*We indicate statistical significance at the $\alpha = 0.05$ (†) and 0.01 level (‡) when compared to the baseline. We did not calculate significance for MRR.

source	acc@1	acc@10
dobj	.02	.10
amod	.15	.37
amod+dobj	.22	.43
ncrd+dobj	.32	.65
ncrd	.34	.60
ncrd+amod	.49	.74
MEE	.45	.77

Table 6: Accuracy of sources for nouns

node type, the "pivot" node type that takes part in edges of all three types. For adjectives and verbs, the performance of MEE is the same as that of the corresponding single-edge model.

We ran three additional experiments each of which combines only two of the three possible sources for noun similarity, namely ncrd+amod, ncrd+dobj and amod+dobj and performed strict evaluation (see Table 6). We found that in general combination increases performance except for ncrd+dobj vs. ncrd. We attribute this to the lack of robustness of dobj mentioned above.

6.2 Comparison MEE vs. All-in-one

An alternative to MEE is to use untyped edges in one large graph. In this *all-in-one* model (AIO), we connect two nodes with an edge if they are linked by any of the different linguistic relations. While MEE consists of small adjacency matrices for each type, the two adjacency matrices for AIO are much larger. This leads to a much denser similarity matrix taking up considerably more memory. One reason for this is that AIO contains similarity entries between words of different parts of speech that are 0 (and require no memory in a sparse matrix representation) in MEE.

Since AIO requires more memory, we had to filter the data much more strictly than before to be able to run an experiment. We applied the following stricter thresholds on relationships to obtain a small graph: 5 instead of 3 for adjective-noun pairs, and 3 instead of 0 for verb-object pairs, thereby reducing the total number of edges from 2.1M to 1.4M. We also applied threshold sieving (see Section 4) with $\delta = 10^{-10}$ for AIO. The results on TS100 (strict evaluation) are reported in Table 7. For comparison, MEE was also run on the smaller graph. Performance of the two models is very similar, with AIO being slightly better (not significant). The slight improvement does not justify the increased memory requirements. MEE is able to scale to more nodes and edge types, which allows for better coverage and performance.

	$\textbf{MEE}_{small}$	$\textbf{AIO}_{small}$
acc@1	.51	.52
acc@10	.72	.75
MRR	.62	.59

Table 7: MEE vs. AIO

7 Analysis and Discussion

Error analysis. We examined the cases where a reference translation was not at the top of the suggested list of translation candidates. There are a number of elements in the translation process that can cause or contribute to this behavior.

Our method sometimes picks a cohyponym of the correct translation. In many of these cases, the correct translation is in the top 10 (together with other words from the same semantic field). For example, the correct translation of *moon*, *Mond*, is second in a list of words belonging to the semantic field of celestial phenomena: Komet *(comet)*, **Mond** *(moon)*, Planet *(planet)*, Asteroid *(asteroid)*, Stern *(star)*, Galaxis *(galaxy)*, Sonne *(sun)*, ... While this behavior is undesirable for strict lexicon extraction, it can be exploited for other tasks, e.g. cross-lingual semantic relatedness (Michelbacher et al., 2010).

Similarly, the method sometimes puts the antonym of the correct translation in first place. For example, the translation for *swift* (*schnell*) is in second place behind *langsam* (*slow*). Based on the syntactic relations we use, it is difficult to discriminate between antonyms and semantically similar words if their syntactic distributions are similar.

Ambiguous source words also pose a problem for the system. The correct translation of *square* (the geometric shape) is *Quadrat*. However, 8 out of its top 10 translation candidates are related to the *location* sense of *square*. The other two are geometric shapes, *Quadrat* being listed second. This is only a concern for strict evaluation, since correct translations of a different sense were included in the extended test set.

bed is also ambiguous (piece of furniture vs. river bed). This introduces translation candidates from the geographical domain. As an additional source of errors, a number of *bed*'s neighbors from the furniture sense have the German translation *Bank* which is ambiguous between the furniture sense and the financial sense. This ambiguity in the target language German introduces spurious translation candidates from the financial domain.

Discussion. The error analysis demonstrates that most of the erroneous translations are words that are incorrect, but that are related, in some obvious way, to the correct translation, e.g. by cohyponymy or antonymy. This suggests another application for bilingual lexicon extraction. One of the main challenges facing statistical machine translation (SMT) today is that it is difficult to distinguish between minor errors (e.g., incorrect word order) and major errors that are completely implausible and undermine the users' confidence in the machine translation system. For example, at some point Google translated "sarkozy sarkozy sarkozy" into "Blair defends Bush". Since bilingual lexicon extraction, when it makes mistakes, extracts closely related words that a human user can understand, automatically extracted lexicons could be used to discriminate smaller errors from grave errors in SMT.

As we discussed earlier, parallel text is not available in sufficient quantity or for all important genres for many language pairs. The method we have described here can be used in such cases, provided that large monolingual corpora and basic linguistic processing tools (e.g. POS tagging) are available. The availability of parsers is a more stringent constraint, but our results suggest that more basic NLP methods may be sufficient for bilingual lexicon extraction.

In this work, we have used a set of seed translations (unlike e.g., Haghighi et al. (2008)). We believe that in most real-world scenarios, when accuracy and reliability are important, seed lexica will be available. In fact, seed translations can be easily found for many language pairs on the web. Although a purely unsupervised approach is perhaps more interesting from an algorithmic point of view, the semisupervised approach taken in this paper may be more realistic for applications.

In this paper, we have attempted to reimplement Rapp's system as a baseline, but have otherwise refrained from detailed comparison with previous work as far as the accuracy of results is concerned. The reason is that none of the results published so far are easily reproducible. While previous publications have tried to infer from differences in performance numbers that one system is better than another, these comparisons have to be viewed with caution since neither the corpora nor the gold standard translations are the same. For example, the paper by Haghighi et al. (2008) (which demonstrates how orthography and contextual information can be successfully used) reports 61.7% accuracy on the 186 most confident predictions of nouns. But since the evaluation data sets are not publicly available it is difficult to compare other work (including our own) with this baseline. We simply do not know how methods published so far stack up against each other.

For this reason, we believe that a benchmark is necessary to make progress in the area of bilingual lexicon extraction; and that our publication of such a benchmark as part of the research reported here is an important contribution, in addition to the linguistically grounded extraction and the new graph-theoretical method we present.

8 Summary

We have presented a new method, based on graph theory, for bilingual lexicon extraction without relying on resources with limited availability like parallel corpora. We have shown that with this graph-theoretic framework, information obtained by linguistic analysis is superior to cooccurrence data obtained without linguistic analysis. We have presented multi-edge extraction (MEE), a scalable graph algorithm that combines different linguistic relations in a modular way. Finally, progress in bilingual lexicon extraction has been hampered by the lack of a common benchmark. We publish such a benchmark with this paper and the performance of MEE as a baseline for future research.

9 Acknowledgement

This research was funded by the *German Research Foundation* (DFG) within the project *A graph-theoretic approach to lexicon acquisition*.

References

Dorow, Beate, Florian Laws, Lukas Michelbacher, Christian Scheible, and Jason Utt. 2009. A graph-theoretic algorithm for automatic extension of translation lexicons. In *EACL 2009 Workshop on Geometrical Models of Natural Language Semantics*.

Dunning, Ted. 1993. Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, 19(1):61–74.

Evert, Stefan. 2004. *The Statistics of Word Cooccurrences - Word Pairs and Collocations*. Ph.D. thesis, Institut für maschinelle Sprachverarbeitung (IMS), Universität Stuttgart.

Fung, Pascale and Lo Yuen Yee. 1998. An IR approach for translating new words from nonparallel, comparable texts. In *COLING-ACL*, pages 414–420.

Garera, Nikesh, Chris Callison-Burch, and David Yarowsky. 2009. Improving translation lexicon induction from monolingual corpora via dependency contexts and part-of-speech equivalences. In *CoNLL '09: Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, pages 129–137, Morristown, NJ, USA. Association for Computational Linguistics.

Haghighi, Aria, Percy Liang, Taylor Berg-Kirkpatrick, and Dan Klein. 2008. Learning bilingual lexicons from monolingual corpora. In *Proceedings of ACL-08: HLT*, pages 771–779, Columbus, Ohio, June. Association for Computational Linguistics.

Jeh, Glen and Jennifer Widom. 2002. Simrank: A measure of structural-context similarity. In *KDD '02*, pages 538–543.

Koehn, Philipp and Kevin Knight. 2002. Learning a translation lexicon from monolingual corpora. In *Proceedings of the ACL-02 Workshop on Unsupervised Lexical Acquisition*, pages 9–16.

Lizorkin, Dmitry, Pavel Velikhov, Maxim N. Grinev, and Denis Turdakov. 2008. Accuracy estimate and optimization techniques for simrank computation. *PVLDB*, 1(1):422–433.

Michelbacher, Lukas, Florian Laws, Beate Dorow, Ulrich Heid, and Hinrich Schütze. 2010. Building a cross-lingual relatedness thesaurus using a graph similarity measure. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, Valletta, Malta, may.

Rapp, Reinhard. 1999. Automatic identification of word translations from unrelated English and German corpora. In *COLING 1999*.

Schmid, Helmut. 1994. Probabilistic part-of-speech tagging using decision trees. In *Proceedings of the International Conference on New Methods in Language Processing*, pages 44–49.

Schmid, Helmut. 2004. Efficient parsing of highly ambiguous context-free grammars with bit vectors. In *COLING '04*, page 162.

Voorhees, Ellen M. and Dawn M. Tice. 1999. The TREC-8 question answering track evaluation. In *Proceedings of the 8th Text Retrieval Conference*.

A Post-processing Approach to Statistical Word Alignment Reflecting Alignment Tendency between Part-of-speeches

Jae-Hee Lee[1], Seung-Wook Lee[1], Gumwon Hong[1],
Young-Sook Hwang[2], Sang-Bum Kim[2], Hae-Chang Rim[1]
[1]Dept. of Computer and Radio Communications Engineering, Korea University
[2]Institute of Future Technology, SK Telecom
[1]{jlee,swlee,gwhong,rim}@nlp.korea.ac.kr,
[2]{yshwang,sangbum.kim}@sktelecom.com

Abstract

Statistical word alignment often suffers from data sparseness. Part-of-speeches are often incorporated in NLP tasks to reduce data sparseness. In this paper, we attempt to mitigate such problem by reflecting alignment tendency between part-of-speeches to statistical word alignment. Because our approach does not rely on any language-dependent knowledge, it is very simple and purely statistic to be applied to any language pairs. End-to-end evaluation shows that the proposed method can improve not only the quality of statistical word alignment but the performance of statistical machine translation.

1 Introduction

Word alignment is defined as mapping corresponding words in parallel text. A word aligned parallel corpora are very valuable resources in NLP. They can be used in various applications such as word sense disambiguation, automatic construction of bilingual lexicon, and statistical machine translation (SMT). In particular, the initial quality of statistical word alignment dominates the quality of SMT (Och and Ney 2000; Ganchev et al., 2008); almost all current SMT systems basically refer to the information inferred from word alignment result.

One of the widely used approaches to statistical word alignment is based on the IBM models (Brown et al., 1993). IBM models are constructed based on words' co-occurrence and positional information. If sufficient training data are given, IBM models can be successfully applied to any language pairs. However, for minority language pairs such as English-Korean and Swedish-Japanese, it is very difficult to obtain large amounts of parallel corpora. Without sufficient amount of parallel corpus, it is very difficult to learn the correct correspondences between words that infrequently occur in the training data.

Part-of-speeches (POS), which represent morphological classes of words, can give valuable information about individual words and their neighbors. Identifying whether a word is a noun or a verb can let us predict which words are likely to be mapped in word alignment and which words are likely to occur in its vicinity in target sentence generation.

Many studies incorporate POS information in SMT. Some researchers perform POS tagging on their bilingual training data (Lee et al., 2006; Sanchis and Sánchez, 2008). Some of them replace individual words as new words, such as in "word/POS" form, producing new, extended vocabulary. The advantage of this approach is that POS information can help to resolve lexical ambiguity and thus improve translation quality.

On the other hand, Koehn et al. (2007) propose a factored translation model that can incorporate any linguistic factors including POS information in phrase-based SMT. The model provides a generalized representation of a translation model, because it can map multiple source and target factors.

Although all of these approaches are shown to improve SMT performance by utilizing POS information, we observe that the influence is virtually marginal in two ways:

Coling 2008: Poster Volume, pages 623–629,
Beijing, August 2010

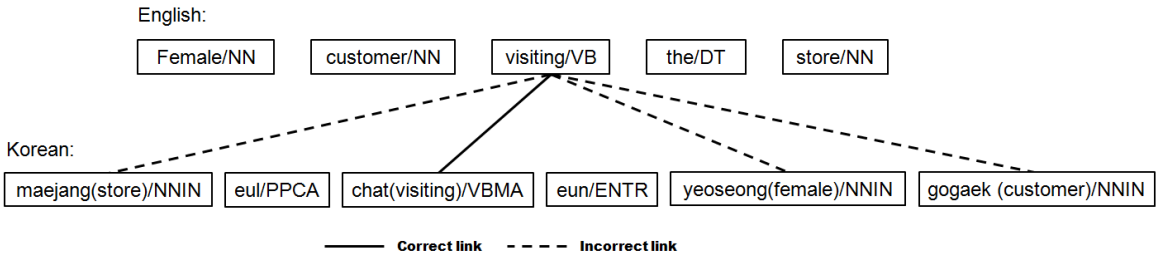


Figure 1. An example of inaccurate word alignment

1) The POS information tagged to each word may help to disambiguate in selecting word correspondences, but the increased vocabulary can also make the training data more sparse.
2) The factored translation model may help to effectively handle out-of-vocabulary (OOV) by incorporating many linguistic factors, but it still crucially relies on the initial quality of word alignment that will dominate the translation probabilities.

This paper focuses on devising a better method for incorporating POS information in word alignment. It attempts to answer the following questions:

1) Can the information regarding POS alignment tendency affect the post-processing of word alignment?
2) Can the result of word alignment affected by such information help improving the quality of SMT?

2 POS Alignment Tendency

Despite the language pairs, words with similar POSs often correspond to each other in statistical word alignment. Similarly, words with different POSs are seldom aligned. For example, Korean proper nouns very often align with English proper nouns very often but seldom align with English adverbs. We believe that this phenomenon occurs not only on English-Korean pairs but also on most of other language pairs.

Thus, in this study we hypothesize that all source language (SL) POSs have some relationship with target language (TL) POSs. Figure 1 exemplifies some results of using the IBM Models in English-Korean word alignment. As can be seen in the figure, the English word "*visiting*" is incorrectly and excessively aligned to four Korean morphemes "*maejang*", "*chat*", "*yeoseong*", and "*gogaek*". One reason for this is the sparseness of the training data; the only correct Korean morpheme "*chat*" does not sufficiently co-occur with "*visiting*" in the training data. However, it is generally believed that an English verb is more likely aligned to a Korean verb rather than a Korean noun. Likewise, we suppose that among many POSs, there are strong relationships between similar POSs and relatively weak relationships between different POSs. We hypothesize that the discovery of such relationships in advance can lead to better word alignment results.

In this paper, we propose a new method to obtain the relationship from word alignment results. The relationships among POSs, henceforth the POS alignment tendency, can be identified by the probability of the given POS pairs' alignment result where the source language POS and the target language POS co-occur in bilingual sentences. We formulate this idea using the maximum likelihood estimation as follows:

$$P\big(align = true \big| pos(f), pos(e)\big) = \frac{count(align = true | pos(f), pos(e))}{\sum_{k\in\{true,false\}} count(align = k, pos(f), pos(e))}$$

where f and e denote source word and target word respectively. *count()* is a function that returns the number of co-occurrence of f and e when they are aligned (or not aligned). Then, we adjust the formula with the existing alignment score between f and e.

$$Score(f,e) = \lambda P_{IBM}(f|e) + (1-\lambda)P\big(align = true \big| pos(f), pos(e)\big)$$

where $P_{IBM}(f \mid e)$ indicates the alignment probability estimated by the IBM models. λ is a weighting parameter to interpolate the reliabilities of both alignment factors. In the expe-

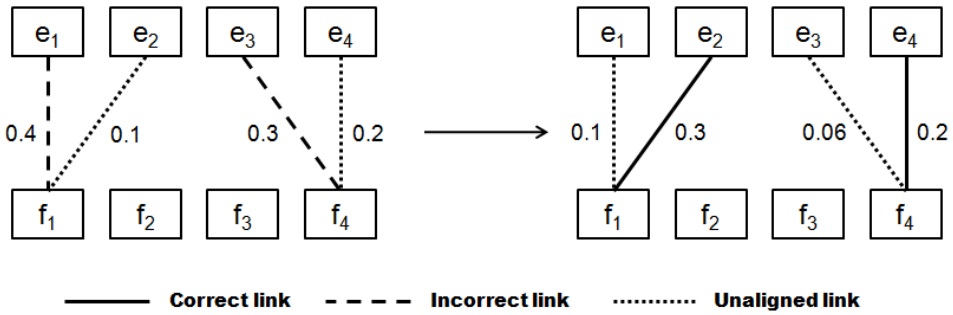


Figure 2. An example of word alignment modification

riment, λ is empirically set to improve the word alignment performance (λ=0.5).

3 Modifying Alignment

Based on the new scoring scheme as introduced in the previous section, we modify the result of the initial word alignment. The modification is performed in the following procedure:

1. For each source word *f* that has out-bound alignment link other than *null*,
2. Find the target word *e* that has the maximum alignment score according to the proposed alignment adjustment measure, and change the alignment result by mapping *f* to *e*.

This modification guarantees that the number of alignment does not change; the algorithm is designed to minimize the risk by maintaining the fertility of a word estimated by the IBM Model. Figure 2 illustrates the result before and after the alignment modification. Incorrectly links from e_1 and e_3 are deleted and missing links from e_2 and e_4 are generated during this alignment modification.

The alignment modification through the reflection of POS alignment tendency is performed on both *e*-to-*f* and *f*-to-*e* bidirectional word alignments. The bidirectional word alignment results are then symmetrized.

4 Experiments

In this paper, we attempt to reflect the POS alignment tendency in improving the word alignment performance. This section provides the experimental setup and the results that demonstrate whether the proposed approach can improve the statistical word alignment performance.

We collected bilingual texts from major bilingual news broadcasting sites. 500K sentence pairs are collected and refined manually to construct correct parallel sentences pairs. The same number of monolingual sentences is also used from the same sites to train Korean language. We also prepared a subset of the bilingual text with the size of 50K to show that the proposed model is very effective when the training set is small.

In order to evaluate the performance of word alignment, we additionally constructed a reference set with 400 sentence pairs. The evaluation is performed using precision, recall, and F-score. We use the GIZA++ toolkit for word alignment as well as four heuristic symmetrizations: intersection, union, grow-diag-final, and grow-diag (Och, 2000).

4.1 Word Alignment

We now evaluate the effectiveness of the proposed word alignment method. Table 1 and 2 report the experimental results by adding POS information to the parallel corpus. “Lexical” denotes the result of conventional word alignment produced by GIZA++. No pre-processing or post-processing is applied in this result. “Lemma/POS” is the result of word alignment with the pre-processing introduced Lee et al. (2006). Compared to the result, lemmatized lexical and POS tags are proven to be useful information for word alignment. “Lemma/POS” consistently outperforms “Lexical” despite the symmetrization heuristics in terms of precision, recall and F-score. We expect this improvement is benefited from the alleviated data sparseness by using lemmatized lexical and POS tags rather than using the lexical itself.

	Alignment heuristic	Precision	Recall	F-score
Lexical	Intersection	94.0%	50.8%	**66.0%**
	Union	53.2%	81.2%	64.3%
	Grow-diag-final	54.6%	80.9%	65.2%
	Grow-diag	60.9%	67.2%	63.9%
Lemma/POS	Intersection	95.8%	55.3%	**70.1%**
	Union	58.1%	83.3%	68.4%
	Grow-diag-final	59.7%	83.0%	69.5%
	Grow-diag	67.0%	71.6%	69.2%
Lemma/POS + POS alignment tendency	Intersection	96.1%	63.5%	76.5%
	Union	67.4%	85.1%	75.2%
	Grow-diag-final	69.8%	84.9%	76.6%
	Grow-diag	80.0%	77.0%	**78.5%**

Table 1. The performance of word alignment using small training set (50k pairs)

Experimental Setup	Alignment heuristic	Precision	Recall	F-score
Lexical	Intersection	96.8%	64.9%	**77.7%**
	Union	66.6%	87.4%	75.6%
	Grow-diag-final	67.8%	87.1%	76.2%
	Grow-diag	74.4%	79.2%	76.7%
Lemma/POS	Intersection	97.3%	66.2%	78.8%
	Union	70.7%	89.0%	78.8%
	Grow-diag-final	72.1%	88.8%	79.6%
	Grow-diag	78.8%	80.5%	**79.7%**
Lemma/POS + POS alignment tendency	Intersection	97.2%	69.3%	80.9%
	Union	73.9%	86.7%	79.8%
	Grow-diag-final	75.6%	86.4%	80.7%
	Grow-diag	85.2%	81.5%	**83.4%**

Table 2. The performance of word alignment using a large training set (500k pairs)

Experimental Setup	Symmetrization Heuristic	BLEU(50k)	BLEU (500k)
Lexical	Intersection	20.1%	29.2%
	Union	18.6%	27.2%
	Grow-diag-final	19.9%	27.7%
	Grow-diag	**20.2**%	**29.4**%
Lemma/POS	Intersection	20.3%	26.4%
	Union	18.5%	27.8%
	Grow-diag-final	20.1%	29.2%
	Grow-diag	**20.4**%	**30.8**%
Factored Model (Lemma, POS)	Intersection	20.5%	30.0%
	Union	18.1%	27.5%
	Grow-diag-final	20.3%	28.2%
	Grow-diag	**20.9**%	**31.1**%
Lemma/POS + POS alignment tendency	Intersection	**21.8**%	**29.3**%
	Union	19.5%	27.2%
	Grow-diag-final	21.3%	28.4%
	Grow-diag	20.8%	29.1%

Table 3. The performance of translation

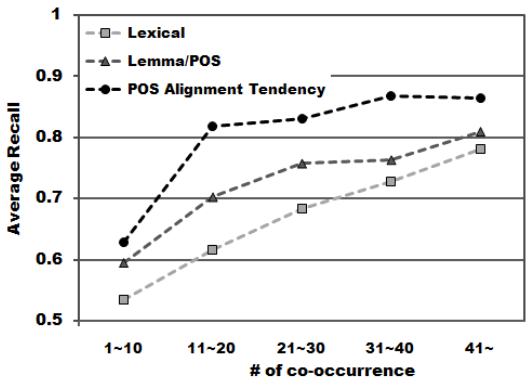


Figure 3. Average recall of word alignment pairs according to the number of their co-occurrence

Since lemmatized lexical and POS tags are shown to be useful, our post-processing method is applied to "Lemma/POS".

The experimental results show that the proposed method consistently improves word alignment in terms of F-score. It is interesting that the proposed method improves the recall of the intersection result and the precision of the union result. Thus, the proposed method achieves the best alignment performance.

As can be seen in Table 1 and 2, our method consistently improves the performance of word alignment despite the size of training data. In a small data set, the improvement of our method is much higher than that in a large set. This implies that our method is more helpful when the training data set is insufficient.

We investigate whether the proposed method actually alleviates the data sparseness problem by analyzing the aligned word pairs of low co-occurrence frequency. There are multiple word pairs that share the same number of co-occurrence in the corpus. For example, let us assume that "report-*bogoha*", "newspaper-*sinmun*" and "China-*jungguk*" pairs are co-occurred 1,000 times. We can calculate the mean of their individual recalls. We refer to this new measurement as average recall. The average recalls of these pairs are relatively higher than those of pairs with low co-occurrence frequency such as "food-*jinji*" and "center-*chojeom*" pairs. These pairs are difficult to be linked, because the word alignment model suffers from data sparseness when estimating their translation probability.

Figure 3 shows the average recall according to the number of co-occurrence. We can observe that the word alignment model tends to link word pairs more correctly if they are more frequently co-occurred. Both "Lemma/POS" and our method consistently show higher average recall throughout all frequencies, and the proposed method shows the best performance. It is also notable that the both "Lemma/POS" and our method achieve much more improvement for low co-occurrence frequencies (e.g., 11~40). This implies that the proposed method incorporates POS information more effectively than the previous method, since the proposed method achieves much higher average recall.

4.2 Statistical Machine Translation

Next, we examine the effect of the improvement of the word alignment on the translation quality. For this, we built some SMT systems with the word alignment results. We use the Moses toolkit for translation (Koehn et al., 2007). Moses is an implementation of phrase-based statistical machine translation model that has shown a state-of-the-art performance in various evaluation sets. We also perform the evaluation of the Factored model (Koehn et al., 2007) using Moses.

To investigate how the improved word alignment affect the quality of machine translation, we calculate the BLEU score for translation results with different word alignment settings as shown in Table 3. First of all, we can easily conclude that the quality of the translation is strongly dominated by the size of the training data. We can also find that the quality of the translation is correlated to the performance of the word alignment.

For a small test set, the proposed method achieved the best performance in terms of BLEU (21.8%). For a larger test set, however, the proposed method could not improve the performance of the translation with better word alignment. It is not feasible to investigate the factors that affect this deterioration, since Moses is a black box module to our system. The training of the phrase-based SMT model involves the extraction of phrases, and the result of word alignment is reflected within this process. When the training data is small, the number of extracted phrases is also apparently small. However, abundant phrases are extracted from a large amount of training data. In this case, we hypothesize that the most plausible

Rank	IBM Model			POS Alignment Tendency		
	translation	$P_{IBM}(f\|e)$	#co-occur	translation	score(f, e)	#co-occur
1	bob/NNP	0.348	83	bob/NNP	0.214	83
2	rice/NN	0.192	73	rice/NN	0.136	73
3	*eat/VB*	0.107	57	meal/NN	0.078	43
4	meal/NN	0.075	43	food/NN	0.062	29
5	food/NN	0.043	29	*eat/VB*	0.061	57
6	bob/NN	0.038	10	bob/NN	0.059	10
7	*feed/VB*	0.010	7	*living/NN*	0.045	4
8	*cook/VB*	0.010	9	dinner/NN	0.044	10
9	*living/NN*	0.008	4	**bread/NN**	0.044	9
10	dinner/NN	0.008	10	**breakfast/NN**	0.043	6

Table 4. Top 10 translations for Korean word "bap" (food).

phrases are already obtained, and the effect of more accurate word alignment seems insignificant. More thorough analysis of this is remained as future work.

4.3 Acquisition of Bilingual Dictionary

One of the most applications of word alignment is the construction of bilingual dictionaries. By using word alignment, we can collect a (ranked) list of bilingual word pairs. Table 4 reports the top 10 translations (the most acceptable target words to align) for Korean word "*bap*" (food). The table contains the probabilities estimated by the IBM Models, the adjusted scores, and the number of co-occurrence, respectively. Italicized translations are in fact incorrect translations. Highlighted ones are new translation candidates that are correct. As can be seen in the table, the proposed approach shows a positive effect of raising new and better candidates for translation. For example, "bread" and "breakfast" have come up to the top 10 translations. This demonstrates that the low co-occurrences of "bap" with "bread" and "breakfast" are not suitably handled by alignments solely based on lexicals. However, the proposed approach ranks them at higher positions by reflecting the alignment tendency of POSs.

5 Conclusion

In this paper, we propose a new method for incorporating the POS alignment tendency to improve traditional word alignment model in post processing step. Experimental results show that the proposed method helps to alleviate the data sparseness problem especially when the training data is insufficient.

It is still difficult to conclude that better word alignment always leads to better translation. We plan on investigating the effectiveness of the proposed method using other translation system, such as Hiero (Chiang et al., 2005). We also plan to incorporate our method into other effective models, such as Factored translation model.

References

David Chiang et al., 2005. *The Hiero machine translation system: Extensions, evaluation, and analysis.* In Proc. of HLT-EMLP:779–786, Oct.

Franz Josef Och. 2000. *Giza++: Training of statistical translation models.* Available at http://www-i6.informatik.rwthaachen.de/ ~och/software/GIZA++.html.

Franz Josef Och & Hermann Ney. 2003. *A Systematic Comparison of Various Statistical Alignment Models.* Computational Linguistics 29 (1):19-51.

G. Sanchis and J.A. Sánchez. *Vocabulary extension via POS information for SMT*. In Mixing Approaches to Machine Translation, 2008.

Jonghoon Lee, Donghyeon Lee and Gary Geunbae Lee. *Improving Phrase-based Korean-English Statistical Machine Translation.* INTERSPEECH 2006.

Kuzman Ganchev, Joao V. Graca and Ben Taskar. 2008. *Better Alignments = Better Translations?* Proceedings of ACL-08: HLT: 986–993.

Peter F. Brown et al.,1993. *The Mathematics of Statistical Machine Translation: Parameter Estimation.* Computational Linguistics 9(2): 263-311

Philipp Koehn and Hieu Hoang. *Factored Translation Models.* EMNLP 2007.

Phillipp Koehn et al., 2007. *Moses: Open source toolkit for statistical machine translation.*In Proceedings of the Annual Meeting of the Association for Computational Linguistics, demonstation session.

Enhancing Multi-lingual Information Extraction via Cross-Media Inference and Fusion

Adam Lee, Marissa Passantino, Heng Ji
Computer Science Department
Queens College and Graduate Center
City University of New York
hengji@cs.qc.cuny.edu

Guojun Qi, Thomas Huang
Department of Electrical and Computer
Engineering & Beckman Institute
University of Illinois at Urbana-Champaign
huang@ifp.uiuc.edu

Abstract

We describe a new information fusion approach to integrate facts extracted from cross-media objects (videos and texts) into a coherent common representation including multi-level knowledge (concepts, relations and events). Beyond standard information fusion, we exploited video extraction results and significantly improved text Information Extraction. We further extended our methods to multi-lingual environment (English, Arabic and Chinese) by presenting a case study on cross-lingual comparable corpora acquisition based on video comparison.

1 Introduction

An enormous amount of information is widely available in various data modalities (e.g. speech, text, image and video). For example, a Web news page about "Health Care Reform in America" is composed with texts describing some events (e.g., Final Senate vote for the reform plans, Obama signs the reform agreement), images (e.g., images about various government involvements over decades) and videos/speech (e.g. Obama's speech video about the decisions) containing additional information regarding the real extent of the events or providing evidence corroborating the text part. These cross-media objects exist in redundant and complementary structures, and therefore it is beneficial to fuse information from various data modalities. The goal of our paper is to investigate this task from both mono-lingual and cross-lingual perspectives.

The processing methods of texts and images/videos are typically organized into two separate pipelines. Each pipeline has been studied separately and quite intensively over the past decade. It is critical to move away from single media processing, and instead toward methods that make multiple decisions jointly using cross-media inference. For example, video analysis allows us to find both entities and events in videos, but it's very challenging to specify some fine-grained semantic types such as proper names (e.g. "Obama Barack") and relations among concepts; while the speech embedded and the texts surrounding these videos can significantly enrich such analysis. On the other hand, image/video features can enhance text extraction. For example, entity gender detection from speech recognition output is challenging because of entity mention recognition errors. However, gender detection from corresponding images and videos can achieve above 90% accuracy (Baluja and Rowley, 2006). In this paper, we present a case study on gender detection to demonstrate how text and video extractions can boost each other.

We can further extend the benefit of cross-media inference to cross-lingual information extraction (CLIE). Hakkani-Tur et al. (2007) found that CLIE performed notably worse than monolingual IE, and indicated that a major cause was the low quality of machine translation (MT). Current statistical MT methods require large and manually aligned parallel corpora as input for each language pair of interest. Some recent work (e.g. Munteanu and Marcu, 2005; Ji, 2009) found that MT can benefit from multi-lingual comparable corpora (Cheung and Fung, 2004), but it is time-consuming to identify pairs of comparable texts; especially when there is

Coling 2008: Poster Volume, pages 630–638,
Beijing, August 2010

lack of parallel information such as news release dates and topics. However, the images/videos embedded in the same documents can provide additional clues for similarity computation because they are 'language-independent'. We will show how a video-based comparison approach can reliably build large comparable text corpora for three languages: English, Chinese and Arabic.

2 Baseline Systems

We apply the following state-of-the-art text and video information extraction systems as our baselines. Each system can produce reliable confidence values based on statistical models.

2.1 Video Concept Extraction

The video concept extraction system was developed by IBM for the TREC Video Retrieval Evaluation (TRECVID-2005) (Naphade et al., 2005). This system can extract 2617 concepts defined by TRECVID, such as "Hospital", "Airplane" and "Female-Person". It uses support vector machines to learn the mapping between low level features extracted from visual modality as well as from transcripts and production related meta-features. It also exploits a Correlative Multi-label Learner (Qi et al., 2007), a Multi-Layer Multi-Instance Kernel (Gu et al., 2007) and Label Propagation through Linear Neighborhoods (Wang et al., 2006) to extract all other high-level features. For each classifier, different models are trained on a set of different modalities (e.g., the color moments, wavelet textures, and edge histograms), and the predictions made by these classifiers are combined together with a hierarchical linearly-weighted fusion strategy across different modalities and classifiers.

2.2 Text Information Extraction

We use a state-of-the-art IE system (Ji and Grishman, 2008) developed for the Automatic Content Extraction (ACE) program[1] to process texts and automatic speech recognition output. The pipeline includes name tagging, nominal mention tagging, coreference resolution, time expression extraction and normalization, relation extraction and event extraction. Entities include coreferred persons, geo-political entities (GPE), locations, organizations, facilities, vehicles and weapons; relations include 18 types (e.g. "*a town some 50 miles south of Salzburg*" indicates a located relation.); events include the 33 distinct event types defined in ACE 2005 (e.g. "*Barry Diller on Wednesday quit as chief of Vivendi Universal Entertainment.*" indicates a "personnel-start" event). Names are identified and classified using an HMM-based name tagger. Nominals are identified using a maximum entropy-based chunker and then semantically classified using statistics from ACE training corpora. Relation extraction and event extraction are also based on maximum entropy models, incorporating diverse lexical, syntactic, semantic and ontological knowledge.

3 Mono-lingual Information Fusion and Inference

3.1 Mono-lingual System Overview

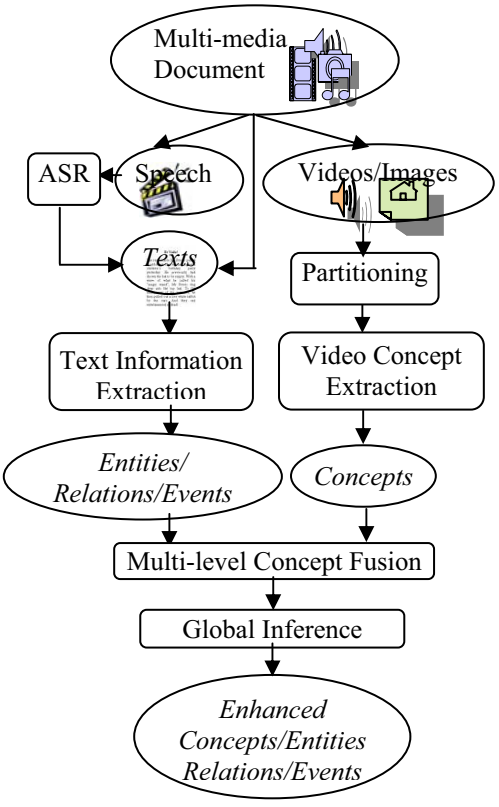


Figure 1. Mono-lingual Cross-Media Information Fusion and Inference Pipeline

Figure 1 depicts the general procedure of our mono-lingual information fusion and inference

[1] http://www.nist.gov/speech/tests/ace/

approach. After we apply two baseline systems to the multi-media documents, we use a novel multi-level concept fusion approach to extract a common knowledge representation across texts and videos (section 3.2), and then apply a global inference approach to enhance fusion results (section 3.3).

3.2 Cross-media Information Fusion

- ***Concept Mapping***

For each input video, we apply automatic speech recognition to obtain background texts. Then we use the baseline IE systems described in section 2 to extract concepts from texts and videos. We construct mappings on the overlapped facts across TRECVID and ACE. For example, "LOC.Water-Body" in ACE is mapped to "Beach, Lakes, Oceans, River, River_Bank" in TRECVID.

Due to different characteristics of video clips and texts, these two tasks have quite different granularities and focus. For example, "PER.Individual" in ACE is an open set including arbitrary names, while TRECVID only covers some famous proper names such as "Hu_Jintao" and "John_Edwards". Geopolitical entities appear very rarely in TRECVID because they are more explicitly presented in background texts. On the other hand, TRECVID defined much more fine-grained nominals than ACE, for example, "FAC.Building-Grounds" in ACE can be divided into 52 possible concept types such as "Conference_Buildings" and "Golf_Course" because they can be more easily detected based on video features. We also notice that TRECVID concepts can include multiple levels of ACE facts, for example "WEA_Shooting" concept can be separated into "weapon" entities and "attack" events in ACE. These different definitions bring challenges to cross-media fusion but also opportunities to exploit complementary facts to refine both pipelines. We manually resolved these issues and obtained 20 fused concept sets.

- ***Time-stamp based Multi-level Projection***

After extracting facts from videos and texts, we conduct information fusion at all possible levels: name, nominal, coreference link, relation or event mention. We rely on the timestamp information associated with video keyframes or shots (sequential keyframes) and background speech to align concepts. During this fusion process, we compare the normalized confidence values produced from two pipelines to resolve the following three types of cases:

- Contradiction – A video fact contradicts a text fact; we only keep the fact with higher confidence.
- Redundancy – A video fact conveys the same content as (or entails, or is entailed by) a text fact; we only keep the unique parts of the facts.
- Complementary – A video fact and a text fact are complementary; we merge these two to form more complete fact sets.

- ***A Common Representation***

In order to effectively extract compact information from large amounts of heterogeneous data, we design an integrated XML format to represent the facts extracted from the above multi-level fusion. We can view this representation as a set of directed "information graphs" $G=\{G_i (V_i, E_i)\}$, where V_i is the collection of concepts from both texts and videos, and E_i is the collection of edges linking one concept to the other, labeled by relation or event attributes. An example is presented in Figure 2. This common representation is applied in both mono-lingual and multi-lingual information fusion tasks described in next sections.

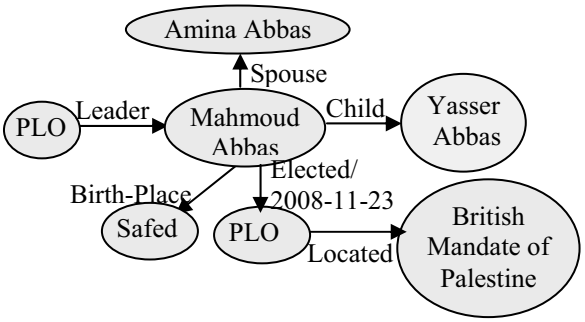


Figure 2. An example for cross-media common fact representation

3.3 Cross-media Information Inference

- ***Uncertainty Problem in Cross-Media Fusion***

However, such a simple merging approach usually leads to unsatisfying results due to uncertainty. Uncertainty in multimedia is induced from noise in the data acquisition procedure

(e.g., noise in automatic speech recognition results and low-quality camera surveillance videos) as well as human errors and subjectivity. Unstructured texts, especially those translated from foreign languages, are difficult to interpret. In addition, automatic IE systems for both videos and texts tend to produce errors.

- ***Case Study on Mention Gender Detection***

We employ cross-media inference methods to reduce uncertainty. We will demonstrate this approach on a case study of gender detection for persons. Automatic gender detection is crucial to many natural language processing tasks such as pronoun reference resolution (Bergsma, 2005). Gender detection for last names has proved challenging; Gender for nominals can be highly ambiguous in various contexts. Unfortunately most state-of-the-art approaches discover gender information without considering specific contexts in the document. The results were stored either as a knowledge base with probabilities (e.g. Ji and Lin, 2009) or as a static gazetteer (e.g. census data). Furthermore, speech recognition normally performs poorly on names, which brings more challenges to gender detection for mis-spelled names.

We consider two approaches as our baselines. The first baseline is to discover gender knowledge from Google N-grams using specific lexical patterns (e.g. "[mention] and his/her/its/their") (Ji and Lin, 2009). The other baseline is a gazetteer matching approach based on census data including person names and gender information, as used in typical text IE systems.

We introduce the third method based on male/female concept extraction from associated background videos. These concepts are detected from context-dependent features (e.g. face recognition). If there are multiple persons in one snippet associated with one shot, we propagate gender information to all instances.

We then linearly combine these three methods based on confidence values. For example, the confidence of predicting a name mention *n* as a male (M) can be computed by combining probabilities *P(n, M, method)*:

$$confidence(n,male)= \lambda_1 * P(n,M,ngram) + \lambda_2 * P(n,M,census) + \lambda_3 * P(n,M,video)$$

In this paper we used $\lambda_1=0.1$, $\lambda_2=0.1$ *and* $\lambda_3=0.8$ which are optimized from a development set.

4 Cross-lingual Comparable Corpora Acquisition

In this section we extend the information fusion approach to a task of discovering comparable corpora.

4.1 Comparable Documents

Figure 3 presents an example of cross-lingual comparable documents. They are both about the rescue activities for the Haiti earthquake.

Figure 3. An example for cross-lingual multi-media comparable documents

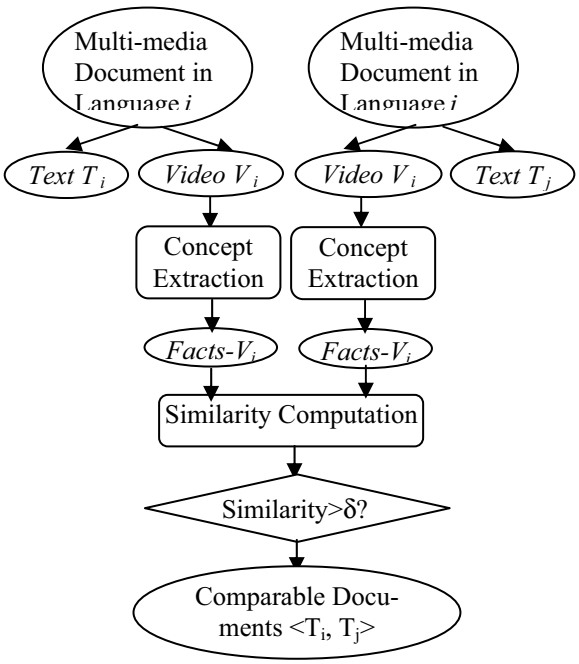


Figure 4. Cross-lingual Comparable Text Corpora Acquisition based on Video Similarity Computation

Traditional text translation based methods tend to miss such pairs due to poor translation quality of informative words (Ji et al., 2009). However, the background videos and images are language-independent and thus can be exploited to identify such comparable documents. This provides a cross-media approach to break language barrier.

4.2 Cross-lingual System Overview

Figure 4 presents the general pipeline of discovering cross-lingual comparable documents based on background video comparison. The detailed video similarity computation method is presented in next section.

4.3 Video Concept Similarity Computation

Most document clustering systems use representations built out of the lexical and syntactic attributes. These attributes may involve string matching, agreement, syntactic distance, and document release dates. Although gains have been made with such methods, there are clearly cases where shallow information will not be sufficient to resolve clustering correctly. Therefore, we should therefore expect a successful document comparison approach to exploit world knowledge, inference, and other forms of semantic information in order to resolve hard cases. For example, if two documents include concepts referring to male-people, earthquake event, rescue activities, and facility-grounds with similar frequency information, we can determine they are likely to be comparable. In this paper we represent each video as a vector of semantic concepts extracted from videos and then use standard vector space model to compute similarity.

Let A=(a_1, …$a_{|\Sigma|}$) and B=(b_1, …$b_{|\Sigma|}$) be such vectors for a pair of videos, then we use cosine similarity to compute similarity:

$$\cos(A,B) = \frac{\sum_{i=1}^{|\sum|} a_i b_i}{\sqrt{\sum_{i=1}^{|\sum|} a_i^2}\sqrt{\sum_{i=1}^{|\sum|} b_i^2}},$$

where $|\Sigma|$ contains all possible concepts. We use traditional TF-IDF (Term Frequency-Inverse Document Frequency) weights for the vector elements a_i and b_i. Let C be a unique concept, V is a video consisting of a series of k shots $V = \{S_1, ..., S_k\}$, then:

$$tf(C,V) = \sum_{i=1}^{k} tf(C,S_i) \Big/ k$$

Let $p(C, S_i)$ denote the probability that C is extracted from S_i, we define two different ways to compute term frequency $tf(C, S_i)$:

(1) $tf(C,S_i) = confidence(C,S_i)$

and

(2) $tf(C,S_i) = \alpha^{confidence(C,S_i)}$

Where $Confidence(C, S_i)$ denotes the probability of detecting a concept C in a shot S_i:

$$confidence(C,S_i) = p(C,S_i) \text{ if } p(C,S_i) > \delta,$$
otherwise 0.

Let: $df(C,S_i) = 1$ if $p(C,S_i) > \delta$, otherwise 0, assuming there are j shots in the entire corpus, we calculate *idf* as follows:

$$idf(C,V) = \log\left(j / \sum_{i=1}^{j} df(C,S_i) \right)$$

5 Experimental Results

This section presents experimental results of all the three tasks described above.

5.1 Data

We used 244 videos from TRECVID 2005 data set as our test set. This data set includes 133,918 keyframes, with corresponding automatic speech recognition and translation results (for foreign languages) provided by LDC.

5.2 Information Fusion Results

Table 1 shows information fusion results for English, Arabic and Chinese on multiple levels. It indicates that video and text extraction pipelines are complementary – almost all of the video concepts are about nominals and events; while text extraction output contains a large amount of names and relations. Therefore the results after information fusion produced much richer knowledge.

Annotation Levels		English	Chinese	Arabic
# of videos		104	84	56
Video	Concept	250880	221898	197233
Text	Name	17350	22154	20057
	Nominal	31528	21852	16253
	Relation	9645	20880	16584
	Event	31132	10348	7148

Table 1. Information Fusion Results

It's also worth noting that the number of concepts extracted from videos is similar across languages, while much fewer events are extracted from Chinese or Arabic because of speech recognition and machine translation errors. We took out 1% of the results to measure accuracy against ground-truth in TRECVID and ACE training data respectively; the mean average precision for video concept extraction is about 33.6%. On English ASR output the text-IE system achieved about 82.7% F-measure on labeling names, 80.5% F-measure on nominals (regardless of ASR errors), 66% on relations and 64% on events.

5.3 Information Inference Results

From the test set, we chose 650 persons (492 males and 158 females) to evaluate gender discovery. For baselines, we used Google n-gram (n=5) corpus Version II including 1.2 billion 5-grams extracted from about 9.7 billion sentences (Lin et al., 2010) and census data including 5,014 person names with gender information.

Since we only have gold-standard gender information on shot-level (corresponding to a snippet in ASR output), we asked a human annotator to associate ground-truth with individual persons. Table 2 presents overall precision (P), recall (R) and F-measure (F).

Methods	P	R	F
Google N-gram	89.1%	70.2%	78.5%
Census	96.2%	19.4%	32.4%
Video Extraction	88.9%	73.8%	80.6%
Combined	89.3%	80.4%	84.6%

Table 2. Gender Discovery Performance

Table 2 shows that video extraction based approach can achieve the highest recall among all three methods. The combined approach achieved statistically significant improvement on recall.

Table 3 presents some examples ("F" for female and "M" for male). We found that most speech name recognition errors are propagated to gender detection in the baseline methods, for example, "Sala Zhang" is mis-spelled in speech recognition output (the correct spelling should be "Sarah Chang") and thus Google N-gram approach mistakenly predicted it as a male. Many rare names such as "Wu Ficzek", "Karami" cannot be predicted by the baselines,

Error analysis on video extraction based approach showed that most errors occur on those shots including multiple people (males and females). In addition, since the data set is from news domain, there were many shots including reporters and target persons at the same time. For example, "Jiang Zemin" was mistakenly associated with a "female" gender because the reporter is a female in that corresponding shot.

5.4 Comparable Corpora Acquisition Results

For comparable corpora acquisition, we measured accuracy for the top 50 document pairs. Due to lack of answer-keys, we asked a bilingual human annotator to judge results manually. The evaluation guideline generally followed the definitions in (Cheung and Fung, 2004). A pair of documents is judged as comparable if they share a certain amount of information (e.g. entities, events and topics).

Without using IDF, for different parameter α and δ in the similarity metrics, the results are summarized in Figure 5. For comparison we present the results for mono-lingual and cross-lingual separately. Figure 5 indicates that as the threshold and normalization values increase, the accuracy generally improves. It's not surprising that mono-lingual results are better than cross-lingual results, because generally more videos with comparable topics are in the same language.

Mention	Google N-gram	Census	Video Extraction	Correct Answer	Context Sentence
Zhang Sala	M: 1 F: 0	-	F: 0.699 M: 0.301	F	World famous meaning violin soloist **Zhang Sala** recently again to Toronto symphony orchestra...
Peter	M: .979 F: 0.021	M: 1	M: 0.699 F: 0.301	M	Iraq, there are in Lebanon Paris pass **Peter** after 10 five Dar exile without peace...
Wu Ficzek	-		M: 0.699 F: 0.301	M	If you want to do a good job indeed **Wu Ficzek**
President	M: .953 F: 0.047	-	M: 0.704 F: 0.296	M	Labor union of Arab heritage publishers **president** to call for the opening of the Arab Book Exhibition.
Jiang Zemin	M: 1 F: 0	-	F: 0.787 M: 0.213	M	It has never stopped the including the former CPC General Secretary Jiang Zemin…
Karami	M: 1 F: 0	-	M: 0.694 F: 0.306	M	all the Gamal Ismail introduced the needs of the Akkar region, referring to the desire on the issue of the President **Karami** to give priority disadvantaged areas

Table 3. Examples for Mention Gender Detection

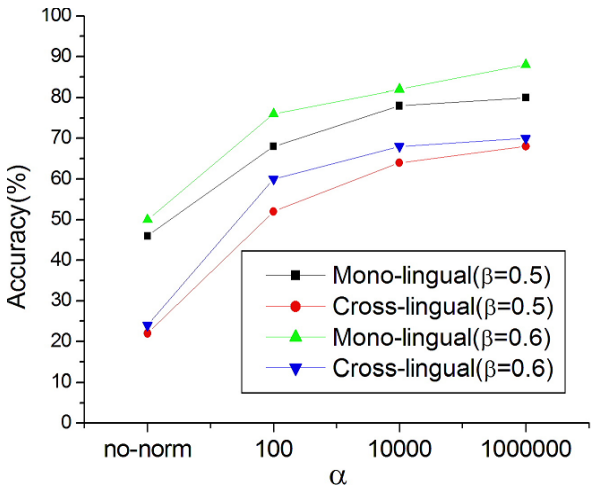


Figure 5. Comparable Corpora Acquisition without IDF

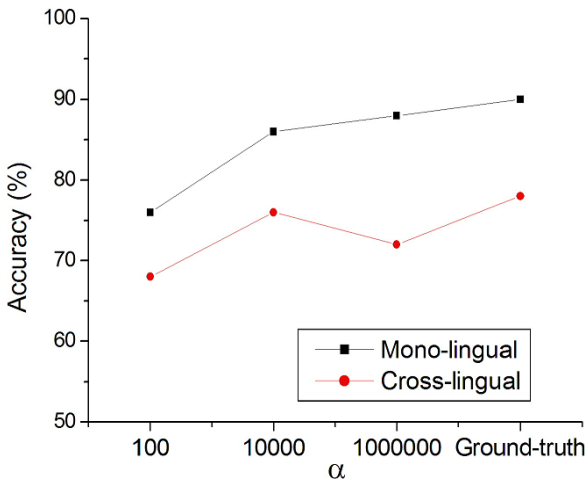


Figure 6. Comparable Corpora Acquisition with IDF (δ=0.6)

We then added IDF to the optimized threshold and obtained results in Figure 6. The accuracy for both languages was further enhanced. We can see that under any conditions our approach can discover comparable documents reliably. In order to measure the impact of concept extraction errors, we also evaluated the results for using ground-truth concepts as shown in Figure 6. Surprisingly it didn't provide much higher accuracy than automatic concept extraction, mainly because the similarity can be captured by some dominant video concepts.

6 Related Work

A large body of prior work has focused on multimedia information retrieval and document classification (e.g. Iria and Magalhaes, 2009). State-of-the-art information fusion approaches can be divided into two groups: formal "top-down" methods from the generic knowledge fusion community and quantitative "bottom-up" techniques from the Semantic Web community (Appriou et al., 2001; Gregoire, 2006). However, very limited research methods have been ex-

plored to fuse automatically extracted facts from texts and videos/images. Our idea of conducting information fusion on multiple semantic levels is similar to the kernel method described in (Gu et al., 2007).

Most previous work on cross-media information extraction focused on one single domain (e.g. e-Government (Amato et al., 2010); soccer game (Pazouki and Rahmati, 2009)) and structured/semi-structured texts (e.g. product catalogues (Labsky et al., 2005)). Saggion et al. (2004) described a multimedia extraction approach to create composite index from multiple and multi-lingual sources. We expand the task to the more general news domain including unstructured texts and use cross-media inference to enhance extraction performance.

Some recent work has exploited analysis of associated texts to improve image annotation (e.g. Deschacht and Moens, 2007; Feng and Lapata, 2008). Some recent research demonstrated cross-modal integration can provide significant gains in improving the richness of information. For example, Oviatt et al. (1997) showed that speech and pen-based gestures can provide complementary capabilities because basic subject, verb, and object constituents almost always are spoken, whereas those describing locative information invariably are written or gestured. However, not much work demonstrated an effective method of using video/image annotation to improve text extraction. Our experiments provide some case studies in this new direction. Our work can also be considered as an extension of global background inference (e.g. Ji and Grishman, 2008) to cross-media paradigm.

Extensive research has been done on video clustering. For example, Cheung and Zakhor (2000) used meta-data extracted from textual and hyperlink information to detect similar videos on the web; Magalhaes et al. (2008) described a semantic similarity metric based on key word vectors for multi-media fusion. We extend such video similarity computing approaches to a multi-lingual environment.

7 Conclusion and Future Work

Traditional Information Extraction (IE) approaches focused on single media (e.g. texts), with very limited use of knowledge from other data modalities in the background. In this paper we propose a new approach to integrate information extracted from videos and texts into a coherent common representation including multi-level knowledge (concepts, relations and events). Beyond standard information fusion, we attempted global inference methods to incorporate video extraction and significantly enhanced the performance of text extraction. Finally, we extend our methods to multi-lingual environment (English, Arabic and Chinese) by presenting a case study on cross-lingual comparable corpora acquisition.

We used a dataset which includes videos and associated speech recognition output (texts), but our approach is applicable to any cases in which texts and videos appear together (from associated texts, captions etc.). The proposed common representation will provide a framework for many byproducts. For example, the monolingual fused information graphs can be used to generate abstractive summaries. Given the fused information we can also visualize the facts from background texts effectively. We are also interested in using video information to discover novel relations and events which are missed in the text IE task.

Acknowledgement

This work was supported by the U.S. Army Research Laboratory under Cooperative Agreement Number W911NF-09-2-0053, the U.S. NSF CAREER Award under Grant IIS-0953149, Google, Inc., DARPA GALE Program, CUNY Research Enhancement Program, PSC-CUNY Research Program, Faculty Publication Program and GRTI Program. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

References

Amato, F., Mazzeo, A., Moscato, V. and Picariello, A. 2010. Information Extraction from Multimedia Documents for e-Government Applications. *Information Systems: People, Organizations, Institutions, and Technologies*. pp. 101-108.

Appriou A., A. Ayoun, Benferhat, S., Besnard, P., Cholvy, L., Cooke, R., Cuppens, F., Dubois, D., Fargier, H., Grabisch, M., Kruse, R., Lang, J. Moral, S., Prade, H., Saffiotti, A., Smets, P., Sossai, C. 2001. Fusion: General concepts and characteristics. *International Journal of Intelligent Systems 16(10).*

Baluja, S. and Rowley, H. 2006. Boosting Sex Identification Performance. *International Journal of Computer Vision.*

Bergsma, S. 2005. Automatic Acquisition of Gender Information for Anaphora Resolution. *Proc. Canadian AI 2005.*

Cheung, P. and Fung P. 2004. Sentence Alignment in Parallel, Comparable, and Quasi-comparable Corpora. *Proc. LREC 2004.*

Cheung, S.-C. and Zakhor, A. 2000. Efficient video similarity measurement and search. *Proc. IEEE International Conference on Image Processing.*

Deschacht K. and Moens M. 2007. Text Analysis for Automatic Image Annotation. *Proc. ACL 2007.*

Feng, Y. and Lapata, M. 2008. Automatic Image Annotation Using Auxiliary Text Information. *Proc. ACL 2008.*

Gregoire, E. 2006. An unbiased approach to iterated fusion by weakening. *Information Fusion. 7(1).*

Gu, Z., Mei, T., Hua, X., Tang, J., Wu, X. 2007. Multi-Layer Multi-Instance Kernel for Video Concept Detection. Proc. *ACM Multimedia 2007.*

Hakkani-Tur, D., Ji, H. and Grishman, R. 2007. Using Information Extraction to Improve Cross-lingual Document Retrieval. *Proc. RANLP 2007 Workshop on Multi-Source Multi-lingual Information Extraction and Summarization.*

Iria, J. and Magalhaes, J. 2009. Exploiting Cross-Media Correlations in the Categorization of Multimedia Web Documents. *Proc. CIAM 2009.*

Ji, H. and Grishman, R. 2008. Refining Event Extraction Through Cross-document Inference. *Proc. ACL 2008.*

Ji, H. 2009. Mining Name Translations from Comparable Corpora by Creating Bilingual Information Networks. *Proc. ACL-IJCNLP 2009 workshop on Building and Using Comparable Corpora (BUCC 2009): from parallel to non-parallel corpora.*

Ji, H., Grishman, R., Freitag, D., Blume, M., Wang, J., Khadivi, S., Zens, R., and Ney, H. 2009. Name Translation for Distillation. *Handbook of Natural Language Processing and Machine Translation: DARPA Global Autonomous Language Exploitation.* Springer.

Ji, H. and Lin, D. 2009. Gender and Animacy Knowledge Discovery from Web-Scale N-Grams for Unsupervised Person Mention Detection. *Proc. PACLIC 2009.*

Oviatt, S. L., DeAngeli, A., & Kuhn, K. 1997. Integration and synchronization of input modes during multimodal human-computer interaction. *Proceedings of Conference on Human Factors in Computing Systems (CHI'97)*, 415-422. New York: ACM Press.

Labsky, M., Praks, P., Sv´atek1, V., and Svab, O. 2005. Multimedia Information Extraction from HTML Product Catalogues. *Proc. 2005 IEEE/WIC/ACM International Conference on Web Intelligence.* pp. 401 – 404.

Lin, D., Church, K., Ji, H., Sekine, S., Yarowsky, D., Bergsma, S., Patil, K., Pitler, E., Lathbury, R., Rao, V., Dalwani, K. and Narsale, S. 2010. New Data, Tags and Tools for Web-Scale N-grams. *Proc. LREC 2010.*

Magalhaes, J., Ciravegna, F. and Ruger, S. 2008. Exploring Multimedia in a Keyword Space. *Proc. ACM Multimedia 2008.*

Munteanu, D. S. and Marcu D. 2005. Improving Machine Translation Performance by Exploiting Non-Parallel Corpora. *Computational Linguistics*. Volume 31, Issue 4. pp. 477-504.

Naphade, M. R., Kennedy, L., Kender, J. R., Chang, S.-F., Smith, J. R., Over, P., and Hauptmann, A. A light scale concept ontology for multimedia understanding for TRECVID 2005. *Technical report, IBM, 2005.*

Pazouki, E. and Rahmati, M. 2009. A novel multimedia data mining framework for information extraction of a soccer video stream. *Intelligent Data Analysis.* pp. 833-857.

Qi,G.-J., Hua,X.-S., Rui, Y., Tang, J., Mei, T., and Zhang,H.-J. 2007. Correlative Multi-label Video Annotation. *Proc. ACM Multimedia 2007*.

Saggion, H., Cunningham, H., Bontcheva, K., Maynard, D., Hamza, O., and Wilks, Y. 2004. Multimedia indexing through multi-source and multi-language information extraction: the MUMIS project. *Data Knowlege Engineering*, 48, 2, pp. 247-264.

Wang, F. and Zhang, C. 2006. Label propagation through linear neighborhoods. *Proc. ICML 2006.*

EM-based Hybrid Model for Bilingual Terminology Extraction from Comparable Corpora

Lianhau Lee, Aiti Aw, Min Zhang, Haizhou Li
Institute for Inforcomm Research
{lhlee, aaiti, mzhang, hli}@i2r.a-star.edu.sg

Abstract

In this paper, we present an unsupervised hybrid model which combines statistical, lexical, linguistic, contextual, and temporal features in a generic EM-based framework to harvest bilingual terminology from comparable corpora through comparable document alignment constraint. The model is configurable for any language and is extensible for additional features. In overall, it produces considerable improvement in performance over the baseline method. On top of that, our model has shown promising capability to discover new bilingual terminology with limited usage of dictionaries.

1 Introduction

Bilingual terminology extraction or term alignment has been well studied in parallel corpora. Due to the coherent nature of parallel corpora, various statistical methods, like EM algorithm (Brown et. al., 1993) have been proven to be effective and have achieved excellent performance in term of precision and recall. The limitation of parallel corpora in all domains and languages has led some researchers to explore ways to automate the parallel sentence extraction process from non-parallel corpora (Munteanu and Marcu, 2005; Fung and Cheung, 2004) before proceeding to the usual term alignment extraction using the existing techniques for parallel corpora. Nevertheless, the coverage is limited since parallel sentences in non-parallel corpora are minimal.

Meanwhile, some researchers have started to exploit comparable corpora directly in a new manner. The motivations for such an approach are obvious: comparable corpora are abundantly available, from encyclopedia to daily newspapers, and the human effort is reduced in either generating or collecting these corpora. If bilingual terminology can be extracted directly from these corpora, evolving or emerging terminologies can be captured much faster than lexicography and this would facilitate many tasks and applications in accessing cross-lingual information.

There remain challenges in term alignment for comparable corpora. The structures of texts, paragraphs and sentences can be very different. The similarity of content in two documents varies through they talk about the same subject matter. Recent research in using transliteration (Udupa et. al., 2008; Knight and Graehl, 1998), context information (Morin et. al., 2007; Cao and Li, 2002; Fung, 1998), part-of-speech tagging, frequency distribution (Tao and Zhai, 2005) or some hybrid methods (Klementiev and Roth, 2006; Sadat et. al., 2003) have shone some light in dealing with comparable corpora. In particular, context information seems to be popular since it is ubiquitous and can be retrieved from corpora easily.

In this paper, we propose an EM-based hybrid model for term alignment to address the issue. Through this model, we hope to discover new bilingual terminology from comparable corpora without supervision. In the following sections, the model will be explained in details.

Coling 2008: Poster Volume, pages 639–646,
Beijing, August 2010

2 System Architecture

It is expensive and challenging to extract bilingual terminologies from a given set of comparable corpora if they are noisy with very diverse topics. Thus the first thing we do is to derive the document association relationship between two corpora of different languages. To do this, we adopt the document alignment approach proposed by Vu et. al. (2009) to harvest comparable news document pairs. Their approach is relying on 3 feature scores, namely Title-n-Content (TNC), Linguistic Independent Unit (LIU), and Monolingual Term Distribution (MTD). In the nutshell, they exploit common words, numbers and identical strings in titles and contents as well as their distribution in time domain. Their method is shown to be superior to Tao and Zai (2005) which simply make use of frequency correlation of words.

After we have retrieved comparable document pairs, we tokenize these documents with prominent monolingual noun terms found within. We are interested only in noun terms since they are more informative and more importantly they are more likely not to be covered by dictionary and we hope to find their translations through comparable bilingual corpora. We adopt the approach developed by Vu et. al. (2008). They first use the state-of-the-art C/NC-Value method (Frantzi and Ananiadou, 1998) to extract terms based on the global context of the corpus, follow by refining the local terms for each document with a term re-extraction process (TREM) using Viterbi algorithm.

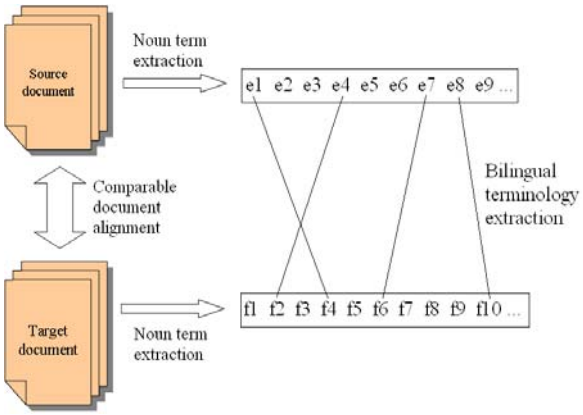


Figure 1. The procedure of bilingual terminology extraction from comparable documents.

After these preprocesses, we have a set of comparable bilingual document pairs and a set of prominent monolingual noun terms for each monolingual document. The aim of our term alignment model is to discover new bilingual terminology formed from these monolingual terms across aligned document pairs (Figure.1).

Like other approaches to comparable corpora, there exist many challenges in aligning bilingual terms due to the presence of noises and the significant text-structure disparity across the comparable bilingual documents. To overcome this, we propose using both corpus-driven and non-corpus-driven information, from which we draw various features and derive our hybrid model. These features are used to make initial guess on the alignment score of term pair candidates. Figure 2 shows the overall process of our term alignment model on comparable corpora. This model is language independent and it comprises several main components:

- EM algorithm
- Term alignment initialization
- Mutual information (MI) & TScore rescoring

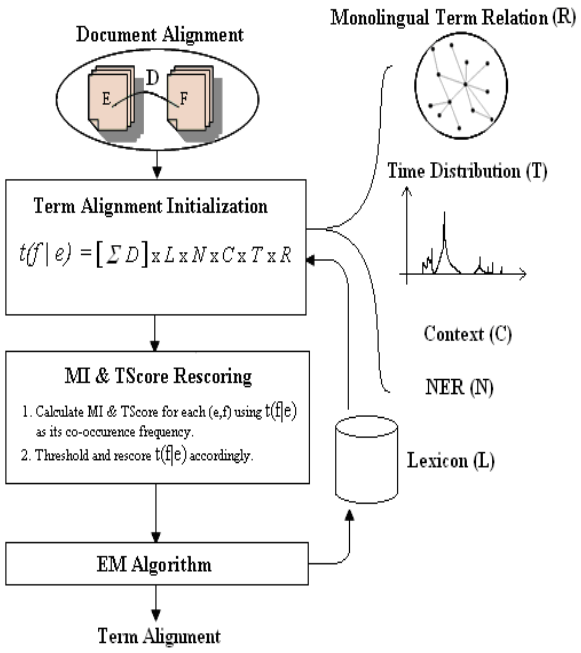


Figure 2. Term alignment model. D = document alignment score, L = lexical similarity, N = named entity similarity, C = context similarity, T = temporal similarity, R = related term similarity.

3 EM Algorithm

We make two assumptions on the preprocesses that the extracted monolingual terms are good representatives of their source documents, and the document alignment scores derived from document alignment process are good indicators of how well the contents of various documents align. Hence, the logical implication suggests that the extracted terms from both well aligned documents could well be candidates of aligned term pairs.

By reformulating the state-of-the-art EM-based word alignment framework IBM model 1 (Brown et. al., 1993), we can derive a term alignment model easily. In IBM word alignment model 1, the task is to find word alignment by using parallel sentences. In the reformulated model for term alignment, parallel sentences are replaced by comparable documents, characterized by document alignment score and their representative monolingual terms.

The significant advantage over the original IBM model 1 is the relaxation of parallel sentences or parallel corpora, by incorporating an additional feature of document alignment score. We initialize the term alignment score of the corresponding term pair candidates with the document alignment score to reflect the confidence level of document alignment. Other than that, we also employ a collection of feature similarity score: lexical similarity, named entity similarity, context similarity, temporal similarity, and related term similarity, to term alignment initialization. We will explain this further in the next section.

As we know, IBM model 1 will converge to the global maximum regardless of the initial assignment. This is truly good news for parallel corpora, but not for comparable corpora which contains a lot of noises. To prevent IBM model 1 from overfitting, we choose to run ten iterations (each iteration consists of one E-step and one M-step) for each cycle of EM in both e-f and f-e directions.

After each cycle of EM process, we simply filter off the weak term alignment pairs of both directions with a high threshold (0.8) and populate the lexicon database with the remaining pairs and use it to start another cycle of EM. The process repeats until no new term alignment pair is found. The EM algorithm for term alignment is shown as follow:

Initialize t(f|e).
for (iteration = 1 to 10)
E step

$$a[i, j, k] = \frac{t(f[k, j] \mid e[k, i])}{\sum_{i} t(f[k, j] \mid e[k, i])}, \; for \quad all \quad i, j, k$$

M step

$$tcount(e, f) = \sum_{\substack{i, j, k: \\ e[k,i]=e, \\ f[k,j]=f}} a[i, j, k], \; for \quad all \quad (e, f)$$

$$t(f \mid e) = \frac{tcount(e, f)}{\sum_{f} tcount(e, f)}, \; for \quad all \quad (e, f)$$

End for.

Figure 3. EM algorithm for e-f direction, where e[k] = k-th aligned source document, f[k] = k-th aligned target document, e[k,i] = i-th term in e[k], f[k,j] = j-th term in f[k], a[i,j,k] = probability of alignment from f[k,j] to e[k,i], t(f|e) = probability of alignment from term e to term f.

4 Term Alignment Initialization

We retrieve term alignment candidates by pairing all possible combinations of extracted monolingual source terms and target terms across the aligned document pairs. Before each cycle of EM, we assign an initial term alignment score, t(f|e) to each of these term pair candidates. Basically, we initialize the term alignment score t(f|e) based on document alignment score (D), lexical similarity (L), named entity similarity (N), context similarity (C), temporal similarity (T), and related term similarity (R). The similarity calculations of the corpus-driven features (D, C, T, R) are derived directly from the corpus and require limited lexical resource. The non-corpus-driven features (L, N) make use of a small word based bilingual dictionary to measure their lexical relevancy. That makes our model not resource-demanding and it shows that our model can work under limited resource condition.

All the above features contribute to the term alignment score t(f|e) independently, and we formulate their cumulative contributions as the following:

$$t(f|e) = \left[\sum_{(E,F): e \in E, f \in F} D(F|E) \right] \times L(f|e) \times N(f|e) \times C(f|e) \times T(f|e) \times R(f|e) \quad (1)$$

where,
e = source term
f = target term
E = source document
F = target document
D = document alignment score
L = lexical similarity
N = named entity similarity
C = context similarity
T = temporal similarity
R = related term similarity

This formula allows us to extend the model with additional features without affecting the existing configuration.

4.1 Document Alignment Score (D)

As explained in the Section 3, the relaxation on the requirement of parallel corpora in the new EM model leads to the incorporation of document alignment score. To indicate the confidence level of document alignment, we credit every aligned term pair candidate formed across the aligned documents with the corresponding document alignment score. Although it is not necessary, document alignment score is first normalized to the range of [0,1], with 1 indicates parallel alignment.

4.2 Lexical Similarity (L)

We design a simple lexical similarity measurement of two terms based on word translation. Term pairs that share more than 50% of word translation pairs will be credited with lexical similarity of L_0, where L_0 is configurable contribution weightage of lexical similarity. This provides us a primitive hint on term alignment without resorting to exhaustive dictionary lookup.

$$L(f|e) = \begin{cases} L_0, & if \quad T_W(f|e) \geq 0.5 \\ 1, & otherwise \end{cases} \quad (2)$$

where $L_0 > 1$ and $T_W(f|e)$ is word translation score.

4.3 Named Entity Similarity (N)

Named entity similarity is a measure of predefined category membership likelihood, such as person, location and organization. Term pairs that belong to the same NE categories will be credited with named entity similarity of N_0, where N_0 is a configurable weightage of named entity similarity. We use this similarity score to discover bilingual terms of same NE categories, yet not covered by bilingual dictionary.

$$N(f|e) = \begin{cases} N_0, if \quad NE \quad categories \quad match \\ 1, otherwise \end{cases} \quad (3)$$

where $N_0 > 1$.

4.4 Context Similarity (C)

We assume that terms with similar contexts are likely to have similar meaning. Thus, we make use of context similarity to measure semantic similarity. Here, only k nearest content words (verbs, nouns, adjectives and adverbs) before or after the terms within the sentence boundary are considered as its contexts. The following shows the calculation of context similarity of two terms based on cosine similarity between their context frequency vectors before scaling to the range of [1, C_0], where C_0 is a configurable contribution weightage of context similarity. As shown in the formula, the t(f'|e') accounts for the translation probability from the source context word to the target context word, hence the cosine similarity calculation is carried out in the target language domain.

$$C(f|e) = 1 + (C_0 - 1) \frac{\sum_{\substack{e' \in context(e) \\ f' \in context(f)}} freq(e') freq(f') t(f'|e')}{\sqrt{\sum_{e' \in context(e)} freq(e')^2} \sqrt{\sum_{f' \in context(f)} freq(f')^2}} \quad (4)$$

where $C_0 > 1$.

4.5 Temporal Similarity (T)

In temporal similarity, we make use of date information which is available in some corpus (e.g. news). We assume aligned terms are synchronous in time, this is especially true for comparable news corpora (Tao and Zai, 2005). We

use Discrete Fourier Transform (DFT) to transform the distribution function of a term in discrete time domain to a representative function in discrete frequency domain, which is usually known as "spectrum". We then calculate the power spectrum, which is defined as magnitude square of a spectrum. Power spectrum is sensitive to the relative spacing in time (or frequency component), yet invariant to the shifting in time, thus it is most suitably to be used for pattern matching of time distribution. The temporal similarity is calculated based on cosine similarity between the power spectrums of the two terms before scaling to the range of [1, T_0], where T_0 is a configurable contribution weightage of temporal similarity.

$$T(f \mid e) = (T_0 - 1)\cos sine\left(P_e(k), P_f(k)\right) + 1 \quad (5)$$

where $T_0 > 1$ and

$$\cos sine\left(u(k), v(k)\right) = \frac{\sum_k u(k)v(k)}{\sqrt{\sum_k u(k)^2}\sqrt{\sum_k v(k)^2}}$$

$$P_x(k) = \left| DFT\{DistributionFunction_x(n)\}\right|^2$$

$$= \left| \sum_{n=0}^{N-1} DistributionFunction_x(n) \times e^{-\frac{2\pi i}{N}kn} \right|^2$$

4.6 Related Term Similarity (R)

Related terms are terms that correlate statistically in the same documents and they can be found by using mutual information or t-test in the monolingual corpus. Basically, related term similarity is a measure of related term likelihood. Aligned terms are assumed to have similar related terms, hence related term similarity contributes to semantic similarity. The related term similarity is calculated based on weighted contribution from the related terms of the source term before scaling to the range of [1, R_0], where R_0 is a configurable contribution weightage of related terms similarity.

$$R(f \mid e) = (R_0 - 1)Rsimilarity(f \mid e) + 1 \quad (6)$$

where $R_0 > 1$ and

$$Rsimilarity(f \mid e) = \frac{\sum_{e' \in R(e)} vote(f \mid e')}{\sum_{f \in F}\sum_{e' \in R(e)} vote(f \mid e')}$$

$$vote(f \mid e') = \frac{\sum_{\substack{e'' \in R(e) \cap \\ [R(e') \cup \{e'\}]}} w(e'', f) \times MI(e, e'') \times vote(f \mid e'')}{\sum_{f \in F}\sum_{\substack{e'' \in R(e) \cap \\ [R(e') \cup \{e'\}]}} w(e'', f) \times MI(e, e'') \times vote(f \mid e'')}$$

$$w(e'', f) = \begin{cases} 1.5, & if \quad Tr(e'') \cap R(f) \neq \varnothing \\ 1, & otherwise \end{cases}$$

$$MI(e, e'') = \log\left(\frac{p(e, e'')}{p(e)p(e'')}\right)$$

vote(f|e') is initialized to 1 before it is computed iteratively until it converges. R(e) is the set of related term of e and Tr(e) is the set of translated term of e.

5 MI & TScore Rescoring

We design the MI & TScore rescoring process to enhance the alignment score t(f|e) of e-f term pairs that have significant co-occurrence frequencies in aligned document pairs, based on pointwise mutual information and TScore (or commonly known as t-test) of the terms. By using both measures concurrently, the association relationship of a term pair can be assumed with higher confidence. On top of that, the association of a term pair can also be suggested by a much higher TScore value alone. In this rescoring process, we scale up the alignment score t(f|e) of any term pair which is strongly associated by a constant factor. The following shows the mathematical expressions of what has been described, with M_0 as the configurable scaling factor.

Rescoring condition:

$$if \quad \{[TScore(e, f) \geq 2.5 \quad and \quad MI(e, f) \geq 0.6 \times \underset{\substack{(e', f'): freq(e') = freq(e) \\ or \quad freq(f') = freq(f)}}{Max} MI(e', f')] \quad (7)$$

$$or \quad TScore(e, f) \geq 5\} \quad then$$

$$T(f \mid e) = T(f \mid e) \times M_0$$

where $M_0 > 1$ and

$$TScore\ (e, f) = \frac{p(e, f) - p(e)p(f)}{\sqrt{\frac{p(e, f)}{2N}}}$$

$$N = NumberOfPair(e, f)$$

6 Experiment and Evaluation

We conduct the experiment on articles from three newspapers of different languages published by Singapore Press Holding (SPH), namely Straits Times[1] (English), ZaoBao[2] (Chinese) and Berita Harian[3] (Malay), in June 2006. There are 3187 English articles, 4316 Chinese articles and 1115 Malay articles. English is chosen to be the source language and the remaining two languages as target languages. To analyze the effect of the quality of comparable document in our term alignment model, we prepare two different input sets of document alignment, namely golden document alignment and automated document alignment for each source-target language pair. The former is retrieved by linguistic experts who are requested to read the contents of the articles in the source and the target languages, and then match the articles with similar contents (e.g. news coverage on same story), while the latter is generated using unsupervised method proposed by Vu et. al. (2009), mentioned in Section 2.

In both cases of document alignments, only monolingual noun terms extracted automatically by program (Vu et. al., 2008) will be used as basic semantic unit. There are 23,107 unique English noun terms, 31,944 unique Chinese noun terms and 8,938 unique Malay noun terms extracted in overall. In average, there are 17.3 noun term tokens extracted for each English document, 16.9 for Chinese document and 13.0 for Malay document. Also note that the term alignment reference list is constructed based on these extracted monolingual terms under the constraints of document alignment. In other words, the linguistic experts are requested to match the extracted terms across aligned document pairs (for both golden document alignment and automated document alignment sets respectively). The numbers of comparable document pairs and the corresponding unique term alignment reference pairs are shown in Table 2.

[1] http://www.straitstimes.com/ an English news agency in Singapore. Source © Singapore Press Holdings Ltd.

[2] http://www.zaobao.com/ a Chinese news agency in Singapore. Source © Singapore Press Holdings Ltd.

[3] http://cyberita.asia1.com.sg/ a Malay news agency in Singapore. Source © Singapore Press Holdings Ltd.

In the experiment, we will conduct the named entity recognition (NER) by using the developed system from the Stanford NLP Group, for English, and an in-house engine, for Chinese. Currently, there is no available NER engine for Malay.

Dictionary	E-C	C-E	E-M	M-E
Entry	23,979	71,287	28,496	18,935

Table 1. Statistics of dictionaries, where E = English, C = Chinese, M = Malay.

Corpus	GoldenDocAlign		AutomatedDocAlign	
	Doc Align	Term Align Ref	Doc Align	Term Align Ref
ST-ZB	90	313	899	777
ST-BH	42	113	475	358

Table 2. Statistics of comparable document alignment pairs and term alignment reference pairs.

For baseline, we make use of IBM model 1, modified in the same way which has been described in the section 3, except that we treat all comparable documents as parallel sentences, i.e. document alignment score is 1. Precision and recall are used to evaluate the performance of the system. To achieve high precision, high thresholds are used in the system and they are kept constant throughout the experiments for consistency. To evaluate the capability of discovering new bilingual terminology, we design a novelty metric, which is the ratio of the number of correct out-of-dictionary term alignment over the total number of correct term alignment.

$$Precision = \frac{C}{T} \qquad Recall = \frac{C}{G} \qquad Novelty = \frac{N}{C} \tag{8}$$

where,

C = total number of correct term alignment result.
T = total number of term alignment result.
G = total number of term alignment reference.
N = total number of correct term alignment result that are out-of-dictionary.

Table 3 shows the evaluation result of term alignment using EM algorithm with incremental feature setting. The particular order of setting is due to the implementation sequences and it is not expected to affect the result of analysis.

We observe that the precision, recall and novelty of the system are comparatively higher when the golden document alignment is used instead of the automated document alignment.

corpora	Setting	GoldenDocAlign			AutomatedDocAlign		
		Precision	Recall	Novelty	Precision	Recall	Novelty
ST-ZB	IBM 1	75.0%	1.92%	50.0%	22.2%	0.26%	50.0%
	(D)	75.0%	1.92%	50.0%	22.2%	0.26%	50.0%
	(D,L)	81.8%	2.88%	55.6%	33.3%	0.52%	25.0%
	(D,L,R)	81.8%	2.88%	55.6%	33.3%	0.52%	25.0%
	(D,L,R,M)	78.6%	3.51%	63.6%	35.7%	0.64%	40.0%
	(D,L,R,M,N)	88.2%	4.79%	53.3%	35.7%	0.64%	40.0%
	(D,L,R,M,N,C)	89.5%	5.43%	52.9%	33.3%	0.64%	40.0%
	(D,L,R,M,N,C,T)	89.5% (17/19)	5.43% (17/313)	52.9% (9/17)	37.5% (6/16)	0.77% (6/777)	16.7% (1/6)
ST-BH	IBM 1	33.3%	0.89%	0.00%	33.3%	0.78%	0.00%
	(D)	33.3%	0.89%	0.00%	33.3%	0.78%	0.00%
	(D,L)	75.0%	5.31%	50.0%	50.0%	1.94%	0.00%
	(D,L,R)	75.0%	5.31%	50.0%	50.0%	1.94%	0.00%
	(D,L,R,M)	75.0%	5.31%	50.0%	54.5%	2.33%	0.00%
	(D,L,R,M,N)	75.0%	5.31%	50.0%	54.5%	2.33%	0.00%
	(D,L,R,M,N,C)	83.3%	8.85%	60.0%	50.0%	1.94%	0.00%
	(D,L,R,M,N,C,T)	83.3% (10/12)	8.85% (10/113)	60.0% (6/10)	50.0% (5/10)	1.94% (5/258)	0.00% (0/5)

Table 3. Performance of term alignment using EM algorithm with incremental feature setting, where D = document alignment, L = lexical similarity, R = related term similarity, M = MI & TScore rescoring, N = named entity similarity, C = context similarity, T = temporal similarity.

This is expected since the golden document alignment provides document pairs with stronger semantic bonding. This also suggests that improving on the document alignment would further improve the term alignment result.

It is noteworthy observation that the implemented features improve the system precision and recall under various scenarios, although the degree of improvement varies from case to case. This shows the effectiveness of these features in the model.

On the other hand, the novelty of the system is around 40%+ and 50%+ for ST-ZB and ST-BH respectively (except for the automated document alignment in ST-BH scenarios). This suggests that the system can discover quite a large percentage of the correct bilingual terminologies that do not exist in the lexicon initially.

Compared with the baseline IBM model 1, there is an increase of 14.5% in precision, 3.51% in recall and 2.9% in novelty for ST-ZB, using the golden document alignment. For ST-BH, there is an even larger increase: 50% in precision, 7.96% in recall and 60% in novelty.

7 Conclusion

We have proposed an unsupervised EM-based hybrid model to extract bilingual terminology from comparable corpora through document alignment constraint. Our strategy is to make use of various information (corpus-driven and non-corpus-driven) to make initial guess on the semantic bonding of the term alignment candidates before subjecting them to document alignment constraint through EM algorithm. The hybrid model allows inclusion of additional features without reconfigurations on existing features, this make it practically attractive. Moreover, the proposed system can be easily deployed in any language with minimal configurations.

We have successfully conducted the experiments in English-Chinese and English-Malay comparable news corpora. The features employed in the model have shown incremental improvement in performance over the baseline method. In particular, the system shows improvement in the capability to discover new bilingual terminology from comparable corpora even with limited usage of dictionaries.

From the experiments, we have found that the quality of comparable bilingual documents is a

major limiting factor to achieve good performance. In future, we want to explore ways to improve on this.

References

R. Agrawal, C. Faloutsos, and A. Swami. 1993. *Efficient similarity search in sequence databases*. In *Proceedings of the 4th International Conference on Foundations of Data Organization and Algorithms*. Chicago, United States.

P. F. Brown, V. S. A. Della Pietra, V. J. Della Pietra, and R. L. Mercer. 1993. *The mathematics of statistical machine translation: Parameter estimation*. Computational Linguistics, 19(2): 263-312.

Yunbo Cao and Hang Li. 2002. *Base Noun Phrase Translation Using Wed Data and the EM Algorithm*, Computational Linguistics, pp.1-7.

Pascale Fung, 1998. *A statistical view on bilingual lexicon extraction: From parallel corpora to non-parallel corpora.* Proceedings of AMTA, pp.1-17.

Pascale Fung and Percy Cheung. 2004. *Mining Very-Non-Parallel Corpora: Parallel Sentence and Lexicon Extraction via Bootstrapping and EM*, Proceedings of EMNLP, pp.57-63.

Alexandre Klementiev and Dan Roth, 2006. *Weakly Supervised Named Entity Transliteration and Discovery from Multilingual Comparable Corpora*. Computational Linguistics, pp. 817-824.

K. Knight and J. Graehl. 1998. *Machine transliteration*, Computational Linguistics, 24(4): 599-612.

E. Morin, B. Daille, K. Takeuchi, K. Kageura. 2007. *Bilingual Terminology Mining – Using Brain, not brawn comparable corpora*, Proceedings of ACL.

Dragos Stefan Munteanu and Daniel Marcu. 2005. *Improving Machine Translation Performance by Exploiting Non-Parallel Corpora.* Computational Linguistics, 31(4): 477-504.

Fatiha Sadat, Masatoshi Yoshikawa, Shunsuke Uemura, 2003. *Learning Bilingual Translations from Comparable Corpora to Cross-Language Information Retrieval: Hybrid Statistics-based and Linguistics-based Approach.* Proceedings of ACL, vol.11, pp.57-64.

Tao Tao and Chengxiang Zhai. 2005. *Mining comparable bilingual text corpora for cross-language information integration*, Proceedings of ACM.

Raghavendra Udupa, K. Saravanan, A. Kumaran, Jagadeesh Jagarlamudi. 2008. *Mining named entity transliteration equivalents from comparable corpora*, Proceedings of ACM.

Thuy Vu, Aiti Aw, Min Zhang, 2008. *Term extraction through unithood and termhood unification.* Proceedings of IJCNLP-08, Hyderabad, India.

Thuy Vu, Aiti Aw, Min Zhang, 2009. *Feature-based Method for Document Alignment in Comparable News Corpora*. Proceedings of EACL-09, Athens, Greece.

Text Mining for Automatic Image Tagging

Chee Wee Leong and Rada Mihalcea and Samer Hassan
Department of Computer Science and Engineering
University of North Texas
cheeweeleong@my.unt.edu, rada@cs.unt.edu, samer@unt.edu

Abstract

This paper introduces several extractive approaches for automatic image tagging, relying exclusively on information mined from texts. Through evaluations on two datasets, we show that our methods exceed competitive baselines by a large margin, and compare favorably with the state-of-the-art that uses both textual and image features.

1 Introduction

With continuously increasing amounts of images available on the Web and elsewhere, it is important to find methods to annotate and organize image databases in meaningful ways. Tagging images with words describing their content can contribute to faster and more effective image search and classification. In fact, a large number of applications, including the image search feature of current search engines (e.g., Yahoo!, Google) or the various sites providing picture storage services (e.g., Flickr, Picasa) rely exclusively on the tags associated with an image in order to search for relevant images for a given query.

However, the task of developing accurate and robust automatic image annotation models entails daunting challenges. First, the availability of large and correctly annotated image databases is crucial for the training and testing of new annotation models. Although a number of image databases have emerged to serve as evaluation benchmarks for different applications, including image annotation (Duygulu et al., 2002), content-based image retrieval (Li and Wang, 2008) and cross language information retrieval (Grubinger et al., 2006), such databases are almost exclusively created by manual labeling of keywords, requiring significant human effort and time. The content of these image databases is often restricted only to a few domains, such as medical and natural photo scenes (Grubinger et al., 2006), and specific objects like cars, airplanes, or buildings (Fergus et al., 2003). For obvious practical reasons, it is important to develop models trained and evaluated on more realistic and diverse image collections.

The second challenge concerns the extraction of useful image and text features for the construction of reliable annotation models. Most traditional approaches relied on the extraction of image colors and textures (Li and Wang, 2008), or the identification of similar image regions clustered as blobs (Duygulu et al., 2002) to derive correlations between image features and annotation keywords. In comparison, there are only a few efforts that leverage on the multitude of resources available for natural language processing to derive robust linguistic-based image annotation models. One of the earliest efforts involved the use of captions for face recognition in photographs through the construction of a specific lexicon that integrates linguistic and photographic information (Srihari and Burhans, 1994). More recently, several approaches have proposed the use of WordNet as a knowledge-base to improve content-based image annotation models, either by removing noisy keywords through semantic clustering (Jin et al., 2005) or by inducing a hierarchical classification of candidate labels (Srikanth et al., 2005).

In this paper, we explore the use of several natural language resources to construct image annotation models that are capable of automatically tagging images from unrestricted domains with good accuracy. Unlike traditional image annotation methodologies that generate tags using image-based features, we propose to extract them in a manner analogous to keyword extraction. Given a target image and its surrounding text, we extract those words and phrases that are most likely to represent meaningful tags. More importantly, we

Coling 2008: Poster Volume, pages 647–655,
Beijing, August 2010

are interested to investigate the potential of such linguistic-based models on image annotation accuracy and reliability. Our work is motivated by the need for annotation models that can be efficiently applied on a very large scale (e.g. harvesting images from the web), which are required in applications that cannot afford the complexity and time associated with current image processing techniques.

The paper makes the following contributions. We first propose a new evaluation framework for image tagging, which is based on an analogy drawn between the tasks of image labeling and lexical substitution. Next, we present three extractive approaches for the task of image annotation. The methods proposed are based only on the text surrounding an image, without the use of image features. Finally, by combining several orthogonal methods through machine learning, we show that it is possible to achieve a performance that is competitive to a state-of-the-art image annotation system that relies on visual and textual features, thus demonstrating the effectiveness of text-based extractive annotation models.

2 Related Work

Several online systems have sprung into existence to achieve annotation of real world images through human collaborative efforts (Flickr) and stimulating competition (von Ahn and Dabbish, 2004). Although a large number of image tags can be generated in short time, these approaches depend on the availability of human annotators and are far from being automatic. Similarly, research in the other direction via text-to-image synthesis (Li and Fei-Fei, 2008; Collins et al., 2008; Mihalcea and Leong, 2009) has also helped to harvest images, mostly for concrete words, by refining image search engines.

Most approaches to automatic image annotation have focused on the generation of image labels using annotation models trained with image features and human annotated keywords (Barnard and Forsyth, 2001; Jeon et al., 2003; Makadia et al., 2008; Wang et al., 2009). Instead of predicting specific words, these methods generally target the generation of semantic classes (e.g. vegetation, animal, building, places etc), which they can achieve with a reasonable amount of success. Recent work has also considered the generation of labels for real-world images (Li and Wang, 2008; Feng and Lapata, 2008). To our knowledge, we are unaware of any other work that performs extractive annotation for images from unrestricted domains through the exclusive use of textual features.

3 Dataset

As the methods we propose are extractive, standard image databases with no surrounding text such as Corel (Duygulu et al., 2002) are not suitable, nor are they representative for the challenges associated with raw data from unrestricted domains. We thus create our own dataset using images randomly extracted from the Web.

To avoid sparse searches, we use a list of the most frequent words in the British National Corpus as seed words, and query the web using the Google Image API. A webpage is randomly selected from the query results if it contains a single image in the specified size range (width and height of 275 to 1000 pixels[1]) and its text contains more than 10 words. Next, we use a Document Object Model (DOM) HTML parser[2] to extract the content of the webpage. Note that we do not perform manual filtering of our images except where they contain undesirable qualities (e.g. porn, corrupted or blank images).

In total, we collected 300 image-text pairs from the web. The average image size is 496 pixels width and 461 pixels height. The average text length is 278 tokens and the average document title length is 6 tokens. In total, there are 83,522 words and the total vocabulary is 8,409 words.

For each image, we also create a gold standard of manually assigned tags, by using the labels assigned by five human annotators. The image annotation is conducted via Amazon Mechanical Turk, which was shown in the past to produce reliable annotations (Snow et al., 2008). For increased annotation reliability, we only accept annotators with an approval rating of 98%.

Given an image, an annotator extracts from the associated text a minimum of five words or collocations.Annotators can choose words freely from the text, while collocation candidates are restricted to a fixed set obtained from the n-grams (n ≤ 7) in the text that also appear as article names or surface forms in Wikipedia. Moreover, when interpreting the image, the annotators are instructed to focus on both the denotational and conotational attributes present in the image[3].

[1] Empirically determined to filter advertisements, banners and undersized images.

[2] http://search.cpan.org/dist/HTML-ContentExtractor/

[3] Annotation instructions, dataset and gold standard can

	Normal Image	**Mode** Image
		
Gold standard	czech (5), festival (5), oklahoma (4), yukon (4), october (4), web page (2), the first (2), event (2), success (1), every (1), year (1)	train (5), station (4), steam (4), trans siberian (4), steam train (4), travel (3), park (3), siberian (3), old (3), photo (1), trans (2), yekaterinburg (2), the web (2), photo host (1)

Table 1: Two sample images. The number besides each label indicates the number of human annotators agreeing on that label. Note that the mode image has a tag (i.e."train") in the gold standard set most frequently selected by the annotators

4 A New Evaluation Framework : Image Tagging as Lexical Substitution

While evaluations of previous work in image annotation were often based on labels provided with the images, such as tags or image captions, in our dataset such annotations are either missing or unreliable. We rely instead on human-produced extractive annotations (as described in the previous section), and formulate a new evaluation framework based on the intuition that an image can be substituted with one or more tags that convey the same meaning as the image itself. Ideally, there is a single tag that "best" describes the image overall (i.e. the gold standard tag agreed by the majority of human annotators), but there are also multiple tags that describe the fine-grained concepts present in the image. Our evaluation framework is inspired by the lexical substitution task (McCarthy and Navigli, 2007), where a system attempts to generate a word (or a set of words) to replace a target word, such that the meaning of the sentence is preserved.

Given this analogy, the evaluation metrics used for lexical substitution can be adapted to the evaluation of image tagging. Specifically, we measure the precision and the recall of a tagging method using four subtasks: **best normal**: provides precision and recall for the top-ranked tag returned by a method; **best mode**: provides precision and recall only if the top-ranked tag by a method matches the tag in the gold standard that was most frequently selected by the annotators; **out of ten (oot) normal**: provides precision and recall for the top ten tags by the system; and **out of ten (oot) mode**: similar to best mode, but it considers the top ten tags returned by the system instead of one. Table 1 show examples of a normal and a mode image.

Formally, let us assume that $\mathtt{H}$ is the set of annotators, namely $\{\mathtt{h}_1, \mathtt{h}_2, \mathtt{h}_3, ...\}$, and $\mathtt{I}$, $\{\mathtt{i}_1, \mathtt{i}_2, \mathtt{i}_3, ...\}$ is the set of images for which each human annotator provide at least five tags. For each $\mathtt{i}_j$, we calculate $\mathtt{m}_j$, which is the most frequent tag for that image, if available. We also collect all $\mathtt{r}_j^k$, which is the set of tags for the image $\mathtt{i}_j$ from the annotator $\mathtt{h}_k$.

Let the set of those images where there is a tag agreed upon by the most annotators (i.e. the images with a mode) be denoted by $\mathtt{IM}$, such that $\mathtt{IM} \subseteq \mathtt{I}$. Also, let $\mathtt{A} \subseteq \mathtt{I}$ be the set of images for which the system provides more than one tag. Let the corresponding set for the images with modes be denoted by $\mathtt{AM}$, such that $\mathtt{AM} \subseteq \mathtt{IM}$. Let $\mathtt{a}_j \in \mathtt{A}$ be the set of system's extracted tags for the image $\mathtt{i}_j$.

Thus, for each image $\mathtt{i}_j$, we have the set of tags extracted by the system, and the set of tags from the human annotators. As the next step, the multiset union of the human tags is calculated, and the frequencies of the unique tags is noted. Therefore, for image $\mathtt{i}_j$, we calculate $\mathtt{R}_j$, which is $\sum \mathtt{r}_j^k$, and the individual unique tag in $\mathtt{R}_j$, say $\mathtt{res}$, will have a frequency associated with it, namely $\mathtt{freq}_{\mathtt{res}}$.

Given this setting, the precision (P) and recall (R) metrics we use are defined below.

be downloaded at
http://lit.csci.unt.edu/index.php/Downloads

Best measures:

$$P = \frac{\sum_{a_j:i_j \in A} \frac{\frac{\sum_{res \in a_j} freq_{res}}{|a_j|}}{|R_j|}}{|A|}$$

$$R = \frac{\sum_{a_j:i_j \in I} \frac{\frac{\sum_{res \in a_j} freq_{res}}{|a_j|}}{|R_j|}}{|I|}$$

$$modeP = \frac{\sum_{bestguess_j \in AM} (1if\_best\_guess = m_j)}{|AM|}$$

$$modeR = \frac{\sum_{bestguess_j \in IM} (1if\_best\_guess = m_j)}{|IM|}$$

Out of ten (oot) measures:

$$P = \frac{\sum_{a_j:i_j \in A} \frac{\sum_{res \in a_j} freq_{res}}{|R_j|}}{|A|}$$

$$R = \frac{\sum_{a_j:i_j \in I} \frac{\sum_{res \in a_j} freq_{res}}{|R_j|}}{|I|}$$

$$modeP = \frac{\sum_{a_j:i_j \in AM} (1if\_any\_guess \in a_j = m_j)}{|AM|}$$

$$modeR = \frac{\sum_{a_j:i_j \in IM} (1if\_any\_guess \in a_j = m_j)}{|IM|}$$

As a simplified example (with less tags), consider $\mathtt{i_j}$ showing a picture of a Chihuahua being labeled by five annotators with the following tags :

Annotator	Tags
1	dog,pet
2	chihuahua
3	animal,dog
4	dog,chihuahua
5	dog

In this case, $\mathtt{r}^1_\mathtt{j}$ = {dog,pet}, $\mathtt{r}^2_\mathtt{j}$ = {chihuahua}, $\mathtt{r}^3_\mathtt{j}$ = {animal,dog} and so on. The tag "dog" appears the most frequent among the five annotators, hence $\mathtt{m_j}$ = {dog}. $\mathtt{R_j}$={dog, dog, dog, dog, chihuahua, chihuahua, animal, pet}. The `res` with associated frequencies would be dog 4, chihuahua 2, animal 1, pet 1. If the system's proposed tag for $\mathtt{i_j}$ is {dog, animal}, then the numerator of P and R for best subtask would be $\frac{\frac{4+1}{2}}{8}$ = 0.313. Similarly, the numerator of P and R for oot subtask is $\frac{4+1}{8}$ = 0.625.

5 Extractive Image Annotation

The main idea underlying our work is that we can perform effective image annotation using information drawn from the associated text. Following (Feng and Lapata, 2008), we propose that an image can be annotated with keywords capturing the denotative (entities or objects depicted) and connotative (semantics or ideologies interpreted) attributes in the image. For instance, a picture showing a group of athletes and a ball may also be tagged with words like "soccer," or "sports activity." Specifically, we use a combination of knowledge sources to model the denotative quality of a word as its picturability, and the connotative attribute as its saliency. The idea of visualness and salience as textual features for discovering named entities in an image was first pursued by (Deschacht and Moens, 2007), using data from the news domain. In contrast, we are able to perform annotation of images from unrestricted domains using content words (nouns, verbs and adjectives). In the following, we first describe three unsupervised extractive approaches for image annotation, followed by a supervised method using a re-ranking hypothesis that combines all the methods.

5.1 Flickr Picturability

Featuring a repository of four billion images, Flickr (http://www.flickr.com) is one of the most comprehensive image resources on the web. As a photo management and sharing application, it provides users with the ability to tag, organize, and share their photos online. Interestingly, an inspection of Flickr tags for randomly selected images reveal that users tend to describe the denotational attributes of images, using concrete and picturable words such as cat, bug, car etc. This observation lends evidence to Flickr's suitability as a resource to model the picturability of words.

Given the text (T) of an image, we can use the $getRelatedTags$ API to retrieve the most frequent Flickr tags associated with a given word, and use them as corpus evidence to filter or promote words in the text. In the filtering phase we ignore any words that return an empty list of Flickr's related tags, based on the assumption that these words are not used in the Flickr tags repository. We also discard words with a length that is less than three characters (α=3). In the promotion phase, we reward any retrieved tags that appear as surface forms in the text. This reward is proportional to the term frequency of these tags in the

Algorithm 1 Flickr Picturability Algorithm

```
Start : L[]=φ , TF[]=tf of each word in T
for each word in T do
  if length(word) ≥ α then
    RelatedTags=getRelatedTags(word);
    if size(RelatedTags) > 0 then
      L[word]+=β*TF[word]
      for each tag in RelatedTags do
        if exists TF[tag] then
          L[tag]+=TF[tag]
        end if
      end for
    end if
  end if
end for
```

text. Additionally, we also include in the final label set any word that returns a non-empty related tags set with a discounted weight (β=0.5) of its term frequency, to the end of enriching our labels set while assuring more credit are given to the picturable words.

To extract multiword labels, we locate all n-grams formed exclusively from our extracted set of possible labels. The subsequent score for each of these n-grams is:

$$L[w_i..w_{i+k}] = (\sum_{j=i}^{j=i+k} L[wj])/k$$

By reverse sorting the associative array in L, we can retrieve the top K words to label the image. For illustration, let us consider the following text snippet.

> *On the Origin of Species, published by Charles Darwin in 1859, is considered to be the foundation of evolutionary biology.*

After removing stopwords, we consider the remaining words as candidate labels. For each of these candidates w_i (i.e. $origin$, $species$, $published$, $charles$, $darwin$, $foundation$, $evolutionary$, and $biology$), we query Flickr and obtain their related tag set R_i. $origin$, $published$, and $foundation$ return an empty set of related tags and hence are removed from our set of candidate labels, leaving $species$, $charles$, $darwin$, $evolutionary$, and $biology$ as possible annotation keywords with the initial score of 0.5. In the promotion phase, we score each w_i based on the number of votes it receives from the remaining w_j (Figure 1). Each vote represents an occurrence of the candidate tag w_i in the related tag set R_j of the candidate tag w_j. For example, $darwin$ appeared in the Flickr related tags for $charles$, $evolutionary$, and $biology$, hence it has a weight of 3.5. The final list of candidate labels are shown in Table 2.

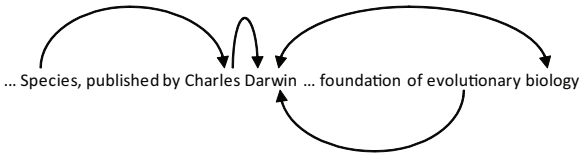


Figure 1: Flickr Picturability Labels

Label	$S(w_i)$
darwin	3.5
charles darwin	2.5
charles	1.5
biology	1.5
evolutionary biology	1.0
evolutionary	0.5
species	0.5

Table 2: Candidate labels obtained for a sample text using the Flickr model

5.2 Wikipedia Salience

We hypothesize that an image often describes the most important concepts in the associated text. Thus, the keywords selected from a text could be used as candidate labels for the image. We use a graph-based keyword extraction method similar to (Mihalcea and Tarau, 2004), enhanced with a semantic similarity measure. Starting with a text, we extract all the candidate labels and add them as vertices in the graph. A measure of word similarity is then used to draw weighted edges between the nodes. Using the PageRank algorithm, the words are assigned with a score indicating their salience within the given text.

To determine the similarity between words, we use a directed measure of similarity. Most word similarity metrics provide a single-valued score between a pair of words w_1 and w_2 to indicate their semantic similarity. Intuitively, this is not always the case, as w_1 may be represented by concepts that are entirely embedded in other concepts, represented by w_2. In psycholinguistics terms, uttering w_1 may bring to mind w_2, while the appearance of w_2 without any contextual clues may not associate with w_1. For example, $Obama$ brings to mind the concept of $president$, but $president$

may trigger other concepts such as $Washington$, $Lincoln$, $Ford$ etc., depending on the existing contextual clues. Thus, the degree of similarity of w_1 with respect to w_2 should be separated from that of w_2 with respect to w_1. Specifically, we use the following measure of similarity, based on the Explicit Semantic Analysis (ESA) vectors derived from Wikipedia (Gabrilovich and Markovitch, 2007):

$$DSim(w_i, w_j) = \frac{C_{ij}}{C_i} * Sim(w_i, w_j)$$

where C_{ij} is the count of articles in Wikipedia containing words w_i and w_j, C_i is the count of articles containing words w_i, and Sim(w_i, w_j) is the cosine similarity of the ESA vectors representing the input words.The *directional weight* (C_{ij}/C_i) amounts to the degree of association of w_i with respect to w_j. Using the directional inferential similarity scores as directed edges and distinct words as vertices, we obtain a graph for each text. The directed edges denotes the idea of "recommendation" where we say w_1 recommends w_2 if and only if there is a directed edge from w_1 to w_2, with the weight of the recommendation being the directional similarity score. Starting with this graph, we use the graph iteration algorithm from (Mihalcea and Tarau, 2004) to calculate a score for each vertex in the graph. The output is a sorted list of words in decreasing order of their ranks, which are used as candidate labels to annotate the image. This is achieved by using C_j instead of C_i for the denominator in the directional weight. As an example, consider the text snippet :

> *Microsoft Corporation is a multinational computer technology corporation that develops, manufactures, licenses, and supports a wide range of software products for computing devices*

after stopword removal, the list of nouns extracted is *Microsoft, computer, corporation, devices, products, technology, software*. Note that the top-ranked word must infer some or all of the words in the text. In this case, the word *Microsoft* infers the terms *computer*, *technology* and *software*.

To calculate the semantic relatedness between two collocations, we use a simplified version of the text-to-text relatedness technique proposed by and (Mihalcea et al., 2006) that incorporate the directional inferential similarity as an underlying semantic metric.

5.3 Topical Modeling

Intuitively, every text is written with a topic in mind, and the associated image serves as an illustration of the text meaning. In this paper, we investigate the effect of topical modeling on image annotation accuracy directly. We use the Pachinko Allocation Model (PAM) (Li and McCallum, 2006) to model the topics in a text, where keywords forming the dominant topic are assumed as our set of annotation keywords. Compared with previous topic modeling approaches, such as Latent Dirichlet allocation (LDA) or its improved variant Correlated Topic Model (CTM) (Blei and Lafferty, 2007), PAM captures correlations between all the topic pairs using a directed acyclic graph (DAG). It also supports finer-grained topic modeling, and has state-of-the-art performance on the tasks of document classification and topical keyword coherence. Given a text, we use the PAM model to infer a list of *super-topics* and *sub-topics* together with words weighted according to the likelihood that they belong to each of these topics. For each text, we retrieve the top words belonging to the dominant super-topic and sub-topic. We use 50 super-topics and 100 sub-topics as operating parameters for PAM, since these values were found to provide good results in previous work on topic modeling. Default values are used for other parameters in the model.

5.4 Supervised Learning

The three tagging methods target different aspects of what constitutes a good label for an image. We use them as features in a machine learning framework, and introduce a final rank attribute S(t_j), which is a linear combination of the reciprocals of the rank of each tag as given by each method,

$$S(t_j) = \sum_{m \in methods} \lambda_m \frac{1}{r^m_{t_j}}$$

where $r^m_{t_j}$ is the rank for tag t_j given by method m. The weight of each method λ_m is estimated from the training set using information gain values. Since our predicted variable (*mode* precision or recall) is continuous, we use the Support Vector Algorithm (nu-SVR) implementation of SVM (Chang and Lin, 2001) to perform regression analysis on the weights for each method via a radial basis function kernel. A ten-fold cross-validation is applied on the entire dataset of 300 images.

	Best				out-of-ten (oot)			
	Normal		Mode		Normal		Mode	
Models	P	R	P	R	P	R	P	R
Flickr picturability	6.32	6.32	78.57	78.57	35.61	35.61	92.86	92.86
Wikipedia Salience	6.40	6.40	7.14	7.14	35.19	35.19	92.86	92.86
Topic modeling	5.99	5.99	42.86	42.86	37.13	37.13	85.71	85.71
Combined (SVM)	6.87	6.87	67.49	67.49	37.85	37.85	100.00	100.00
Doc Title	6.40	6.40	75.00	75.00	18.97	18.97	82.14	82.14
*tf * idf*	5.94	5.94	14.29	14.29	38.40	38.40	78.57	78.57
Random	3.76	3.76	3.57	3.57	30.20	30.20	50.00	50.00
Upper bound (human)	12.23	12.07	81.48	81.48	82.44	81.55	100.00	100.00

Table 3: Results obtained on the Web dataset

6 Experiments and Evaluations

We evaluate the performance of each of the three tagging methods separately, followed by an evaluation of the combined method. Each system produces a ranked list of K words or collocations as tags assigned to a given image. A system can discretionary generate less (but not more) than K tags, depending on its confidence level.

For comparison, we implement three baselines: *tf*idf*, *Doc Title* and *Random*. For *tf*idf*, we use the British National Corpus to calculate the *idf* scores, while the frequency of a term is calculated from the entire text associated with an image. The *Doc Title* baseline is similar, except that the term frequency is calculated based on the title of the document. The *Random* baseline randomly selects words from a co-occurrence window of size K before and after an image as its annotation. Following other tagging methods, we apply a pre-processing stage, where we part-of-speech tag the text (to retain only nouns), followed by stemming. We also determine an upper bound, which is calculated as follows. For each image, the labels assigned by each of the five annotators are in turn evaluated against a gold standard consisting of the annotations of the other four annotators. The best performing annotator is then recorded. This process is repeated for each of the 300 images, and the average precision and recall are calculated. This represents an upper bound, as it is the best performance that a human can achieve on this dataset. Table 3 shows our experimental results.

Among the individual methods, the method implementing Flickr picturability has the highest individual score for *best* and *oot* modes, yielding a precision and recall of 78.57% and 92.86% respectively. The Wikipedia Saliency method also scores the highest (jointly with Flickr) in the *oot* mode, but for the *best* mode achieves a score only marginally better than the random baseline. A plausible explanation is that it tends to favor "all-inferring" over-specific labels, while the most frequently selected tags in mode pictures are typically more "picturable" than being specific (e.g. "train" for the mode picture in Table 1). The topic modeling method has mixed results: its scores for *oot* normal and mode are somewhat competitive with *tf*idf*, but it scores consistently lower than the DocTitle in the *best* subtask, possibly due to the absence of a more sophisticated reranking algorithm tailored for the image annotation task other than the intrinsic ranking mechanism in PAM. It is worth noting that the combined supervised system provides the overall best results (6.87%) on the *best* normal, and achieves a perfect precision and recall (100%) for *oot* mode, which means perfect agreement with the human tagging.

7 Comparison with Related Work

We also compare our work against (Feng and Lapata, 2008) as it allows for a direct comparison with models using both image and textual features under a standard evaluation framework. We obtained the BBC dataset used in their experiments, which consists of 3121 training and 240 testing images. In this dataset, images are implicitly tagged with captions by the author of the corresponding BBC article. The evaluations are run against these captions.

In their experiments, Feng and Lapata created four annotation models. The first two (tf*idf and Document Title) are the same as used in our baseline experiments. The third model (Lavrenko03) is an application of the continuous relevance model in (Jeon et al., 2003), trained with the BBC image features and captions. Finally, the forth (ExtModel) is an extension of the relevance model using additional information in auxiliary texts. Briefly, the model assumes a multiple Bernoulli distribution for words in a caption, and generates tags for a test image using a weighted combination of the accompanying document, caption and image features learned during training.

	Top 10			Top 15			Top 20		
Models	P	R	F1	P	R	F1	P	R	F1
*tf*idf*	4.37	7.09	5.41	3.57	8.12	4.86	2.65	8.89	4.00
DocTitle	9.22	7.03	7.20	9.22	7.03	7.20	9.22	7.03	7.20
Lavrenko03	9.05	16.01	11.81	7.73	17.87	10.71	6.55	19.38	9.79
ExtModel	14.72	27.95	19.82	11.62	32.99	17.18	9.72	36.77	15.39
Flickr picturability	12.13	22.82	15.84	9.52	26.82	14.05	8.23	29.80	12.90
Wikipedia Salience	11.63	21.89	15.18	9.28	26.20	13.70	7.81	29.41	12.35
Topic Modeling	11.42	21.49	14.91	9.28	26.20	13.70	7.86	29.57	12.42
Combined (SVM)	13.38	25.17	17.47	11.08	31.29	16.37	9.50	35.76	15.01

Table 4: Results obtained on the BBC dataset used in (Feng and Lapata, 2008)

The experimental setup is similar to the earlier section, but a few modifications are made for a fair and direct comparison. First, we extend our models coverage to include content words (i.e. nouns, verbs, adjectives) determined using the Tree Tagger (Schmid, 1994). Second, no collocations are used. Third, we adopt the evaluation framework used by Feng and Lapata to extract the top 10, 15 and 20 tags. Note that in our methods, the extraction of tags for a test image is only done on the document surrounding the image, after excluding the caption. As the number of negative examples (words not present in the caption) greatly outnumber the positive instances, we employ an undersampling method (Kubat and Matwin, 1997) to balance the dataset for training.

The results are shown in Table 4. Interestingly, all our unsupervised extraction-based models perform consistently above the supervised Lavrenko03 model, indicating that textual features are more informative than captions and image features taken together. Comparing with models using significantly less document information (tf*idf and Doc title), our models gain even greater advantage. Note that the title of any BBC article does not exceed 10 words, hence comparison is only meaningful given the top 10 tags retrieved.

Feng and Lapata used LDA to perform reranking of final candidates in their ExtModel. However, when used as a model alone, the PAM topic model achieved promising scores in all the categories, performing best for top 10 keywords (F1 of 14.91%). Flickr picturability stands out as the best performing unsupervised method, scoring the highest precision (12.13%, top 10), recall (29.80%, top 20) and F1 (15.84%, top 10).

Overall, this comparative evaluation yields some important insights. First, our combined model using SVM is statistically better ($p<0.1$ for top 10, 15, 20) than the Laverenko03 model, but not statistically different from the ExtModel. This demonstrates the effectiveness of textual-based models over traditional models trained with image features and captions. While it is intuitively clear that image features help in improving tagging performance, we show that mining only the text surrounding an image, where it exists, can yield a performance that is comparable to a state-of-the-art system that uses both textual and visual features. Moreover, an increase in complexity of a model by using more features may hinder its applicability to large datasets, but not necessarily improving annotation performance (Makadia et al., 2008). On this, text-based annotation models can provide a desirable compromise. For instance, our unsupervised models implementing Flickr picturability and Wikipedia Salience are able to extract annotations from a BBC article (average 133.85 tokens) in approximately 1 second and 20 seconds respectively.

8 Conclusions and Future Work

In this paper, we introduced several text-based extractive approaches for automatic image annotation and showed that they compare favorably with the state-of-the-art in image annotation using both text and image features. We believe our work has practical applications in mining and annotating images over the Web, where texts are naturally associated with images, and scalability is important. Our next direction seeks to derive robust annotation models using additional ontological knowledge-bases. We would also like to advance the the state-of-the-art by augmenting current textual models with image features.

Acknowledgments

This material is based in part upon work supported by the National Science Foundation CAREER award #0747340. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

Kobus Barnard and David Forsyth. 2001. Learning the semantics of words and pictures. In *Proceedings of International Conference on Computer Vision*.

David Blei and John Lafferty. 2007. A correlated topic model of science. In *Annals of Applied Statistics*, volume 1, pages 17–35.

Chih-Chung Chang and Chih-Jen Lin, 2001. *LIBSVM: a library for support vector machines*.

Brendan Collins, Jia Deng, Kai Li, and Li Fei-Fei. 2008. Towards scalable dataset construction: An active learning approach. In *Proceedings of European Conference on Computer Vision*.

Koen Deschacht and Marie-Francine Moens. 2007. Text analysis for automatic image annotation. In *Proceedings of the Association for Computational Linguistics*.

Pinar Duygulu, Kobus Barnard, Nando de Freitas, and David Forsyth. 2002. Object recognition as machine translation:learning a lexicon for a fixed image vocabulary. In *Proceedings of the 7th European Conference on Computer Vision*.

Yansong Feng and Mirella Lapata. 2008. Automatic image annotation using auxiliary text information. In *Proceedings of the Association for Computational Linguistics*.

Rob Fergus, Pietro Perona, and Andrew Zisserman. 2003. Object class recognition by unsupervised scale-invariant learning. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*.

Evgeniy Gabrilovich and Shaul Markovitch. 2007. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *International Joint Conferences on Artificial Intelligence*.

Michael Grubinger, Clough Paul, Mller Henning, and Deselaers Thomas. 2006. The iapr benchmark: A new evaluation resource for visual information systems. In *International Conference on Language Resources and Evaluation*.

Jiwoon Jeon, Victor Lavrenko, and R Manmatha. 2003. Automatic image annotation and retrieval using cross-media relevance models. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*.

Yohan Jin, Latifur Khan, Lei Wang, and Mamoun Awad. 2005. Image annotations by combining multiple evidence & wordnet. In *Proceedings of Annual ACM Multimedia*.

Miroslav Kubat and Stan Matwin. 1997. Addressing the curse of imbalanced training sets: one-sided selection. In *Proceedings of International Conference on Machine Learning*.

Li-Jia Li and Li Fei-Fei. 2008. Optimol: automatic online picture collection via incremental model learning. In *International Journal of Computer Vision*.

Wei Li and Andrew McCallum. 2006. Pachinko allocation: Dag-structured mixture models of topic correlations. In *Proceedings of the International Conference on Machine learning*.

Jia Li and James Wang. 2008. Real-time computerized annotation of pictures. In *Proceedings of International Conference on Computer Vision*.

Ameesh Makadia, Vladimir Pavlovic, and Sanjiv Kumar. 2008. A new baseline for image annotation. In *Proceedings of European Conference on Computer Vision*.

Diana McCarthy and Roberto Navigli. 2007. The semeval English lexical substitution task. In *Proceedings of the ACL Semeval workshop*.

Rada Mihalcea and Chee Wee Leong. 2009. Towards communicating simple sentences using pictorial representations. In *Machine Translation*, volume 22, pages 153–173.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into texts. In *Proceedings of Empirical Methods in Natural Language Processing*.

Rada Mihalcea, Courtney Corley, and Carlo Strapparava. 2006. Corpus-based and knowledge-based measures of text semantic similarity. In *Proceedings of Association for the Advancement of Artificial Intelligence*, pages 775–780.

Helmut Schmid. 1994. Probabilistic part-of-speech tagging using decision trees. In *Proceedings of the International Conference on New Methods in Language Processing*.

Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Ng. 2008. Cheap and fast - but is it good? evaluating non-expert annotations for natural language tasks. In *Proceedings of Empirical Methods in Natural Language Processing*.

Srihari and Burhans. 1994. Visual semantics: Extracting visual information from text accompanying pictures. In *Proceedings of the American Association for Artificial Intelligence*.

Munirathnam Srikanth, Joshua Varner, Mitchell Bowden, and Dan Moldovan. 2005. Exploiting ontologies for automatic image annotation. In *Proceedings of the ACM Special Interest Group on Research and Development in Information Retrieval*.

Luis von Ahn and Laura Dabbish. 2004. Labeling images with a computer game. In *Proceedings of the ACM Special Interest Group on Computer Human Interaction*.

Chong Wang, David Blei, and Li Fei-Fei. 2009. Simultaneous image classification and annotation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*.

Unsupervised Discriminative Language Model Training for Machine Translation using Simulated Confusion Sets

Zhifei Li and **Ziyuan Wang** and **Sanjeev Khudanpur** and **Jason Eisner**
Center for Language and Speech Processing
Johns Hopkins University
zhifei.work@gmail.com,{zwang40, khudanpur, eisner}@jhu.edu

Abstract

An *unsupervised* discriminative training procedure is proposed for estimating a language model (LM) for machine translation (MT). An English-to-English synchronous context-free grammar is derived from a baseline MT system to capture *translation alternatives*: pairs of words, phrases or other sentence fragments that potentially compete to be the translation of the same source-language fragment. Using this grammar, a set of impostor sentences is then created for each English sentence to *simulate* confusions that would arise if the system were to process an (unavailable) input whose correct English translation is that sentence. An LM is then trained to discriminate between the original sentences and the impostors. The procedure is applied to the IWSLT Chinese-to-English translation task, and promising improvements on a state-of-the-art MT system are demonstrated.

1 Discriminative Language Modeling

A language model (LM) constitutes a crucial component in many tasks such as machine translation (MT), speech recognition, information retrieval, handwriting recognition, etc. It assigns a priori probabilities to word sequences. In general, we expect a low probability for an ungrammatical or implausible word sequence. The dominant LM used in such systems is the so-called n-gram model, which is typically derived from a large corpus of target language text via maximum likelihood estimation, mitigated by some smoothing or regularization. Due to the Markovian assumptions implicit in n-gram models, however, richer linguistic and semantic dependencies are not well captured. Rosenfeld (1996) and Khudanpur and Wu (2000) address such shortcoming by using maximum entropy models with long-span features, while still working with a *locally* normalized left-to-right LM. The whole-sentence maximum entropy LM of Rosenfeld et al. (2001) proposes a *globally* normalized log-linear LM incorporating several sentence-wide features.

The n-gram as well as the whole-sentence model are *generative* or descriptive models of text. However, in a task like Chinese-to-English MT, the de facto role of the LM is to *discriminate* among the alternative English translations being contemplated by the MT system for a particular Chinese input sentence. We call the set of such alternative translations a *confusion set*. Since a confusion set is typically a minuscule subset of the set of all possible word sequences, it is arguably better to train the LM parameters so as to make the *best candidate* in the confusion set more likely than its competitors, as done by Roark et al. (2004) for speech recognition and by Li and Khudanpur (2008) for MT. Note that identifying the best candidate requires *supervised* training data—bilingual text in case of MT—which is expensive in many domains (e.g. weblog or newsgroup) and for most language pairs (e.g. Urdu-English).

We propose a novel discriminative LM in this paper: a globally normalized log-linear LM that can be trained in an *efficient* and *unsupervised* manner, using only monolingual (English) text.

The main idea is to exploit (translation) uncertainties inherent in an MT system to derive an English-to-English confusion grammar (CG), illustrated in this paper for a Hiero system (Chiang, 2007). From the *bilingual* synchronous context-free grammar (SCFG) used in Hiero, we extract a *monolingual* SCFG, with rules of the kind, $X \rightarrow \langle \text{strong tea}, \text{powerful tea} \rangle$ or

Coling 2008: Poster Volume, pages 656–664,
Beijing, August 2010

$X \rightarrow \langle \text{in } X_1, \text{in the } X_1 \rangle$. Thus our CG is also an SCFG that generates pairs of English sentences that differ from each other in ways that alternative English hypothesis considered during translation would differ from each other. This CG is then used to "translate" each sentence in the LM training corpus into what we call its *confusion set* — a set of other "sentences" with which that sentence would likely be confused by the MT system, were it to be the target translation of a source-language sentence. Sentences in the training corpus, each paired with its confusion set, are then used to train a discriminative LM to prefer the training sentences over the alternatives in their confusion sets.

Since the monolingual CG and the bilingual Hiero grammar are both SCFGs, the confusion sets are isomorphic with translation hypergraphs that are used by supervised discriminative training. The confusion sets thus *simulate* the supervised case, with a key exception: lack of any (Chinese) source-language information. Therefore, only target-side "language model" probabilities may be estimated from confusion sets.

We carry out this discriminative training procedure, and empirically demonstrate promising improvements in translation quality.

2 Discriminative LM Training

2.1 Whole-sentence Maximum Entropy LM

We aim to train a globally normalized log-linear language model $p_\theta(y)$ of the form

$$p_\theta(y) \quad = \quad \mathrm{Z}^{-1}\, e^{f(y)\cdot\theta} \tag{1}$$

where y is an English sentence, $f(y)$ is a vector of arbitrary features of y, θ is the (weight) vector of model parameters, and $\mathrm{Z} \stackrel{\text{def}}{=} \sum_{y'} e^{f(y')\cdot\theta}$ is a normalization constant. Given a set of English training sentences $\{y_i\}$, the parameters θ may be chosen to maximize likelihood, as

$$\theta^* \quad = \quad \arg\max_\theta \prod_i p_\theta(y_i). \tag{2}$$

This is the so called whole-sentence maximum entropy (WSME) language model[1] proposed by Rosenfeld et al. (2001). Training the model of (2) requires computing Z, a sum over all possible word sequences y' with any length, which is computationally intractable. Rosenfeld et al. (2001) approximate Z by random sampling.

[1]Note the contrast with the maximum entropy n-gram LM (Rosenfeld, 1996; Khudanpur and Wu, 2000), where the normalization is performed for each n-gram history.

2.2 Supervised Discriminative LM Training

In addition to the computational disadvantage, (2) also has a modeling limitation. In particular, in a task like MT, the primary role of the LM is to discriminate among alternative translations of a *given* source-language sentence. This set of alternatives is typically a minuscule subset of all possible target-language word sequences. Therefore, a better way to train the global log-linear LM, given bilingual text $\{(x_i, y_i)\}$, is to generate the *real* confusion set $\mathcal{N}(x_i)$ for each input sentence x_i using a specific MT system, and to adjust θ to discriminate between the reference translation y_i and $y' \in \mathcal{N}(x_i)$ (Roark et al., 2004; Li and Khudanpur, 2008).

For example, one may maximize the conditional likelihood of the bilingual training data as

$$\begin{aligned} \theta^* \quad &= \quad \arg\max_\theta \prod_i p_\theta(y_i \mid x_i) \\ &= \quad \arg\max_\theta \prod_i \frac{e^{f(x_i,y_i)\cdot\theta}}{\sum_{y'\in\mathcal{N}(x_i)} e^{f(x_i,y')\cdot\theta}}\,, \end{aligned} \tag{3}$$

which entails summing over *only* the candidate translations y' of the given input x_i. Furthermore, if the features $f(x_i, y)$ are depend on *only* the output y, i.e. on the English-side features of the bilingual text, the resulting discriminative model may be interpreted as a *language model*.

Finally, in a Hiero style MT system, if $f(x_i, y)$ depends on the target-side(s) of the bilingual rules used to construct y from x_i, we essentially have a *syntactic* LM.

2.3 Unsupervised Discriminative Training using Simulated Confusion Sets

While the supervised discriminative LM training has both computational and modeling advantages over the WSME LM, it relies on bilingual data, which is expensive to obtain for several domains and language pairs. For such cases, we propose a novel discriminative language model, which is

still a global log-linear LM with the modeling advantage and computational *efficiency* of (3) but requires only monolingual text $\{y_i\}$ for training θ. Specifically, we propose to modify (3) as

$$\theta^* = \arg\max_\theta \prod_i p_\theta(y_i \,|\, \mathcal{N}(y_i)) \quad (4)$$
$$= \arg\max_\theta \prod_i \frac{e^{f(y_i)\cdot\theta}}{\sum_{y' \in \mathcal{N}(y_i)} e^{f(y')\cdot\theta}},$$

where $\mathcal{N}(y_i)$ is a *simulated* confusion set for y_i obtained by applying a confusion grammar to y_i, as detailed in Section 3. Our hope is that $\mathcal{N}(y_i)$ resembles the actual confusion set $\mathcal{N}(x_i)$ that an MT system would generate if it were given the input sentence x_i.

Like (3), the maximum likelihood training of (4) does not entail the expensive computation of a global normalization constant Z, and is therefore very *efficient*. Unlike (3) however, where the input x_i for each output y_i is needed to create $\mathcal{N}(x_i)$, the model of (4) can be trained in an *unsupervised* manner with only $\{y_i\}$.

3 Unsupervised Discriminative Training of the Language Model for MT

The following is thus the proposed procedure for unsupervised discriminative training of the LM.

1. Extract a *confusion grammar* (CG) from the baseline MT system.
2. "Translate" each English sentence in the LM training corpus, using the CG as an English-to-English translation model, to generate a *simulated* confusion set.
3. Train a discriminative language model on the simulated confusion sets, using the corresponding original English sentences as the training references.

The trained model may then be used for actual MT decoding. We next describe each step in detail.

3.1 Extracting a Confusion Grammar

We assume a synchronous context free grammar (SCFG) formalism for the confusion grammar (CG). While the SCFG used by the MT system is bilingual, the CG we extract will be *monolingual*, with both the source and target sides being English. Some example CG rules are:

$$\begin{aligned} X &\rightarrow \langle \text{ strong tea , powerful tea } \rangle, \\ X &\rightarrow \langle X_0 \text{ at beijing , beijing 's } X_0 \rangle, \\ X &\rightarrow \langle X_0 \text{ of } X_1 \text{ , } X_0 \text{ of the } X_1 \rangle, \\ X &\rightarrow \langle X_0 \text{ 's } X_1 \text{ , } X_1 \text{ of } X_0 \rangle. \end{aligned}$$

Like a regular SCFG, a CG contains rules with different "arities" and reordering of the nonterminals (as shown in the last example) capturing the confusions that the MT system encounters when choosing word *senses*, *reordering patterns*, etc.

3.1.1 Extracting a Confusion Grammar from the Bilingual Grammar

The confusion grammar is derived from the MT system's bilingual grammar. In Hiero, the bilingual rules are of the form $X \rightarrow \langle c, e \rangle$, where both c and e may contain (a matched number of) nonterminal symbols. For every c which appears on the source-side of two different Hiero rules $X \rightarrow \langle c, e_1 \rangle$ and $X \rightarrow \langle c, e_2 \rangle$, we extract two CG rules, $X \rightarrow \langle e_1, e_2 \rangle$ and $X \rightarrow \langle e_2, e_1 \rangle$, to capture the confusion the MT system would face were it to encounter c in its input. For each Hiero rule $X \rightarrow \langle c, e \rangle$, we also extract $X \rightarrow \langle e, e \rangle$, the *identity* rule. Therefore, if a pattern c appears with $|E|$ different translation options, we extract $|E|^2$ different CG rules from c. In our current work, the rules of the CG are unweighted.

3.1.2 Test-set Specific Confusion Grammars

If the bilingual grammar contains all the rules that are extractable from the bilingual training corpus, the resulting confusion grammar is likely to be huge. As a way of reducing computation, the bilingual grammar can be restricted to a specific test set, and only rules used by the MT system for translating the test set used for extracting the CG.[2]

To economize further, one may extract a CG from the translation *hypergraphs* that are generated for the test-set. Recall that a *node* in a hypergraph corresponds to a specific source (Chinese) span, and the node has many incident *hyperedges*, each associated with a different bilin-

[2]Test-set specific CGs are of course only practical for offline applications.

gual rule. Therefore, all the bilingual rules associated with the incoming hyperedges of a given node translate the same Chinese string. At each hypergraph node, we extract CG rules to represent the competing English sides as described above. Note that even though different rules associated with a node may have different "arity," we extract CG rules only from pairs of bilingual rules that have the same arity.

A CG extracted from only the bilingual rule pairs incident on the same node in the test hypergraphs is, of course, much smaller than a CG extracted from the entire bilingual grammar. It is also more suitable for our task, since the test hypergraphs have already benefited from a baseline n-gram LM and pruning, removing all confusions that are easily resolved (rightly or wrongly) by other system components.

3.2 Generating Simulated Confusion Sets

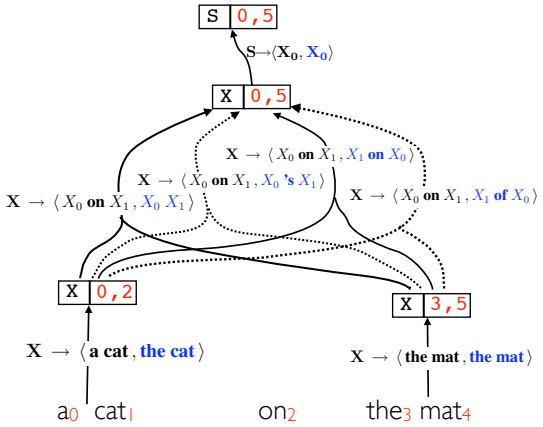
For each English sentence y in the training corpus, we use the extracted CG to produce a simulated confusion set $\mathcal{N}(y)$. This is done like a regular MT decoding pass, because we can treat the CG as a Hiero style "translation" grammar[3] for an English-to-English translation system.

Since the CG is an SCFG, the confusion set $\mathcal{N}(y)$ generated for a sentence y is a *hypergraph*, encoding not only the alternative sentences y' but also the hierarchical derivation tree for each y' from y (e.g., which phrase in y has been replaced with what in y'). As usual, many different derivation trees d may correspond to the same string/sentence y' due to spurious ambiguity. We use $\mathrm{D}(y)$ to denote the set of derivations d, which is a hypergraph representation of $\mathcal{N}(y)$.

Figure 1 presents an example confusion hypergraph for the English sentence y ="*a cat on the mat*," containing four alternative hypotheses:

$$\begin{array}{lcl} X & \rightarrow & \langle \text{a cat} \,,\, \text{the cat} \rangle \\ X & \rightarrow & \langle \text{the mat} \,,\, \text{the mat} \rangle \\ X & \rightarrow & \langle X_0 \text{ on } X_1 \,,\, X_0 \; X_1 \rangle \\ X & \rightarrow & \langle X_0 \text{ on } X_1 \,,\, X_0 \text{ 's } X_1 \rangle \\ X & \rightarrow & \langle X_0 \text{ on } X_1 \,,\, X_1 \text{ on } X_0 \rangle \\ X & \rightarrow & \langle X_0 \text{ on } X_1 \,,\, X_1 \text{ of } X_0 \rangle \\ S & \rightarrow & \langle X_0 \,,\, X_0 \rangle \end{array}$$

(a) An example confusion grammar.



(b) An example hypergraph generated by the confusion grammar of (a) for the input sentence "*a cat on the mat*."

Figure 1: **Example confusion grammar and simulated confusion hypergraph.** Given an input sentence $y =$ "*a cat on the mat*," the confusion grammar of (a) generates a hypergraph $\mathrm{D}(y)$ shown in (b), which represents the confusion set $\mathcal{N}(y)$ containing four alternative sentences y'.

$\mathcal{N}(y) = \{$ "*the cat the mat*," "*the cat 's the mat*," "*the mat of the cat*," "*the mat on the cat*"$\}$.

Notice that each competitor $y' \in \mathcal{N}(y)$ can be regarded as the result of a "round-trip" translation $y \rightarrow x \rightarrow y'$, in which we reconstruct a possible Chinese source sentence x that our Hiero bilingual grammar could translate into both y and y'.[4] We will train our LM to prefer y, which was actually observed. Our CG-based round-trip forces $x \rightarrow y'$ to use the *same* hierarchical segmentation of x as $y \rightarrow x$ did. This constraint leads to efficient training but artificially reduces the diversity

[3] To make sure that we produce at least one derivation tree for each y, we need to add to the CG the following two glue rules, as done in Hiero (Chiang, 2007).

$$\begin{array}{lcl} S & \rightarrow & \langle X_0 \,,\, X_0 \rangle \,, \\ S & \rightarrow & \langle S_0 \; X_1 \,,\, S_0 \; X_1 \rangle \,. \end{array}$$

We also add an out of vocabulary rule $X \rightarrow \langle word, \text{oov} \rangle$ for each *word* in y and set the cost of this rule to a high value so that the OOV rule will get used only when the CG does not know how to "translate" the *word*.

[4] This is because of the way we construct our CG from the Hiero grammar. However, the identity and glue rules in our CG allow almost any portion of y to be preserved untranslated through the entire $y \rightarrow x \rightarrow y'$ process. Much of y will necessarily be preserved in the situation where the CG is extracted from a small test set and hence has few non-identity rules. See (Li, 2010) for further discussion.

of $\mathcal{N}(y)$. In other recent work (Li et al., 2010), we have taken the round-trip view more seriously, by imputing *likely* source sentences x and translating them back to *separate*, *weighted* confusion forests $\mathcal{N}(y)$, *without* any same-segmentation constraint.

3.3 Confusion-based Discriminative Training

With the training sentences y_i and their simulated confusion sets $\mathcal{N}(y_i)$ — represented as hypergraphs $\mathrm{D}(y_i)$) — we can perform the discriminative training using any of a number of procedures such as MERT (Och, 2003) or MIRA as used by Chiang et al. (2009). In our paper, we use hypergraph-based minimum risk (Li and Eisner, 2009),

$$\begin{aligned}\theta^* &= \arg\min_\theta \sum_i \mathrm{Risk}_\theta(y_i) \qquad (5)\\ &= \arg\min_\theta \sum_i \sum_{d\in \mathrm{D}(y_i)} \mathrm{L}(\mathrm{Y}(d), y_i) p_\theta(d \,|\, \mathrm{D}(y_i)),\end{aligned}$$

where $\mathrm{L}(y', y_i)$ is the loss (e.g negated BLEU) incurred by producing y' when the true answer is y_i, $\mathrm{Y}(d)$ is the English *yield* of a derivation d, and $p_\theta(d \,|\, \mathrm{D}(y_i))$ is defined as,

$$p_\theta(d \,|\, \mathrm{D}(y_i)) = \frac{e^{f(d)\cdot\theta}}{\sum_{d\in \mathrm{D}(y_i)} e^{f(d)\cdot\theta}}, \qquad (6)$$

where $f(d)$ is a feature vector over d. We will specify the features in Section 5, but in general they should be defined such that the training will be efficient and the actual MT decoding can use them conveniently.

The objective of (5) is differentiable and thus we can optimize θ by a gradient-based method. The risk and its gradient on a hypergraph can be computed by using a second-order expectation semiring (Li and Eisner, 2009).

3.3.1 Iterative Training

In practice, the full confusion set $\mathcal{N}(y)$ defined by a confusion grammar may be too large and we have to perform pruning when training our model. But the pruning itself may depend on the model that we aim to train. How do we solve this circular dependency problem? We adopt the following procedure. Given an initial model θ, we generate a hypergraph (with pruning) for each y, and train an optimal θ^* of (5) on these hypergraphs. Then, we use the optimal θ^* to *regenerate* a hypergraph for each y, and do the training again. This iterates until convergence. This procedure is quite similar to the k-best MERT (Och, 2003) where the training involves a few iterations, and each iteration uses a new k-best list generated using the latest model.

3.4 Applying the Discriminative LM

First, we measure the goodness of our language model in a simulated task. We generate simulated confusion sets $\mathcal{N}(y)$ for some held out English sentences y, and test how well $p_\theta(d \,|\, \mathrm{D}(y))$ can recover y from $\mathcal{N}(y)$. This is merely a proof of concept, and may be useful in deciding which features $f(d)$ to employ for discriminative training.

The intended use of our model is, of course, for actual MT decoding (e.g., translating Chinese to English). Specifically, we can add the discriminative model into an MT pipeline as a feature, and tune its weight relative to other models in the MT system, including the baseline n-gram LM.

4 Related and Similar Work

The detailed relation between the proposed procedure and other language modeling techniques has been discussed in Sections 1 and 2. Here, we review two other methods that are related to our method in a broader context.

4.1 Unsupervised Training of Global Log-linear Models

Our method is similar to the contrastive estimation (CE) of Smith and Eisner (2005) and its successors (Poon et al., 2009). In particular, our confusion grammar is like a *neighborhood function* in CE. Also, our goal is to improve both efficiency and accuracy, just as CE does. However, there are two important differences. First, the neighborhood function in CE is manually created based on human insights about the particular task, while our neighborhood function, generated by the CG, is automatically *learnt* (e.g., from the bilingual grammar) and specific to the MT system being used. Therefore, our neighborhood function is more likely to be informative and adaptive to the task. Secondly, when tuning θ, CE uses the maximum likelihood training, but we use the minimum

risk training of (5). Since our training uses a task-specific loss function, it is likely to perform better than maximum likelihood training.

4.2 Paraphrasing Models

Our method is also related to methods for training paraphrasing models (Quirk et al., 2004; Bannard and Callison-Burch, 2005; Callison-Burch et al., 2006; Madnani et al., 2007). Specifically, the form of our *confusion grammar* is similar to that of the *paraphrase model* they use, and the ways of extracting the grammar/model are also similar as both employ a second language (e.g., Chinese in our case) as a *pivot*. However, while a "translation" rule in a paraphrase model is expected to contain a pair of phrases that are good alternatives for each other, a confusion rule in our CG is based on an MT system processing unseen test data and contains pairs of phrases that are typically bad (and only rarely good) alternatives for each other.

The motivation and goal are also different. For example, the goal of Bannard and Callison-Burch (2005) is to extract paraphrases with the help of parallel corpora. Callison-Burch et al. (2006) aim to improve MT quality by adding paraphrases in the translation table, while Madnani et al. (2007) aim to improve the minimum error rate training by adding the automatically generated paraphrases into the English reference sets. In contrast, our motivation is to train a *discriminative* language model to improve MT (by using the confusion grammar to decide what alternatives the model should learn to discriminate).

5 Experimental Results

We have applied the confusion-based discriminative language model (CDLM) to the IWSLT 2005 Chinese-to-English text translation task[5] (Eck and Hori, 2005). We see promising improvements over an n-gram LM for a solid **Joshua**-based baseline system (Li et al., 2009).

5.1 Data Partitions for Training & Testing

Four kinds of data are used for CDLM training:

[5]This is a relatively small task compared to, say, the NIST MT tasks. We worked on it for a proof-of-concept. Having been successful, we are now investigating larger MT tasks.

	Data Usage	# sentences	
		ZH	EN
Set1	TM & LM training	40k	40k
Set2	Min-risk training	1006	1006×16
Set3	CDLM training	—	1006×16
Set4	Test	506	506×16

Table 1: **Data sets used.** Set1 contains translation-equivalent Chinese-English sentence pairs, while for each Chinese sentence in Set2 and Set4, there are 16 English translations. Set3 happens to be the English side of Set2 due to lack of additional in-domain English text, but this is not noteworthy; Set3 could be any in-domain target-language text corpus.

Set1 a bilingual training set on which 10 individual MT system components are trained,

Set2 a small bilingual, in-domain set for tuning relative weights of the system components,

Set3 an in-domain monolingual target-language corpus for CDLM training, and

Set4 a test set on which improvements in MT performance is measured.

We partition the IWSLT data into four such subsets as listed in Table 1.

5.2 Baseline MT System

Our baseline translation model components are estimated from 40k pairs of utterances from the travel domain, called Set1 in Table 1. We use a *5-gram* language model with modified Kneser-Ney smoothing (Chen and Goodman, 1998), trained on the English side of Set1, as our baseline LM.

The baseline MT system comprises 10 component models (or "features") that are standard in Hiero (Chiang, 2007), namely the baseline language model (BLM) feature, three baseline translation model features, one word-insertion penalty (WP) feature, and five *arity* features — three to count how many rules with an arity of zero/one/two are used in a derivation, and two to count how many times the unary and binary glue rules are used in a derivation. The relative weights of these 10 features are tuned via hypergraph-based minimum risk training (Li and Eisner, 2009) on the *bilingual* data Set2.

The resulting MT system gives a BLEU score of 48.5% on Set4, which is arguably a solid baseline.

5.3 Unsupervised Training of the CDLM

We extract a test-set specific CG from the hypergraphs obtained by decoding Set2 and Set4, as described in Section 3.1.2. The number of rules in the bilingual grammar and the CG are about 167k and 1583k respectively. The CG is used as the "translation" model to generate confusion hypergraphs for sentences in Set3.

Two CDLMs, corresponding to different feature sets $f(d)$ in equation (6), were trained.

Only n-gram LM Features: We consider a CDLM with only two features $f(d)$: a baseline LM feature (BLM) that equals the 5-gram probability of $\mathrm{Y}(d)$ and a word penalty feature (WP) equal to the length of $\mathrm{Y}(d)$.

Target-side Rule Bigram Features[6]: For each CG rule used in d, we extract counts of bigrams that appear on the target-side of the CG rule. For example, if the confusion rule $X \rightarrow \langle\, X_0 \text{ of } X_1 \,,\, X_0 \text{ of the } X_1 \,\rangle$ is used in d, the bigram features in $f(d)$ whose counts are incremented are: "*X of*," "*of the*" and "*the X*."[7] Note that the indices on the nonterminals in the rule have been removed. To avoid very rare features, we only consider the 250 most freqent terminal symbol (English words) in the English of Set1 and map all other terminal symbols into a single class. Finally, we replace the identities of words with their dominant POS tags. These restrictions result in 525 target-side rule bigram (TsRB) features $f(d)$ in the model of (6).

For each choice of the feature vector $f(d)$, be it 2- or 527-dimensional, we use the training procedure of Section 3.3.1 to iteratively minimize the objective of (5) and get the CDLM parameter θ^*.

Note that each English sentence in Set3 has 15 other paraphrases. We generate a separate confusion hypergraph $\mathrm{D}(y)$ for each English sentence y, but for each such hypergraph we use both y and its 15 paraphrases as "reference translations" when computing the risk $\mathrm{L}(\mathrm{Y}(d), \{y\})$ in (5).[8]

[6]Note that these features are novel in MT.

[7]With these target-side rule-based features, our LM is essentially a *syntactic* LM, not just an LM on English strings.

[8]We take unfair advantage of this unusual dataset to combat an unrelated complication—a seemingly problematic instability in the minimum risk training procedure.

As an illustration of this problem, we note that in supervised tuning of the baseline MT system ($|f(d)|$=10) with 500 sentences from Set2, the BLEU score on Set4 varies from 38.6% to 44.2% to 47.8% if we use 1, 4 and 16 reference translations during the supervised training respectively. We choose a system tuned on 16 references on Set2 as our baseline. In order not to let the unsupervised CDLM training suffer from this unrelated limitation of the tuning procedure, we give it too the benefit of being able to compute risk on Set3 using y plus its 15 paraphrases.

We wish to emphasize that this trait of Set3 having 15 paraphrases for each sentence is otherwise unnecessary, and *does not* detract much from the main claim of this paper.

5.4 Results on Monolingual Simulation

We first probe how our novel CDLM performs as a language model itself. One usually uses the perplexity of the LM on some unseen text to measure its goodness. But since we did not optimize the CDLM for likelihood, we instead examine how it performs in discriminating between a good English sentence and sentences with which the MT system may confuse that sentence. The test is performed as follows. For each test English sentence y of Set4, the confusion grammar defines a full confusion set $\mathcal{N}(y)$ via a hypergraph $\mathrm{D}(y)$. We use a LM to pick the most likely y^* from $\mathcal{N}(y)$, and then compute its BLEU score by using y and its 15 paraphrase sentences as references. The higher the BLEU, the better is the LM in picking out a good translation from $\mathcal{N}(y)$.

Table 2 shows the results[9] under a regular n-gram LM and the two CDLMs described in Section 5.3.

The baseline LM (BLM) entails no weight optimization a la (5) on Set3. The CDLM with the BLM and word pentaly (WP) features improves over the baseline LM. Compared to either of them, the CDLM with the target-side rule bigram features (TsRB) performs dramatically better.

5.5 Results on MT Test Data

We now examine how our CDLM performs during actual MT decoding. To incorporate the CDLM into MT decoding, we add the log-probability (6) of a derivation d under the CDLM as an additional

[9]Note that the scores in Table 2 are very low compared to scores for actual translation from Chinese shown in Table 3. This is mainly because in this monolingual simulation, the LM is the only model used to rank the $y' \in \mathcal{N}(y)$. Said differently, y^* is being chosen in Table 2 entirely for its fluency with no consideration whatsoever for its adequacy.

LM used for rescoring	Features used BLM	WP	TsRB	BLEU on Set4
Baseline LM	✓			12.8
CDLM	✓	✓		14.2
CDLM	✓	✓	✓	25.3

Table 2: **BLEU scores in monolingual simulations.** Rescoring the *confusion sets* of English sentences created using the CG shows that the CDLM with TsRB features recovers hypotheses much closer to the sentence that generated the confusion set than does the baseline n-gram LM.

Model used for rescoring	Features used 10 models	TsRB	BLEU on Set4
Joshua	✓		48.5
+ CDLM	✓	✓	49.5

Table 3: **BLEU scores on the test set.** The baseline MT system has ten models/features, and the proposed system has one *additional* model, the CDLM. Note that for the CDLM, only the TsRB features are used during MT decoding.

feature, on top of the 10 features already present in baseline MT system (see Section 5.2). We then (re)tune relative weights for these 11 features on the *bilingual* data Set2 of Table 1.

Note that the MT system also uses the BLM and WP features whose weights are now retuned on Set2. Therefore, when integrating a CDLM into MT decoding, it is mathematically equivalent to use only the TsRB features of the CDLM, with the corresponding weights as estimated alongside its "own" BLM and WP features during unsupervised discriminative training on Set3.

Table 3 reports the results. A BLEU score improvement of 1% is seen, reinforcing the claim that the unsupervised CDLM helps select better translations from among the system's alternatives.

5.6 Goodness of Simulated Confusion Sets

The confusion set $\mathcal{N}(y)$ generated by applying the CG to an English sentence y aims to simulate the real confusion set that would be generated by the MT system if the system's input was the Chinese sentence whose English translation is y. We investigate, in closing, how much the simulated confusion set resembles to the real one. Since we know the actual input-output pairs (x_i, y_i) for Set4, we generate two confusion sets: the simulated set $\mathcal{N}(y_i)$ and the real one $\mathcal{N}(x_i)$.

One way to measure the goodness of $\mathcal{N}(y_i)$ as a proxy for $\mathcal{N}(x_i)$, is to extract the n-gram types witnessed in the two sets, and compute the ratio of the number of n-grams in the intersection to the number in their union. Another is to measure the precision and recall of $\mathcal{N}(y_i)$ relative to $\mathcal{N}(x_i)$.

n-gram	Precision	Recall
unigram	36.5%	48.2%
bigram	10.1%	12.8%
trigram	3.7%	4.6%
4-gram	2.0%	2.4%

Table 4: **n-gram precision and recall of simulated confusion sets** relative to the true confusions when translating Chinese sentences. The n-grams are collected from k-best strings in both cases, with $k = 100$. The precision and recall change little when varying k.

Table 4 presents such precision and recall figures. For convenience, the n-grams are collected from the 100-best strings, instead of the hypergraph $\mathrm{D}(y_i)$ and $\mathrm{D}(x_i)$. Observe that the simulated confusion set does a reasonably good job on the real unigram confusions but the simulation needs improving for higher order n-grams.

6 Conclusions

We proposed a novel procedure to discriminatively train a globally normalized log-linear language model for MT, in an efficient and unsupervised manner. Our method relies on the construction of a confusion grammar, an English-to-English SCFG that captures translation alternatives that an MT system may face when choosing a translation for a given input. For each English training sentence, we use this confusion grammar to generate a simulated confusion set, from which we train a discriminative language model that will prefer the original English sentence over sentences in the confusion set. Our experiments show that the novel CDLM picks better alternatives than a regular n-gram LM from simulated confusion sets, and improves performance in a real Chinese-to-English translation task.

7 Acknowledgements

This work was partially supported by the National Science Foundation via grants No SGER-0840112 and RI-0963898, and by the DARPA GALE program. The authors thank Brian Roark and Damianos Karakos for insightful discussions.

References

Bannard, Colin and Chris Callison-Burch. 2005. Paraphrasing with bilingual parallel corpora. In *ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 597–604, Morristown, NJ, USA. Association for Computational Linguistics.

Callison-Burch, Chris, Philipp Koehn, and Miles Osborne. 2006. Improved statistical machine translation using paraphrases. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 17–24, Morristown, NJ, USA. Association for Computational Linguistics.

Chen, Stanley F. and Joshua Goodman. 1998. An empirical study of smoothing techniques for language modeling. Technical report.

Chiang, David, Kevin Knight, and Wei Wang. 2009. 11,001 new features for statistical machine translation. In *NAACL*, pages 218–226.

Chiang, David. 2007. Hierarchical phrase-based translation. *Computational Linguistics*, 33(2):201–228.

Eck, Matthias and Chiori Hori. 2005. Overview of the iwslt 2005 evaluation campaign. In *In Proc. of the International Workshop on Spoken Language Translation*.

Khudanpur, Sanjeev and Jun Wu. 2000. Maximum entropy techniques for exploiting syntactic, semantic and collocational dependencies in language modeling. In *Computer Speech and Language*, number 4, pages 355–372.

Li, Zhifei and Jason Eisner. 2009. First- and second-order expectation semirings with applications to minimum-risk training on translation forests. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 40–51, Singapore, August. Association for Computational Linguistics.

Li, Zhifei and Sanjeev Khudanpur. 2008. Large-scale discriminative n-gram language models for statistical machine translation. In *AMTA*, pages 133–142.

Li, Zhifei, Chris Callison-Burch, Chris Dyer, Juri Ganitkevitch, Sanjeev Khudanpur, Lane Schwartz, Wren Thornton, Jonathan Weese, and Omar. Zaidan. 2009. Joshua: An open source toolkit for parsing-based machine translation. In *WMT09*, pages 26–30.

Li, Zhifei, Ziyuan Wang, Jason Eisner, and Sanjeev Khudanpur. 2010. Minimum imputed risk training for machine translation. In review.

Li, Zhifei. 2010. Discriminative training and variational decoding in machine translation via novel algorithms for weighted hypergraphs. PHD Dissertation, Johns Hopkins University.

Madnani, Nitin, Necip Fazil Ayan, Philip Resnik, and Bonnie J. Dorr. 2007. Using paraphrases for parameter tuning in statistical machine translation. In *Proceedings of the Workshop on Statistical Machine Translation*, Prague, Czech Republic, June. Association for Computational Linguistics.

Och, Franz Josef. 2003. Minimum error rate training in statistical machine translation. In *ACL*, pages 160–167.

Poon, Hoifung, Colin Cherry, and Kristina Toutanova. 2009. Unsupervised morphological segmentation with log-linear models. In *NAACL '09: Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 209–217, Morristown, NJ, USA. Association for Computational Linguistics.

Quirk, Chris, Chris Brockett, and William Dolan. 2004. Monolingual machine translation for paraphrase generation. In *In Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 142–149.

Roark, Brian, Murat Saraclar, Michael Collins, and Mark Johnson. 2004. Discriminative language modeling with conditional random fields and the perceptron algorithm. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04), Main Volume*, pages 47–54, Barcelona, Spain, July.

Rosenfeld, Roni, Stanley F. Chen, and Xiaojin Zhu. 2001. Whole-sentence exponential language models: a vehicle for linguistic-statistical integration. *Computers Speech and Language*, 15(1).

Rosenfeld, Roni. 1996. A maximum entropy approach to adaptive statistical language modeling. In *Computer Speech and Language*, number 3, pages 187–228.

Smith, Noah A. and Jason Eisner. 2005. Contrastive estimation: Training log-linear models on unlabeled data. In *Proceedings of the Association for Computational Linguistics (ACL 2005)*, Ann Arbor, Michigan.

Combining Constituent and Dependency Syntactic Views for Chinese Semantic Role Labeling

Shiqi Li[1], Qin Lu[2], Tiejun Zhao[1], Pengyuan Liu[3] and Hanjing Li[1]
[1]School of Computer Science and Technology,
Harbin Institute of Technology
{sqli,tjzhao,hjlee}@mtlab.hit.edu.cn
[2]Department of Computing,
The Hong Kong Polytechnic University
csluqin@comp.polyu.edu.hk
[3]Institute of Computational Linguistics,
Peking University
liupengyuan@pku.edu.cn

Abstract

This paper presents a novel feature-based semantic role labeling (SRL) method which uses both constituent and dependency syntactic views. Comparing to the traditional SRL method relying on only one syntactic view, the method has a much richer set of syntactic features. First we select several important constituent-based and dependency-based features from existing studies as basic features. Then, we propose a statistical method to select discriminative combined features which are composed by the basic features. SRL is achieved by using the SVM classifier with both the basic features and the combined features. Experimental results on Chinese Proposition Bank (CPB) show that the method outperforms the traditional constituent-based or dependency-based SRL methods.

1 Introduction

Semantic role labeling (SRL) is a major method in current semantic analysis which is important to NLP applications. The SRL task is to identify semantic roles (or arguments) of each predicate and then label them with their functional tags, such as 'Arg0' and 'ArgM' in PropBank (Palmer et al., 2005), or 'Agent' and 'Patient' in FrameNet (Baker et al., 1998).

The significance of syntactic analysis in SRL has been proven by (Gildea and Palmer, 2002; Punyakanok et al., 2005), and syntactic parsing has been applied by almost all current studies. In terms of syntactic representations, the SRL approaches are mainly divided into three categories: constituent-based, chunk-based and dependency-based. Constituent-based SRL has been studied intensively with satisfactory results. Chunk-based SRL has been found to be less effective than the constituent-based by (Punyakanok et al., 2005). In recent years, the dependency-based SRL has been greatly promoted by the CoNLL shared tasks on semantic parsing (Hajic et al., 2009). However, there is not much research on combined use of different syntactic views (Pradhan et al., 2005), on the feature level of SRL.

This paper introduces a novel method for Chinese SRL utilizing both constituent-based and dependency-based features. The method takes constituent as the basic unit of argument and adopts the labeling of PropBank. It follows the prevalent feature-based SRL methods to first turn predicate-argument pairs into flat structures by well-defined linguistic features, and then uses machine learning methods to predict the semantic labels. The method also involves two classification phases: semantic role identification (SRI) and semantic role classification (SRC). In addition, a heuristic-based pruning preprocessing (Xue and Palmer, 2004) is used to filter out a lot of apparently inappropriate constituents at the beginning.

Coling 2008: Poster Volume, pages 665–673,
Beijing, August 2010

And it has been widely reported that, in feature-based SRL, the performance can be improved by adding several combined features each of which is composed by two single features (Xue and Palmer, 2004; Toutanova et al., 2005; Zhao et al., 2009). Thus, in this work, we exploit combined use of both constituent-based and dependency-based features in addition to using features of singular types of syntactic view. We propose a statistical method to select effective combined features using both constituent-based and dependency-based features to make full use of two syntactic views.

2 Related Work

In recent years, many advances have been made on SRL using singular syntactic view, such as constituent (Gildea and Jurafsky, 2002; Xue and Palmer, 2004; Surdeanu et al., 2007), dependency (Hacioglu, 2004; Johansson and Nugues, 2008; Zhao et al., 2009), and CCG (Chen and Rambow, 2003; Boxwell et al, 2009). However, there are few studies on the use of multiple syntactic views. We briefly review the relevant studies of SRL using multiple syntactic views as follows.

Pradhan et al. (2005) built three semantic role labelers using constituent, dependency and chunk syntactic views, and then heuristically combined them at the output level. The method was further improved in Pradhan et al. (2008) which trains two semantic role labelers for constituents and dependency separately, and then uses the output of the two systems as additional features in another labeler using chunk parsing. The result shows an improvement to each labeler alone. A possible reason for the improvement is that the errors caused by different syntactic parsers are compensated. Yet, the features of different syntactic views can hardly complement each other in labeling. And the complexity of using multiple syntactic parsers is extremely high. Hacioglu (2004) proposed a SRL method to combine constituent and dependency syntactic views where the dependency parses are ob-tained through automatic mapping of constitu-ent parses. It uses the constituent parses to get candidates and then, the dependency parses to label them.

Boxwell et al. (2009) proposed a SRL method using features of three syntactic views: CCG, CFG and dependency. It primarily uses CCG-based features associated with 4 CFG-based and 2 dependency-based features. The combination of these syntactic views leads to a substantial performance improvement. Nguyen et al. (2009) proposed a composite kernel based on both constituent and dependency syntactic views and achieved a significant improvement in a relation extraction application.

3 Design Principle and Basic Features

Compared to related work, the proposed method integrates the constituent and dependency views in a collaborative manner. First, we define a basic feature set containing features from constituent and dependency syntactic views. Then, to make better use of two syntactic views, we introduce a statistical method to select effective combined features from the basic feature set. Finally we use both the basic features and the combined features to identify and label arguments. One of the drawbacks of the related work is the considerable complexity caused by multiple syntactic parsing processes. In our method, the cost of syntactic parsing will increase only slightly as we derive dependency parsing from constituent parsing using a constituent-to-dependency converter instead of using an additional dependency parser.

In our method, the feature set used for SRL consists of two parts: the basic feature set and the combined feature set built upon the basic feature set. The basic feature set can be further divided into constituent-based features and dependency-based features. Constituent features focus on hierarchical relations between multi-word constituents whereas dependency features focus on dependencies between individual words, as shown in Figure 1. Take the predicate '提高' (increased) as an example, in Figure 1(a), the NP constituent '中国的地位' (China's position) is labeled as 'Arg0'. The argument and the predicate are connected by the path of node types: 'NP-IP-VP-VP'. But in Figure 1(b), the individual word '地位' (position) is labeled as 'Arg0'. And the connection between the argument and the predicate is only one edge with the relation 'nsubj', which is more explicit than the path in the constituent structure. So the two syntactic views can complement each other on different linguistic units.

3.1 Constituent-Based Features

As a prevalent syntactic feature set for SRL, constituent-based features have been extensively studied by many researchers. In this work, we simply take 26 constituent-based features tested by existing studies, and add 8 new features define by us. Firstly, the 26 constituent-based features used by others are:

- The seven "standard" features: *predicate* (c1), *path* (c2), *phrase type* (c3), *position* (c4), *voice* (c5), *head word* (c6) and *predicate subcategorization* (c7) features proposed by (Gildea and Jurafsky, 2002).
- *Syntactic frame* (c8) feature from (Xue and Palmer, 2004).
- *Head word POS* (c9), *partial path* (c10), *first/last word in constituent* (c11/c12), *first/last POS in constituent* (c13/c14), *left/right sibling constituent* (c15/c16), *left/right sibling head* (c17/c18), *left/right sibling POS* (c19/c20), *constituent tree distance* (c21) and *temporal cue words* (c22) features from (Pradhan et al., 2004).
- *Predicate POS* (c23), *argument's parent constituent* (c24), *argument's parent constituent head* (c25) and *argument's parent constituent POS* (c26) inspired by (Pradhan et al., 2004).

Secondly, the 8 new features that we define are (we take the 'Arg0' node in Figure 1(a) as the example to illustrate them):

- *Locational cue words* (c27): a binary feature indicating whether the constituent contains location cue words, similar to the *temporal cue words* (c22). This feature is defined to distinguish the arguments with the 'ArgM-LOC' type from others.
- *POS pattern of argument's children* (c28): the left-to-right chain of the POS tags of the argument's children, e.g. 'NR-DEG-NN'.
- *Phrase type pattern of argument's children* (c29): the left-to-right chain of the phrase type labels of the argument's children, similar with the *POS pattern of argument's children* (c28), e.g. 'DNP-NP'.
- *Type of LCA and left child* (c30): The phrase type of the Lowest Common Ancestor (LCA) combined with its left child, e.g. 'IP-NP'.
- *Type of LCA and right child* (c31): The phrase type of the LCA combined with its right child, e.g. 'IP-VP'.

Three features: *bag of words of path* (c32), *bag of words of POS pattern* (c33) and *bag of words of type pattern* (c34), for generalizing three sparse features: *path* (c2), *POS pattern of argument's children* (c28) and *phrase type pattern of argument's children* (c29) by the bag-of-words representation.

3.2 Dependency-Based Features

The dependency parse can effectively represent the head-dependent relationship between words, yet, it lacks constituent information. If we want to label constituents using dependency-based features, we should firstly map each constituent to one or more appropriate words in the dependency tree. In this paper, we use the head word of a constituent to represent the constituent in the dependency parses.

The selection method of dependency-based features is similar to the method of constituent-based features. The 35 selected dependency-based features include:

- *Predicate/Argument relation type* (d1/d2), *relation path* (d3), *POS pattern of predicate's children* (d4) and *relation pattern of predicate's children* (d5) features from (Hacioglu, 2004).
- *Child relation set* (d6), *child POS set* (d7), *predicate/argument parent word* (d8/d9), *predicate/argument parent POS* (d10/d11), *left/right word* (d12/d13), *left/right POS* (d14/d15), *left/right relation* (d16/d17), *left/right sibling word* (d18/d19), *left/right sibling POS* (d20/d21) and *left/right sibling relation* (d22/d23) features as described in (Johansson and Nugues, 2008).
- *Dep-exists* (d24) and *dep-type* (d25) features from (Boxwell et al., 2009).
- *POS path* (d26), *POS path length* (d27), *REL path length* (d28) from (Che et al., 2008).
- *High/low support verb* (d29/d30), *high/low support noun* (d31/d32) features from (Zhao et al., 2009).
- *LCA's word/POS/relation* (d33/d34/d35) inspired by (Toutanova et al., 2005).

To maintain the consistency between two syntactic views, the dependency parses are generated by a constituent-to-dependency converter (Marneffe et al., 2006), which is suitable for semantic analysis as it retrieves the semantic head rather than the general syntactic head, using a set of modified Bikel's head rules.

4 Selection of Combined Features

The combined features, each of which consists of two different basic features, have proven to be positive for SRL. Several combined features have been widely used in SRL, such as '*predicate+head word*' and '*position+voice*'. But to our knowledge, there is no prior report about the selection method of combined features for SRL. The common entropy-based criteria are invalid here because the combined features always take lots of distinct values. And the greedy method is too complicated to be practical due to the large number of combinations.

In this paper, we define two statistical criteria to efficiently estimate the classification performance of each combined feature on the corpus. Inspired by Fisher Linear Discriminant Analysis (FLDA) (Fisher, 1938) in which the separation of two classes is defined as the ratio of the variance between the classes to the variance within the classes, namely larger ratio can lead to better separation between two classes, and the discriminant plane can be achieved by maximizing the separation. Therefore, in this paper, we adopt the ratio of inter-class distance to intra-class distance to measure to what extent a combined feature can partition the data.

Initially, the feature set contains only the N basic features. We construct one combined feature f_{ab} at each iteration by combining two basic features f_a and f_b, where $a,b \in [1,N]$ and $a \neq b$. We push f_{ab} into the feature set and take it as the $N+1$ *th* feature. Then, all the training instances are represented by feature vectors using the new feature set, and we then quantize the feature vectors of positive and negative data orderly to keep their intrinsic statistical difference. If the training dataset is denoted as $D:\{D_{pos}, D_{neg}\}$, then the separation criterion, namely the ratio of inter-class to intra-class distance for feature f_i can be given as

$$g(f_i) = \frac{InterDist_{f_i}(D_{pos}, D_{neg})}{IntraDist_{f_i}(D_{pos}, D_{neg})} \quad (1)$$

where the inter-class and the intra-class distance between D_{pos} and D_{neg} for feature f_i are specified by (2) and (3), respectively.

$$InterDist_{f_i}(D_{pos}, D_{neg}) = \left(Mean_{f_i}(D_{pos}) - Mean_{f_i}(D_{neg})\right)^2 \quad (2)$$

$$IntraDist_{f_i}(D_{pos}, D_{neg}) = S^2_{f_i}(D_{pos}) + S^2_{f_i}(D_{neg}) \quad (3)$$

$Mean_{f_i}(D)$ in (2) and $S_{f_i}(D)$ in (3) represents the sample mean and the corresponding sample standard deviation of feature f_i in dataset D as given in (4) and (5).

$$Mean_{f_i}(D) = \frac{\sum_{x \in D} x(i)}{|D|}, i \in [1, N+1] \quad (4)$$

$$S_{f_i}(D) = \sqrt{\frac{\sum_{x \in D}\left(Mean_{f_i}(D) - x(i)\right)^2}{N}}, i \in [1, N+1] \quad (5)$$

Essentially, the inter-class distance reflects the distance between the center of positive dataset and the center of negative dataset, and the intra-class distance indicates the intensity of all instances relative to the corresponding center. Therefore, larger ratio will lead to a better partition for a feature, as has been pointed out by FLDA. In order to compare the ratio between different combined features, we further standardize the value of $g(f_i)$ by computing its *z*-score $Z(f_i)$ which indicates how many standard deviations between a sample and its mean, as given in (6).

$$Z(f_i) = \frac{g(f_i) - \overline{g(f_i)}}{S_G} \quad (6)$$

where $\overline{g(f_i)}$ represents the sample mean as given in (7), and S_G represents the sample standard deviation of the sequence $g(f_i)$ where i ranges from *1* to *N+1* as given in (8).

$$\overline{g(f_i)} = \frac{\sum_{i=1}^{N+1} g(f_i)}{N+1}, i \in [1, N+1] \quad (7)$$

$$S_G = \sqrt{\frac{\sum_{i=1}^{N+1}(g(f_i) - \overline{g(f_i)})^2}{N}}, i \in [1, N+1] \quad (8)$$

After figuring out the $Z(f_a)$ and $Z(f_b)$ for the basic feature f_a and f_b, and $Z(f_{ab})$ for the combined feature f_{ab} by (6), we define the other criterion, namely the improvement $I(f_{ab})$ of the combined feature, as the smaller difference between the *z*-score of the combined

feature and its two corresponding basic features as given in (9).

$$I(f_{ab}) = Z(f_{ab}) - \mathrm{Max}\left(Z(f_a),\ Z(f_b)\right) \qquad (9)$$

Finally, the combined feature with a negative $I(f_{ab})$ value is eliminated. Then, we will rank the combined features in terms of their *z*-score, and use the top *N* of them for later classification. The selection method based on the two criteria can effectively filter out combined features whose means have no significant difference between positive and negative data, and hence retain the potentially useful combined features for the separation. Meanwhile, it has a relatively fast speed when dealing with a large number of features in comparison to the greedy method due to its simplicity.

5 Performance Evaluation

5.1 Experimental Setting

In our experiments, we adopt the three-step strategy proposed by (Xue and Palmer, 2004). First, argument candidates are generated from the input constituent parse tree using the prevalent heuristic-based pruning algorithm in (Xue and Palmer, 2004). Then, each predicate-argument pair is converted to a flat feature structure by which the similarity between two instances can be easily measured. Finally we employ the Support Vector Machines (SVM) classifier to identify and classify the arguments. It is noteworthy that we use the same basic features, but different combined features for the identification and classification of arguments. We present the result comparison between using gold-standard parsing and automatic parsing, and also offer an analysis of the contribution of the combined features.

To evaluate the proposed method and compare it with others, we use the most commonly used corpus in Chinese SRL, Chinese Proposition Bank (CPB) version 1.0, as the dataset. The CPB corpus contains 760 documents, 10,364 sentences, 37,183 target predicates and 88,134 arguments. In this paper, we focus on six main types of semantic roles: Arg0, Arg1, Arg2, ArgM-ADV, ArgM-LOC and ArgM-TMP. The number of semantic roles of the six types accounted for 95% of all the semantic roles in CPB. For SRC, we use the one-versus-all approach, in which six SVMs will be trained to separate each semantic type from the remaining types. We divide the corpus into three parts: the first 99 documents (chtb_001.fid to chtb_099.fid) serve as the test data, the last 32 documents (chtb_900.fid to chtb_931.fid) serve as the development data and the left 629 documents (chtb_100.fid to chtb_899.fid) serve as the training data.

We use the SVM-Light Toolkit version 6.02 (Joachims, 1999) for the implementation of SVM, and use the Stanford Parser version 1.6 (Levy and Manning, 2003) as the constituent parser and the constituent-to-dependency converter. In classifications, we employ the linear kernel for SVM and set the regularization parameter to the default value which is the reciprocal of the average Euclidean norm of training data. The performance metrics are: accuracy (A), precision (P), recall (R) and *F*-score (F).

5.2 Combined Feature Selection

First, we select the combined features for classifications of SRI and SRC using the method described in Section 4 on the training data with gold-standard parse trees. Due to the limit of this paper, we only list the top-10 combined features for SRI and SRC for the 6 different types, as shown in Table 1 in which each combined feature is expressed by the IDs of its two basic features with a plus sign between them.

Rank	SRI	ARG0	ARG1	ARG2	ADV	LOC	TMP
1	c1+c6	c1+c6	c1+c6	c1+c6	c1+c6	c5+c27	c1+c6
2	c1+d3	c32+c30	c30+d31	c1+d1	c30+d27	c9+d17	c22+c27
3	d25+d14	c7+c6	c30+d32	c1+c7	c30+d28	c9+d13	c7+c6
4	c4+d25	c1+c2	c5+c30	c7+c6	c1+c11	c9+c2	d26+d27
5	d25+d22	c1+c12	c30+d24	c1+c5	c24+d33	c23+c27	d26+d28
6	d25+d20	c23+c6	c30+c21	c1+c23	c30+d25	c9+c20	c23+d26
7	d25+d21	c1+c3	c5+c4	c23+c6	c24+d9	c14+c32	c5+d26
8	d25+d18	c10+d35	c1+c10	c1+c3	c27+c2	c14+c10	d26+d31
9	d25+d19	c10+d1	c30+d10	c5+c6	c22+c2	c9+c26	d26+d32
10	d25+d35	c10+d28	c4+c6	c1+d5	c24+d13	c14+c2	c23+c6

Table 1. Top-10 combined features for SRI and SRC ranked by *z*-score

Table 1 shows that the commonly used combined features, such as '*predicate+head word*' (c1+c6) and '*position+voice*' (c4+c5) proposed by (Xue and Palmer, 2004) are also included. In particular, the '*predicate+head word*' feature takes first place in all semantic

categories except LOC, in which the combination of the new feature '*locational cue words*' (c27) and the '*voice* (c5)' feature performs the best. The results also show that the most frequently occurred basic features in the combined set are '*predicate*' (c1), '*head word*' (c6), '*type of LCA and left child*' (c30), '*dep-type*' (d25) and '*POS path*' (d26). These basic features should be more discriminative when combined with others. Additionally, we find some other latent effective combined features, such as '*predicate subcategorization+head word*' (c7+c6), '*predicate POS+head word*' (c23+c6) and '*predicate+phrase type*' (c1+c3), whose performance will be further validated and analyzed later in this section. It is obvious that the obtained combined features for SRI and SRC are different, and the obtained combined features for each type are also different as our selection method is based on positive and negative data which are completely different for each argument type. In SRI phase, we will use the combined features for all the six semantic types (after removing duplicates).

Then, we evaluate the performance of SRL based on the top-*N* combined features. The preliminary evaluation on the development set suggests that the performance becomes stable when *N* exceeds 20. Therefore, we vary the value of *N* to 5, 10 and 20 in the experiments to evaluate the performance of combined features. Corresponding to the three different values of *N*, we finally obtained 28, 60 and 114 combined features for the SRL, respectively.

5.3 SRL Using Gold Parses

To illustrate each component of the method, we constructed 6 SRL systems using 6 different feature sets: 'Constituent Only' (CO) – uses the constituent-based features, as presented in Section 3.1; 'Dependency Only' (DO) – uses the dependency-based features, as presented in Section 3.2; 'CD' – uses both the constituent-based features and the dependency-based features, but no combined features; 'CD+Top5' – obtained by adding the top-5 combined features to the 'CD' system; and similarly for the 'CD+Top10' and the 'CD+Top20' systems. And 'CO' serves as the baseline in our experiments.

First, we evaluate the performance of SRI using the held-out test set with gold-standard constituent parse trees. The corresponding dependency parse trees are automatically generated by the constituent-to-dependency converter included in the Stanford Parser. The testing results of the six systems on the SRI phase are shown in Table 2.

System	A (%)	P (%)	R (%)	F (%)
CO	97.87	97.04	97.30	97.17
DO	92.76	92.90	84.19	88.33
CD	97.98	97.44	97.25	97.34
CD+Top5	98.12	97.56	97.58	97.57
CD+Top10	98.15	97.61	97.62	97.61
CD+Top20	98.18	97.68	97.64	97.66

Table 2. Results of SRI using gold parses

It can be seen from Table 2 that 'CD' and 'CD+Top20' give only slightly improvement over 'CO' by less than 1% point. In other words, feature combinations do not seem to be very effective for SRI. Then we label all recognized constituents in the SRI phase with one of the six semantic role types. Table 3 displays the *F*-score of each semantic type and the overall SRC on the test set with gold-standard parses.

System	Arg0	Arg1	Arg2	ADV	LOC	TMP	ALL
CO	92.40	90.57	59.98	96.25	86.80	98.14	91.23
DO	90.70	88.22	56.95	94.54	81.23	97.37	89.14
CD	92.85	91.29	63.35	96.55	87.55	98.32	91.86
CD+Top5	93.96	92.79	73.48	97.13	88.63	98.31	93.22*[1]
CD+Top10	94.15	93.23	74.18	97.42	87.17	98.57	93.41*
CD+Top20	94.10	93.19	75.13	97.23	88.05	98.48	93.46*

Table 3. Results of SRC using gold parses

Table 3 shows that the proposed method performs much better in SRC. It improves the constituent-based method by more than 2% in SRC. The effectiveness of combined features can also be clearly seen because the overall *F*-scores of the three systems using combined features all exceed 93%, significant greater than the systems using singular features. The improvement is noticeable for all semantic role types except the 'TMP' type. It means that the dependency parses cannot provide additional information to the labeling of this type. The results of Table 2 and Table 3 together show

[1] The F-score value with an asterisk (*) indicates that there is a statistically significant difference between this system and the baseline ('CO') using the chi-square test ($p<0.05$).

that our method using combined features can effectively improve the performance of SRL on the SRC phases, when using gold parses.

5.4 SRL Using Automatic Parses

To measure the performance of the algorithm in practical conditions, we replicate the above experiments using Stanford Parser on the raw texts of the test set, without segmentation or POS tagging. The dependency parses are also generated from the automatic constituent parses, as described in Section 5.3. The results are shown in Table 4.

System	A (%)	P (%)	R (%)	F (%)
CO	71.54	68.72	70.62	69.66
DO	68.86	65.06	60.68	62.79
CD	73.53	70.63	72.75	71.67*
CD+Top5	73.62	70.69	72.98	71.82*
CD+Top10	73.65	70.71	73.08	71.88*
CD+Top20	73.67	70.70	73.16	71.91*

Table 4. Results of SRI using automatic parses

Table 4 shows that the proposed method is also effective when using automatic parses despite the dramatic decrease in *F*-scores in comparison to using gold-standard parses. The decline is mainly caused by the heuristic-based pruning strategy in which a number of real arguments are pruned when using the constituent parses with errors. Further analysis shows that, in SRI using gold parses, the ratio of incorrectly pruned arguments to the total is less than 2%, but the ratio jumps to 17% when using automatic parses. Next, on the basis of the SRI results, we test the performance of SRC using the automatic parses, as shown in Table 5.

System	Arg0	Arg1	Arg2	ADV	LOC	TMP	ALL
CO	89.20	88.90	54.47	93.93	81.80	94.38	88.24
DO	88.79	89.32	50.21	91.27	78.26	93.86	87.63
CD	89.75	89.87	57.71	95.28	84.22	94.71	89.16*
CD+Top5	90.75	90.97	65.64	95.53	84.45	94.45	90.16*
CD+Top10	90.96	91.37	67.25	95.31	84.49	94.61	90.45*
CD+Top20	90.94	91.29	67.42	95.22	84.39	94.65	90.42*

Table 5. Results of SRC using auto parses

Table 5 shows only a slight decline in comparison with the result of using gold-standard parses, and it maintains the same trend of performance for each semantic role in the Table 3, which proves the validity of the proposed method when using automatic parses. Table 6 shows the *F*-score of the overall SRL on both the gold-standard and the automatic parse data.

System	Gold Parse (F%)	Auto Parse (F%)
CO	89.29	63.13
DO	82.69	60.34
CD	90.01	65.56*
CD+Top5	91.47*	66.37*
CD+Top10	91.68*	66.61*
CD+Top20	91.76*	66.61*

Table 6. Results of overall SRL

Table 6 shows that the *F*-score of the 'CD+Top20' surpasses that of the 'CO' system by more than 2% on the gold parses, and more than 3% on the automatic parse. In other words, the method using constituent and dependency syntactic views performs even more effective for the automatic parses. The last three rows of Table 6 shows that the top-10 combined features perform better than the top-5 features by adding 32 more features, but the top-20 combined features obtain similar results to the top-10 features by adding 54 more features. It suggests that only several salient combined features can actually improve the performance.

5.5 Combined Feature Performance

To evaluate the performance of each combined feature to identify the salient combined features for SRL, we rank the 60 combined features used by the 'CD+Top10' system on the test data with gold-standard parses, according to the *F*-score improvement achieved by each combined feature. Here we list the top 20 of them which are shown in Table 7.

Rank	Feature	Δ F(%)	Rank	Feature	Δ F(%)
1	c1+c6	0.611	11	c10+d1	0.413
2	c1+c10	0.593	12	c5+d26	0.404
3	c4+c6	0.557	13	c24+d9	0.395
4	c9+c20	0.503	14	d25+d35	0.395
5	c23+c6	0.494	15	c30+d24	0.377
6	c1+c3	0.458	16	c9+c26	0.377
7	c9+d13	0.449	17	c10+d28	0.368
8	c14+c10	0.431	18	c30+d29	0.365
9	c1+c5	0.422	19	c30+d30	0.361
10	c24+d33	0.413	20	c7+c6	0.361

Table 7. Top-20 combined features

As can be seen from Table 7, a half of combined features are composed by constituent

features only, and the other half contain at least one dependency-based feature. This indicates that dependency features can be helpful to construct combined features for SRL. Through analyzing the performance of each combined features, we have obtained some new and effective combined features which were not recognized before, such as '*predicate+partial path*' (c1+c10), '*position+head word*' (c4+c6), '*Head word POS+right sibling POS*' (c9+c20). Observation from these combined features suggests that not all combined features are composed by two significant basic features. Some not significant ones, such as '*partial path*' (c10) and '*Head word POS*' (c9) can also produce salient combined features.

Furthermore, we find that the relative order of the combined features in Table 7 is not exactly consistent with their orders in Table 1. The inconsistency indicates that the estimation criteria used for combined features selection is not perfect. In estimation, the effect of combined features is evaluated simply based on the distance between the positive and the negative dataset by considering the efficiency. But in practice, the effects of them are determined through one-by-one classification.

5.6 Comparison to Other Work

Finally, we compare the proposed method with other four representative Chinese SRL systems. First, the 'Xue[1]' system (Xue and Palmer, 2005) is a typical feature-based system using 9 basic features, 2 combined features and the Maximum Entropy (ME) classifier. Second, the 'Liu' system (Liu et al. 2007) which uses 19 basic features, 10 combined features and also the ME classifier. Third, the 'Che' (Che, 2008) system use a hybrid convolution tree kernel to directly measure the similarity between two constituent structures. Fourth, the 'Xue[2]' system described in (Xue, 2008), which is similar to 'Xue[1]' on basic framework, but using a new feature set. The 'Xue[2]' system evaluates the SRL of the verbal predicates and the nominalized predicates separately, and offers no consolidated evaluation in (Xue, 2008). So in the comparison, we refer to its performance on the verbal predicates and the nominalized predicates as 'Xue[21]' and 'Xue[22]'.

All the four systems mentioned above use the constituent as the labeling unit and use the CPB corpus as the data set, the same as our method. And we use the same training and test data splits as in the 'Xue[1]' and 'Che' systems. Table 8 shows the comparison results in terms of *F*-score on both gold parses and auto parses.

System	Gold Parse (F%)	Auto Parse (F%)
Xue[22]	69.6	57.3
Xue[1]	91.3	61.3
Liu	91.31	—
Che	91.67	65.42
Ours	91.76	66.61
Xue[21]	92.0	66.8

Table 8. Comparison to other work

Table 8 shows that our method performs better than the 'Xue[1]', 'Liu' and 'Che' systems on both gold parses and automatic parses. It is only slightly worse than the 'Xue[21]', namely the verbal predicates part of the 'Xue[2]' system. But for the other part of the 'Xue[2]' system for the nominalized predicates, namely the 'Xue[22]', our method performs much better than it. The results further verify the validity of the method.

6 Conclusions

This paper presents a novel feature-based SRL approach for Chinese. Compared to the traditional feature-based methods, the method can effectively integrate the constituent and the dependency syntactic views at the feature level. The method provides an effective way to connect two syntactic views by a statistical selection method of combined features to substantially improve the feature-based SRL method. The complexity of the method will not increase significantly compared to the method using one syntactic view as we use a constituent-to-dependency conversion rather than additional dependency parsing. The effectiveness of the method has been proven by the experiments on CPB using SVM classifier with linear kernel.

Acknowledgments

This work is supported by the Key Program of National Natural Science Foundation of China under Grant No. 60736014, the Key Project of the National High Technology Research and Development Program of China under Grant No. 2006AA010108, and the Hong Kong Polytechnic University under Grant No. G-U297 and G-U596.

References

Collins F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The Berkeley FrameNet Project. *Proceedings of Coling-ACL-1998.*

Stephen A. Boxwell, Dennis Mehay, and Chris Brew. 2009. Brutus: A Semantic Role Labeling System Incorporating CCG, CFG, and Dependency Features. *Proceedings of ACL-2009.*

Wanxiang Che. 2008. *Kernel-based Semantic Role Labeling.* Ph.D. Thesis. Harbin Institute of Technology, Harbin, China.

John Chen and Owen Rambow. 2003. Use of Deep Linguistic Features for the Recognition and Labeling of Semantic Arguments. *Proceedings of EMNLP-2003.*

Weiwei Ding and Baobao Chang. 2008. Improving Chinese Semantic Role Classification with Hierarchical Feature Selection Strategy. *Proceedings of EMNLP-2008.*

Ronald A. Fisher. 1938. The Statistical Utilization of Multiple Measurements. *Annals of Eugenics*, 8:376-386.

Daniel Gildea and Daniel Jurafsky. 2002. Automatic Labeling of Semantic Roles. *Computational Linguistics*, 28(3):245-288.

Daniel Gildea and Martha Palmer. 2002. The Necessity of Syntactic Parsing for Predicate Argument Recognition. *Proceedings of ACL-2002.*

Kadri Hacioglu. 2004. Semantic Role Labeling Using Dependency Trees. *Proceedings of COLING-2004.*

Jan Hajic, Massimiliano Ciaramita, Richard Johansson, et al. The CoNLL-2009 Shared Task: Syntactic and Semantic Dependencies in Multiple Languages. *Proceedings of CoNLL-2009.*

Thorsten Joachims. 1999. Making large-Scale SVM Learning Practical. Advances in Kernel Methods. Support Vector Learning, B. Schölkopf and C. Burges and A. Smola (ed), MIT Press.

Richard Johansson and Pierre Nugues. 2008. Dependency-based Semantic Role Labeling of PropBank. *Proceedings of EMNLP-2008.*

Roger Levy and Christopher D. Manning. 2003. Is it harder to parse Chinese, or the Chinese Treebank. *Proceedings of ACL-2003.*

Huaijun Liu, Wanxiang Che, and Ting Liu. 2007. Feature Engineering for Chinese Semantic Role Labeling. *Journal of Chinese Information Processing*, 21(2):79-85.

Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. 2006. Generating Typed Dependency Parses from Phrase Structure Parses. *Proceedings of LREC-2006.*

Truc-Vien T. Nguyen, Alessandro Moschitti, and Giuseppe Riccardi. 2009. Convolution Kernels on Constituent, Dependency and Sequential Structures for Relation Extraction. *Proceedings of EMNLP-2009.*

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The proposition Bank: An Annotated Corpus of Semantic Roles. *Computational Linguistics*, 31(1):71-106

Sameer Pradhan, Wayne Waed, Kadri Haciolgu, and James H. Martin. 2004. Shallow Semantic Parsing using Support Vector Machines. *Proceedings of HLT/NAACL-2004*

Sameer Pradhan, Wayne Ward, Kadri Hacioglu, James H. Martin, and Daniel Jurafsky. 2005. Semantic Role Labeling Using Different Syntactic Views. *Proceedings of ACL-2005.*

Sameer Pradhan, Wayne Ward, and James H. Martin. 2008. Towards Robust Semantic Role Labeling. *Computational Linguistics*, 34(2): 289-310.

Vasin Punyakanok, Dan Roth, Wentau Yih. 2005. The Necessity of Syntactic Parsing for Semantic Role Labeling. *Proceedings of IJCAI-2005*.

Mihai Surdeanu, Lluis Marquez, Xavier Carreras, and Pere R. Comas. 2007. Combination Strategies for Semantic Role Labeling. *Journal of Artificial Intelligence Research*, 29:105-151.

Kristina Toutanova, Aria Haghighi, and Christopher D. Manning. 2005. Joint learning improves semantic role labeling. *Proceedings of ACL-2005.*

Nianwen Xue and Martha Palmer. 2004. Calibrating Features for Semantic Role Labeling. *Proceedings of EMNLP-2004.*

Nianwen Xue and Martha Palmer. 2005 Automatic semantic role labeling for Chinese verbs. *Proceedings of IJCAI-2005*.

Nianwen Xue. 2008. Labeling Chinese Predicates with Semantic Roles. *Computational Linguistics*, 34(2):225-255.

Hai Zhao, Wenliang Chen, and Chunyu Kit. 2009. Semantic Dependency Parsing of NomBank and PropBank: An Efficient Integrated Approach via a Large-scale Feature Selection. *Proceedings of EMNLP-2009.*

Chinese Frame Identification using T-CRF Model

Ru Li[*], Haijing Liu[+], Shuanghong Li[#]
School of Computer and Information Technology
Shanxi University
[*]liru@sxu.edu.cn
[+]bukaohuaxue@163.com
[#]lishuanghong09@gmail.com

Abstract

As one of the important tasks of SemEval Evaluation, Frame Semantic Structure Extraction based on the FrameNet has received much more attention in NLP field. This task is often divided into three sub-tasks: recognizing target words which are word expressions that evoke semantic frames, assigning the correct frame to them, namely, Frame Identification (FI), and for each target word, detecting and labeling the corresponding frame elements properly. Frame identification is the foundation of this task. Since the existence of links between frame semantics and syntactic features, we attempt to study FI on the basis of dependency syntax. Therefore, we adopt a tree-structured conditional random field (T-CRF) model to solve Chinese frame identification based on Dependency Parsing. 7 typical lexical units which belong to more than one frame in Chinese FrameNet were selected to be researched. 940 human annotated sentences serve as the training data, and evaluation on 128 test data achieved 81.46% precision. Compared with previous works, our result shows obvious improvement.

1 Introduction

In recent years, semantic research has roused great interest in NLP field. With the progress of many semantic lexicons, this research gradually becomes promising and exciting. As one of the tasks of SemEval Evaluation, Frame Semantic Structure Extraction based on the FrameNet grows to be highlighted for special attention.

Given a sentence, the task of Frame Semantic Structure Extraction consists of the following three parts: recognizing the word expressions (target words) that evoke semantic frames; discriminating the word sense (frame) of each evoking expression; for each target word, labeling its syntactic dependents with regard to which roles in that frame they fill (Baker et al., 2006). Among of these three components, frame identification is the fundamental and key problem. However, current research of this task in Chinese is only focused on semantic role labeling based on the given target words and their corresponding frames (Xue, 2008). We insist that whether target words can be assigned correct frames in context is a crucial problem demanding prompt solution in this task.

Chinese FrameNet (CFN) (You and Liu, 2005), developed by Shanxi University, is an ongoing effort of building a semantic lexicon for Chinese based on the theory of Frame Semantics (Fillmore, 1982), referencing the FrameNet(Baker et al., 1998) and supported by corpus evidence. The CFN project currently contains more than 2100 lexical units, more than 300 semantic frames, and has exemplified more than 21600 annotated sentences. The ultimate goal of this project is to generate information about the articulation of the semantic and syntactic requirements of Chinese lexical items and presents this information in a variety of web-based reports and represents the lexical semantics of all the sentences in a Chinese text.

Coling 2008: Poster Volume, pages 674–682,
Beijing, August 2010

According to statistics, there are 332 lexical units belonging to more than one frame in the current CFN databases. For example, lexical unit “表示”can evoke the following three frames: “表达 (Expressing_publicly) ”, “陈述 (Statement) ” and “代表(representative) ”. In order to extract the semantic structure of a sentence containing ambiguous target words, the first step is to assign the correct frame to the target words in a given context.

This task is similar with the word sense disambiguation (WSD) task to a certain extent (Katrin Erk, 2005). WSD is to resolve the inherent polysemia of words by determining the appropriate sense for each ambiguous word in a given text, while frame identification is assigning a correct frame for the ambiguous target word in the current sentence context. Nevertheless, essential difference exists between them. WSD prefers to disambiguation on static sense, whereas based on the frame semantics, frame identification lays particular emphasis on consistency between sentence scene and the dynamic scene described by the candidate frames.

Since the existence of links between frame semantics and syntactic features, we adopt a tree-structured conditional random field (T-CRF) model to solve Chinese frame identification based on Dependency Parsing. 7 typical lexical units which belong to more than one frame in CFN were selected to be researched. 940 human annotated sentences were collected for the training data, and 128 for test data.

The rest of this paper is organized as follows. Section 2 introduces some related work. Section 3 gives a simple system description. Section 4 describes Chinese frame identification using T-CRF model. Section 5 presents our experimental results and some analysis. Section 6 is the conclusions.

2 Related Work

With the development and improvement of FrameNet, the research based on this lexical resource is increasing gradually. Frame Semantic Structure Extraction based on FrameNet is such hot topics. One sub-tasks of this research is frame identification, which is the research problem in this paper.

At present, there are some but not much work on frame identification. Main works are as follows: CL Research participated in the SemEval-2007 task for Frame Semantic Structure Extraction. They integrated the use of FrameNet in the Text Parser component of the CL Research KMS. In particular, they created a FrameNet dictionary from the FrameNet databases with the CL Research DIMAP dictionary software and used this dictionary as a lexical resource. The current FrameNet DIMAP dictionary contains 7575 entries, with many entries having multiple senses. For each sense, the FrameNet part of speech, the definition, the frame name, the ID number, and the definition source (identified as FN or COD) are captured from the FrameNet files. When a lexical unit is recognized in processing the text, the first step is to retrieve the entry for that item in the dictionary and use the frame element realization patterns to disambiguate among the senses. A score is computed for each sense and the score with the highest sense was selected. They evaluated on three texts and the best result is 66.10% precision (Litkowski, 2007).

Adrian Bejan and Hathaway (2007) selected from the FN lexicon 556 target words that evoke at least two semantic frames and have at least five sentences annotated for each frame. And then they assembled a multi-class classifier using two types of models: SVM and Maximum Entropy for each ambiguous target word. They extracted features used in word sense disambiguation (Florain et al., 2002), lexical features of the target word, and NAMED ENTITY FLAGS associated with the root vertex in a syntactic parse tree. For the rest of the ambiguous target words that have less than five sentences annotated, they randomly chose a frame as being the correct frame in a given context. For FI sub-task, they obtained 76.71% accuracy compared to a baseline of 60.72% accuracy that always predicts the most annotated frame for each of the 556 target words.

Johansson and Nugues (2007) firstly used some filtering rules to detect target words, and for the target words left after the filtering, they trained a disambiguating SVM classifier on all ambiguous words listed in FrameNet. The classifier used the following features: target lemma, target word, sub categorization frame, the set of dependencies of the target, the set of words of the child vertexes, and the parent word of the target. Its accuracy was 84% on the ambiguous

words, compared to a first-sense baseline score of 74%.

The above researches focused on English based on FrameNet. To our knowledge, there exists no work for Chinese by far. Most methods mentioned above treat the frame identification as an independent classification problem for each ambiguous target word in a sentence. However, because of neglecting the relations between the candidate frames, the resulting frame assignment may be semantically inconsistent over the sentence.

3 System Description

Our system consists of three stages. The first is corpus construction of our experiments. We selected 7 typical lexical units from the current CFN lexicon which can evoke at least two semantic frames. They are "表示","想","有","叫","倒","下降","装载", respectively. For each of them, we collected sentences containing this word from Sogou Corpus and CCL Contemporary Chinese Corpus of Beijing University. Through a series of refining, 940 sentences annotated correct frame for each target word comprise a standard corpus as the training data. Another 128 sentences serve as the test data.

The second stage is dependency parsing. We used LTP of Information Retrieval Research Center, Harbin Institute of Technology (HIT-CIR) to POS tagging and dependency parsing the training and test sentences. For the obvious lexical and syntax errors in the outputs, manually corrected was conducted.

At last, Chinese frame identification task is regarded as a labeling task on the dependency tree structure. By using T-CRF, we can model this as the maximization of the probability of word sense (frame) trees, given the scores for vertexes and edges. In the training phase, appropriate features of vertex and edge are extracted, and the weight vectors are optimized over the training data.

Figure 1 gives an illustration of the system.

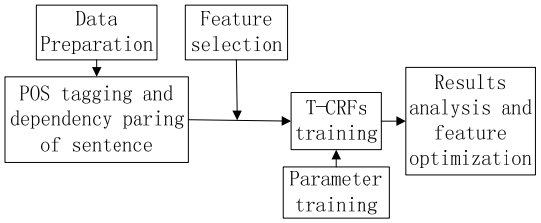


Figure 1. Framework of the system

4 Chinese Frame Identification

Given a sentence, frame identification is to determine an appropriate frame for each of target words by comparing consistency between sentence context and the dynamic scene described by their candidate frames. Currently, most researchers addressed this task as an independent classification problem for each target word in a sentence. Consequently, the resulting frame assignment for each target word may be semantically inconsistent over the sentence.

We regard Chinese frame identification problem as a labeling task on the dependency tree structure due to the links between syntactic features and frame semantics. Our empirical study shows that the frame of target word not only influenced by the adjacent words in position but also its governor and dependents words in syntactic structure. Therefore, we try to solve this problem based on dependency parsing. T-CRF model is a special CRF model, which is different from widely used linear-chain CRFs, in which the random variables are organized in a tree structure. As we can see, it should be feasible and reasonable to adopt a T-CRF model to frame identification after parsing the sentence.

In this section, we firstly introduce the linear-chain CRFs briefly, and then explain the T-CRF model for Chinese frame identification, especially the feature selection and parameter estimation.

4.1 Tree-Structured Conditional Random Field（T-CRF）

Conditional Random Fields (CRFs) are undirected graphical models (Lafferty et al, 2001). For the observation sequence $X = x_1 x_2 x_3 \cdots x_n$ and its corresponding label sequence $Y = y_1 y_2 y_3 \cdots y_n$, CRF defines the conditional probability as:

$$P(Y \mid X)$$
$$= \frac{1}{Z(X)} \{ \exp(\sum_i \sum_k \lambda_k f_k(y_{i-1}, y_i, X)) + \exp(\sum_i \sum_k \mu_k g_k(y_i, X)) \}$$

where X is the observation sequence, and y_i is the label at position i in label sequence Y. $f_k(\cdot)$ and $g_k(\cdot)$ are feature functions. λ_k and μ_k are the weight vectors. $Z(X)$ is the normalization factor. CRFs are state-of-the-art methods for sequence labeling problem in many NLP tasks.

Tree-Structured Conditional Random Field (Tang et al., 2006) is a particular case of CRFs, which can model dependencies across hierarchically laid-out information, such as dependency syntactic relations between words in a sentence.

The graphical structure of T-CRF is a tree, in which three main relations exist for a vertex: parent-child, child-parent and sibling vertexes. In our experiments, we only used parent-child edges and child-parent edges. The sibling-vertexes edges were ignored because of weak dependency syntactic relation between words in a sentence. So the probability distribution in our T-CRF model can be written as below.

$$p(y \mid x) = \frac{1}{Z(x)} \exp \sum_{v \in V} \{F + G + S\}$$
$$F = \sum_j \lambda_j f_j(v, y(v), x)$$
$$G = \sum_k \mu_k g_k(v, y(v), x, v', y(v'))$$
$$S = \sum_l \sigma_l s_l(v, y(v), x, v'', y(v''))$$

where F 、 G 、 S represent the feature functions of current vertex, feature functions of parent vertex of current vertex and feature functions of child vertexes of current vertex, respectively. v is a word corresponding to the vertex in the tree, v' is the parent vertex of v and v'' are the child vertexes of v.

In Chinese frame identification, the observation *x* in T-CRF corresponds to a word in the current sentence. The label *y* thus corresponds to the frame name for the word. In the experimental corpus, for the target word, y is annotated its correct frame name, while for the other words left, y is annotated tag "null". These target words are the 7 lexical units we selected and their frames come from the current CFN lexicon. At present, only the frame identification of target word was studied, the disambiguation of the other multi-senses words in the sentence was not being processed.

Although T-CRFs are relatively new models, they have already been applied to several NLP tasks, such as semantic role labeling, semantic annotation, word sense disambiguation, image modeling.(Cohn and Blunsom, 2005; Tang et al., 2006; Jun et al., 2009; Awasthi et al., 2007). All these works proved this model to be useful in modeling the semantic structure in a sentence or a text. Our study is the first application of T-CRFs to frame identification.

4.2 Feature Selection

In order to apply T-CRF model, it is necessary to represent the sentence with a hierarchical structure. We used LTP of HIT-CIR to POS tagging and dependency parsing the training and test sentences. To facilitate the description of feature selection based on the dependency tree structure, figure 2 gives the dependency output of an example.

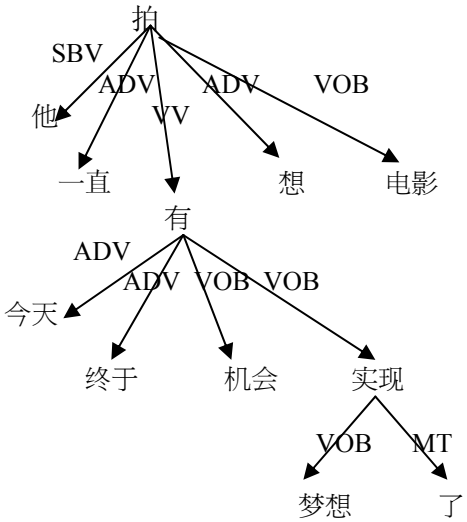


Figure 2. Example of a dependency parsed sentence.

This example sentence is:"他一直想拍电影，今天终于有机会实现梦想了". In English, it reads "He has been want to make films, and finally has the opportunity to realize his dream

today”. In the dependency tree structure, arrow points from the parent vertex to child vertex, the label on a arc is the type of dependency relation between the parent and the child vertex.

Feature selection is a core problem in sequence labeling model. In our experiments, 18 template settings were conducted to discover the best features for frame identification. During this process, we considered two main factors: firstly, the number of features should not be too large so as to avoid the over-fitting phenomenon; secondly, the selected features should be able to provide enough information conditioned on tolerated computation, for the purpose of improving the performance of system. With the increasing of the number of features and the cost of the system, if the performance of system can not be improved obviously, we stopped to add features and regard the parameter of current template as the best. At this moment, a good balance between the performance and cost of computation was achieved.

We experimented with two different types of feature settings. One we used was the very basic feature sets based on the words and Part of Speech (POS) and their bigram features. In order to see the effectiveness of dependency features, the other type of feature settings include more informative tree features. These features capture information about a vertex's parent, its children and the relation with its parent and children. These features are semantically and structurally very informative and we expect to improve our performance with them. The base and tree features we used are listed in table 1.

In these features, the setting of basic features is fundamental and meaningful because it can be used to compare T-CRF and linear chain CRF. For the tree features, given the i-th vertex in the observation x_i, $f(y_p, y_c)$ and $f(y_c, y_p)$ represent whether the current vertex has a parent-child dependency with a parent vertex and whether it has a parent-child dependency with a child vertex, respectively. In dependency grammars (Igor' A. Melchuk, 1988), every vertex has only one parent as its governor, and may have more than one child as its dependents. Words in a sentence through certain syntactic relations form the semantic structure of this sentence. Therefore, we argue that the words that have syntactic dependency relations with the target word are more important than the ones neighboring with it in position for frame identification. For this reason, we added the parent vertex and children vertexes into the tree features. With respective to the relation type, we used the annotation sets defined by HIT-CIR in LTP, which contain 24 kinds of dependency relation types. One thing should be concerned is that we don't consider all types of children vertexes. This is because that according to our empirical study, not all of the children have strong dependencies with the target word. On the contrary, more features would bring the noise and affect the efficiency seriously. Hence, we chose 4 types of children relation from the linguistic point of view. They are, "SBV(subject-verb)" representing "主谓关系", "VOB(verb-object)" representing "动宾关系", "ADV(adverbial)" representing "状中结构" and "ATT(attribute)" representing "定中关系". From the point of grammars and semantics, these four relations are more influenced on the words in a sentence. As we know, the subject, predicate and object constitute the semantic core of a sentence. The good news is that experimental results proved this hypothesis relatively correct.

Category	Features	
Base features	Word and bigram of word, POS and bigram of POS	
Tree features	$f(y_p, y_c)$	Parent vertex of current word
		The edge between current word and its parent
		The dependency relation type between current word and its parent
	$f(y_c, y_p)$	child vertex of current word
		The edge between current word and its child
		The dependency relation type between current word and its child

Table 1. Base Features & Tree Features

4.3 Parameter Estimation

The parameter estimation is to optimize the parameters $\theta = \{\lambda 1, \lambda 2, ...; \mu 1, \mu 2, ...\}$ from training data $D\{(x1, y1), (x2, y2), ...\}$ with empirical distribution $p(x, y)$. Nowadays, the commonly used method for parameter estimation is maximum likelihood function. That is $L_\theta = \arg\max \sum_i \log(p_\theta(y_i / x_i))$ given the observation sequences $\{x_1, x_2, ...\}$ and label sequences $\{y_1, y_2, ...\}$.

In this paper, the conventional L-BFGS method was used to estimate the optimal parameters $\theta = \{\lambda 1, \lambda 2, ...; \mu 1, \mu 2, ...\}$ (Jorge Nocedal and Stephen J. Wright. 1999).

5 Experiments

5.1 Data preparation

So far, there has no research on Chinese frame identification, thus it is unfeasible to do experiments based on readily available corpus. Accordingly, preparing a good and reasonable training and test data is our fundamental task.

At present, there are 332 lexical units that can evoke at least two frames in the CFN lexicon. In this paper, we selected 7 typical ambiguous lexical units to be researched. They are "表示","想","有","叫","倒","下降","装载". The selection principle is following: first of all, it is time-consuming to construct corpus for all of the 332 lexical units, so currently we just studied part of them to prove the validity of the method we proposed. Secondly, the frames evoked by these lexical units should be distinguished clearly by human annotators. For example, lexical unit "高兴" can evoke these three frames: "心理刺激(Experiencer_obj)", "情感体验 (Experiencer_subj)" and "情感反应 (Emotion_directed)". All these frames describe a tender feeling in psychology, so it is difficult to discriminate among them and thus hard to annotate sentences correctly. Thirdly, these 7 lexical units are high frequency words so it is easier to collect sentences and make the experiments more practical.

For each of 7 lexical units, we collected sentences containing this word from Sogou Corpus and Contemporary Chinese Corpus of Beijing University. After a preliminary screening, about 1000 sentences compose the original and coarse corpus.

Although these sentences were complete and relatively standard, some of them didn't meet the criterion of Chinese frame identification research. Such cases mainly include three aspects. For one thing, the correct frame of ambiguous target word is difficult to decide by human annotator. For the other, the meaning of target word can't correspond to any frame definition in current CFN version. For example, lexical unit "想" can express the meaning of opinion and wish which have the corresponding frames in CFN, while the meaning of thinking and memory did not. Lastly, some words couldn't evoke frames though their word forms are the same as lexical unit. We removed the sentences belonging to the above situations and got a refined corpus containing 940 sentences for training data and 128 for test data. And then, we used LTP to POS tagging and dependency parsing the training and test sentences.

5.2 Experimental Results and Analysis

For the linear-chain CRF, we defined the features based on the words, POS of words and their bigram features as the base features. For T-CRF, we used the base features and tree features. Six different types of template settings on these features are listed in table 2.

template	features
T1	Base features
T2	Add edge between current word and its parent on T1
T3	Add dependency type between current word and its parent on T2
T4	Add edge between current word and its four types children vertexes on T1
T5	Add dependency type between current word and its four types children vertexes on T4
T6	Add all these tree features on T1

Table 2. Template settings on different features

For each of these template settings, we experimented on different observation window size of 1, 2 and 3, which represents one word,

two words and three words previous and next to the current word respectively.

We use the $precision = \frac{n}{s}$ to evaluate our system, where n is the number of target words labeled correctly, and s is the total number of target words need to be labeled. In our 128 test sentences, there are 151 target words because there are some sentences containing more than one ambiguous target word. Experimental results on 18 templates are listed in table 3.

From the table 3, we can get four conclusions. Firstly, the best performance 81.46% in T-CRF model increases about 5% over the best performance 76.82% in CRF model. This suggests the dependencies on the tree structure can capture more important characteristics than those on the linear chains do. Secondly, when we added the edge feature between current word and its parent, the performance declined unexpectedly. This can be explained in linguistics: in a dependency parsed sentence, the clique of a governor and its dependents forms "a small world" which can express partial meaning of the sentence, while the parent of current vertex (except the root vertex which has no parent) can not influence much on it because its parent has its own clique, and current word is just a tiny fragment of the clique of its parent, on the contrary, the parent vertex feature will bring negative effect on the current word. For example, the target word "有" in figure 2 can illustrate this case clearly. Thirdly, when we added the children vertexes, the performance increased, that is because current word and its dependents together can form a semantic clique of the sentence. Lastly, when we added the dependency relation type on the features of parent-child edge and child-parent edge, the performance improved slightly because the relation type of edge is coarser than the edge between parent and child. There are only 24 kinds of dependency types but exist hundreds of edge combination possibilities between parent and child. Thus, this feature relived the data sparseness problem to a certain extent.

There are two main types of errors in the results: one is that the labeling frames of target words are not correct. For example, in the sentence "白洁异常强硬地表示，"不获全胜决不收兵"。" , the correct frame of "表示"should be "表达" instead of "陈述", because it described the attitude of "白洁" not declared a fact or a phenomena. However, this kind of deep semantics of sentence couldn't be capured by T-CRF model based on the dependency syntax. The other is that the labeling frames of some target words are tag "null". The reason is that some lexical units can't evoke a frame sometimes, so in the training data, these words are annotated "null".

5.3 Contrast Experiments

Qu (2008) argues that any words in a sentence has a certain attraction between each other and thus constitute the grammars and semantic structure of the sentence. Based on this cognition, he proposed a generalized collocation theory, which includes fixed collocation, loose collocation and Co-occurrence collocation. According to this theory, a context computing model RFR_SUM was presented to deal with the WSD task.

In essence, frame identification also belongs to context computing, so it should be reasonable to solve this problem with the generalized collocation theory. However, our current corpus is too insufficient to reflect all these three collocations in the statistical sense. Hence, we proposed a method named compatibility of lexical unit based on the Co-occurrence collocation to identify frame for ambiguous target word.

Window size	Precision					
	T1	T2	T3	T4	T5	T6
1	0.7682	0.7219	0.7351	0.8013	0.8146	0.7947
2	0.7682	0.7152	0.6887	0.7881	0.8146	0.7947
3	0.7417	0.6623	0.6689	0.7351	0.8013	0.7616

Table 3. Precisions of different templates based on three types of window size

The connotation of compatibility of lexical unit is as follows. In the CFN frame database, every frame defines a lexical units set, in which each of lexical unit can evoke this frame. When one of these lexical units serves as the target word in a sentence, we can use the compatibilities of other lexical units in this set with the sentence to reflect the consistency between this frame and current sentence. The compatibility of lexical unit with the sentence is computed by the Co-occurrence frequency of lexical unit and the notional words in the sentence in a large corpus. The calculation is as below.

Suppose l_i in the lexical units set $L = \{l_1, l_2, ...l_i, ..., l_m\}$ serves as the target word in the sentence S. The words in S except the functional words and l_i constitute a word set $W = \{w_1, w_2, ..., w_n\}$. And the compatibility of L with S is denoted as C.

$$C = \frac{c(l_1, W) + c(l_2, W) + ...c(l_m, W)}{m},$$

where m is the number of lexical units in L.

$$c(l_j, W) = \frac{f(l_j, w_1) + f(l_j, w_2) + ... + f(l_j, w_n)}{n},$$

where n is the number of words in W.

$$f(l_j, w_k) = \frac{count(l_j, w_k)}{sum},$$ where $count(l_j, w_k)$ represents the number of sentences, in which l_j and w_k occur together, and these sentences come from the corpus of Peking University People's Daily, January 1998. sum is the total number of sentences in the same People's Daily corpus.

In this way, the consistency between a frame and the current sentence is scored by the compatibility of L belonging to the candidate frame with this sentence, and the one with highest score is regarded as the correct frame. For our test data, 71.73% precision based on this method was obtained.

This model displayed a decline in precision of about 10% over the T-CRF. Analysis of the results found that the compatibility based on Co-occurrence collocation can only reflect a weak correlation between words, neglecting the position and syntactic structure information in a sentence.

In addition, we used the most-frequency-frame experiment as the baseline. In the corpus consisted of 940 training sentences and 128 test sentences, the frequency of each frame was counted for ranking. The result of this method obtained 61.23% precision, which proved that T-CRF model performed obvious improvement.

6 Conclusions

In this paper, we investigated the problem of Frame Identification in Chinese which is the first work on Chinese FrameNet. A tree-structured conditional random field (T-CRF) model was applied to this task based on the dependency syntactic structure. This model provides a way to incorporating the long-distance dependencies between target words and the syntactic related words with it. In our experiments, the syntactic dependency features were shown to work effectively for Frame Identification, with 71.73%, 76.82%, and 81.46% precision for compatibility of lexical unit, CRF and T-CRF, respectively.

Although a relatively good performance was achieved on the test data, the small-scale and simplicity of sentence structure in corpus cannot be ignored compared with the FrameNet corpus. However, the experimental results that we gained is still promising, suggesting that our model is comparatively appropriate to the Frame Identification task and still has a great potential for improvement. The next work will focus on the three aspects: firstly, build a larger corpus containing various sentence structures in Chinese; the other is that more semantic features will be tried to add in the T-CRF model, such as the frame elements and the semantic relations between frames, finally, we will try to identify frames of target words using other machine learning methods which has been proved high performance in this task.

Acknowledgements

This work is supported by NSFC Grant: 60970053 and International Scientific and Technological Cooperation of Shanxi Province Grant: 2010081044. In addition, the au-

thors would like to thank HIT-CIR for their LTP.

References

Charles J. Fillmore. 1982. Frame Semantics. In *Linguistic in the Morning Calm*, pages 111-137, Seoul, Korea: Hanshin Publishing Company.

Collin Baker, Michael Ellsworth and Katrin Erk. 2007. SemEval'07 Task 19: Frame Semantic Structure Extraction. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 99-104, Prague.

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The Berkeley FrameNet project. In *Proceedings of the COLING-ACL,* pages 86-90, Montreal, Canada.

Cosmin Adrian Bejan and Hathaway Chris. 2007. UTD-SRL: A Pipeline Architecture for Extracting Frame Semantic Structures. In *45th annual meeting of Association for Computational Linguistics*. pages 460-463, Prague.

Igor A. Mel'ˇcuk. 1988. Dependency Syntax: Theory and Practice. *State University Press of New York*, Albany.

John Lafferty, Andrew McCallum and Fernando Pereira. 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *proceedings of the 18th International Conference on Machine Learning,* pages 282-289, San Francisco, CA, USA.

Jorge Nocedal and Stephen J. Wright. 1999. *Numerical Optimization*. Springer, New York.

Jun Hatori, Yusuke Miyao and Jun'ichi Tsujii. 2009. On Contribution of Sense Dependencies to Word Sense Disambiguation. *Natural Language Processing*, 16(5):51-77.

Katrin Erk. 2005. Frame assignment as word sense disambiguation. In *Proceedings of the 6th International Workshop on Computational Semantics (IWCS-6)* .

Ken. Litkowski. 2007. CLR: Integration of FrameNet in a Text Representation System. In *45th annual meeting of Association for Computational Linguistics*. pages 113-116, Prague.

Pranjal Awasthi, Aakanksha Gagrani and Balaraman Ravindran. 2007. Image modeling using tree structured conditional random fields. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI 2007).* Pages 2060-2065.

Qu Weiguang. 2008. Automatic Disambiguation of Modern Chinese Words in Word-level. *Beijing: Science Press(in Chinese).*

Richard Johansson and Nugues Pierre. 2007. LTH: Semantic Structure Extraction using Nonprojective Dependency Trees. In *45th annual meeting of Association for Computational Linguistics*. pages 227-230, Prague.

Tang Jie, Mingcai Hong, Juanzi Li, and Bangyong Liang. 2006. Tree-structured Conditional Random Fields for Semantic Annotation. In *Proceedings of 5th International Conference of Semantic Web (ISWC'2006)*, Athens, GA, USA

Trevor Cohn and Philip Blunsom. 2005. Semantic role labeling with tree conditional random fields. In *Proceedings of CoNLL2005.*

Wang Ruiqin and Fansheng-Kong. 2009. The Research of Unsupervised Word Sense Disambiguation. *Journal of Software*, (20)8: pages 2138−2152.

Xue Nianwen and Martha Palmer. 2005. Automatic Semantic Role Labeling for Chinese Verbs. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence*. Edinburgh, Scotland.

You Liping, Kaiying Liu. 2005. Building Chinese FrameNet database. In *Proceedings of IEEE NLP-KE'05.*

Linguistic Cues for Distinguishing Literal and Non-Literal Usages

Linlin Li and **Caroline Sporleder**
Department of Computational Linguistics
Saarland University
{linlin, csporled}@coli.uni-saarland.de

Abstract

We investigate the effectiveness of different linguistic cues for distinguishing literal and non-literal usages of potentially idiomatic expressions. We focus specifically on features that generalize across different target expressions. While idioms on the whole are frequent, instances of each particular expression can be relatively infrequent and it will often not be feasible to extract and annotate a sufficient number of examples for each expression one might want to disambiguate. We experimented with a number of different features and found that features encoding lexical cohesion as well as some syntactic features can generalize well across idioms.

1 Introduction

Nonliteral expressions are a major challenge in NLP because they are (i) fairly frequent and (ii) often behave idiosyncratically. Apart from typically being semantically more or less opaque, they can also disobey grammatical constraints (e.g., *by and large*, *lie in wait*). Hence, idiomatic expressions are not only a problem for semantic analysis but can also have a negative effect on other NLP applications (Sag et al., 2001), such as parsing (Baldwin et al., 2004).

To process non-literal language correctly, NLP systems need to recognise such expressions automatically. While there has been a significant body of work on idiom (and more generally multiword expression) detection (see Section 2), until recently most approaches have focused on a *type-based classification*, dividing expressions into "idiomatic" or "not idiomatic" irrespective of their actual use in a discourse context. However, while some expressions, such as *by and large*, always have a non-compositional, idiomatic meaning, many other expressions, such as *break the ice* or *spill the beans*, can be used literally as well as idiomatically and for some expressions, such as *drop the ball*, the literal usage can even dominate in some domains. Consequently, those expressions have to be disambiguated in context (*token-based classification*).

We investigate how well models for distinguishing literal and non-literal use can be learned from annotated examples. We explore different types of features, such as the local and global context, syntactic properties of the local context, the form of the expression itself and properties relating to the cohesive structure of the discourse. We show that several feature types work well for this task. However, some features can generalize across specific idioms, for instance features which compute how well an idiom "fits" its surrounding context under a literal or non-literal interpretation. This property is an advantage because such features are not restricted to training data for a specific target expression but can also benefit from data for other idioms. This is important because, while idioms as a general linguistic class are relatively frequent, instances of each particular idiom are much more difficult to find in sufficient numbers. The situation is exacerbated by the fact the distributions of literal vs. non-literal usage tend to be highly skewed, with one usage (often the non-literal one) being much more frequent than the other. Finding sufficient examples of the minority class can then be difficult, even if instances are extracted from large corpora. Furthermore, for highly skewed distributions, many more majority class examples have to be annotated to obtain an acceptable number of minority class instances.

We show that it is possible to circumvent this problem by employing a generic feature space that

Coling 2008: Poster Volume, pages 683–691,
Beijing, August 2010

looks at the cohesive ties between the potential idiom and its surrounding discourse. Such features generalize well across different expressions and lead to acceptable performance even on expressions unseen in the training set.

2 Related Work

Until recently, most studies on idiom classification focus on type-based classification; sofar there are only comparably few studies on token-based classification. Among the earliest studies on token-based classification were the ones by Hashimoto et al. (2006) on Japanese and Katz and Giesbrecht (2006) on German. Hashimoto et al. (2006) present a rule-based system in which lexico-syntactic features of different idioms are hard-coded in a lexicon and then used to distinguish literal and non-literal usages. The features encode information about the passivisation, argument movement, and the ability of the target expression to be negated or modified. Katz and Giesbrecht (2006) compute meaning vectors for literal and non-literal examples in the training set and then classify test instances based on the closeness of their meaning vectors to those of the training examples. This approach was later extended by Diab and Krishna (2009), who take a larger context into account when computing the feature vectors (e.g., the whole paragraph) and who also include prepositions and determiners in addition to content words.

Cook et al. (2007) and Fazly et al. (2009) take a different approach, which crucially relies on the concept of *canonical form* (CForm). It is assumed that for each idiom there is a fixed form (or a small set of those) corresponding to the syntactic pattern(s) in which the idiom normally occurs (Riehemann, 2001).The canonical form allows for inflectional variation of the head verb but not for other variations (such as nominal inflection, choice of determiner etc.). It has been observed that if an expression is used idiomatically, it typically occurs in its canonical form (Riehemann, 2001). Cook et al. exploit this behaviour and propose an unsupervised method in which an expression is classified as idiomatic if it occurs in canonical form and literal otherwise. Canonical forms are determined automatically using a statistical, frequency-based measure.

Birke and Sarkar (2006) model literal vs. non-literal classification as a word sense disambiguation task and use a clustering algorithm which compares test instances to two seed sets (one with literal and one with non-literal expressions), assigning the label of the closest set.

Sporleder and Li (2009) propose another unsupervised method which detects the presence or absence of cohesive links between the component words of the idiom and the surrounding discourse. If such links can be found the expression is classified as literal otherwise as non-literal. Li and Sporleder (2009) later extended this work by combining the unsupervised classifier with a second-stage supervised classifier.

Hashimoto and Kawahara (2008) present a supervised approach to token-based idiom distinction for Japanese, in which they implement several features, such as features known from other word sense disambiguation tasks (e.g., collocations) and idiom-specific features taken from Hashimoto et al. (2006). Finally, Boukobza and Rappoport (2009) also experimented with a supervised classifier, which takes into account various surface features.

In the present work, we also investigate supervised models for token-based idiom detection. We are specifically interested in which types of features (e.g., local context, global context, syntactic properties) perform best on this task and more specifically which features generalize across idioms.

3 Data

We used the data set created by Sporleder and Li (2009), which consists of 13 English expressions (mainly V+PP or V+NP) that can be used both literally and idiomatically, such as *break the ice* or *play with fire*.[1] To create the data set all instances of the target expressions were extracted from the Gigaword corpus together with five paragraphs of context and then labelled manually as 'literal' or 'non-literal'. Overall the data set consists of just under 4,000 instances. For most ex-

[1] We excluded four expressions from the original data set because their number of literal examples was very small (< 2).

pressions the distribution is heavily skewed towards the idiomatic interpretation, however for some, like *drop the ball*, the literal reading is more frequent. The number of instances varies, ranging from 15 for *pull the trigger* to 903 for *drop the ball*. While the instances were extracted from a news corpus, none of them are domain-specific and all expressions also occur in the BNC, which is a balanced, multi-domain corpus.

To compute the features which we extract in the next section, all instances in our data sets were part-of-speech tagged by the MXPOST tagger (Ratnaparkhi, 1996), parsed with the MaltParser[2], and named entity tagged with the Stanford NE tagger (Finkel et al., 2005). The lemmatization was done by RASP (Briscoe and Carroll, 2006).

4 Indicators of Idiomatic and Literal Usage

In this study we are particularly interested in which linguistic indicators work well for the task of distinguishing literal and idiomatic language use. The few previous studies have mainly looked at the lexical context in which and expression occurs (Katz and Giesbrecht, 2006; Birke and Sarkar, 2006). However, other properties of the linguistic context might also be useful. We distinguish these features into different groups and discuss them in the following sections.

4.1 Global Lexical Context (glc)

That the lexical context might be a good indicator for the usage of an expression is obvious when one looks at examples as in (1) and (2), which suggest that literal and non-literal usages of a specific idiom co-occur with different sets of words. Nonliteral uses of *break the ice* (1), for instance, tend to occur with words like *discuss*, *bilateral* or *relations*, while literal usages (2) predictably occur with, among others, *frozen*, *cold* or *water*. What we are looking at here is the global lexical context of an expression, i.e., taking into account previous and following sentences. We are specifically looking for words which are either semantically related (in a wide sense) to the literal or the nonliteral sense of the target expression. The presence or absence of such words can be a good indicator of how the expression is used in a context.

(1) "Gujral will meet Sharif on Monday and **discuss bilateral relations**," the Press Trust of India added. The minister said Sharif and Gujral would be able to "break the ice" over Kashmir.

(2) Meanwhile in Germany, the **cold** penetrated Cologne cathedral, where worshippers had to break the ice on the **frozen** holy **water** in the font.

We implemented two sets of features which encode the global lexical context: *salient words* and *related words* as described in Li and Sporleder (2009). The former feature uses a variant of tf.idf to identify words that are particulary salient for different usages. The latter feature identifies words which are most strongly related to the component words of the idiom.

We notice that sometimes several idioms cooccur within the same instance. This is to say that nonliteral usages may be indicators of each other since authors may put them in a same context to convey a specific opinion (e.g., irony). Due to this, global lexical context features may also generalize across idioms to some extend.

4.2 Local Lexical Context (locCont)

In addition to the global context, the local lexical context, i.e., the words preceding and following the target expression, might also provide important information. One obvious local clue are words like *literally* or *metaphorically speaking*, which when preceding or following an expression might indicate its usage. Unfortunately, such clues are not only very rare (we only found a handful in nearly 4,000 annotated examples) but also not always reliable. For instance, it is not difficult to find examples like (3) and (4) where the word *literally* is used even though the idiom clearly has a non-literal meaning.

(3) In the documentary the producer **literally** spills the beans on the real deal behind the movie production.

(4) The new philosophy is blatantly permissive and **literally** passes the buck to the House's other committees.

[2] http://maltparser.org/index.html

However, there are other local cues. For example, we found that the word *just* before *get ones feet wet* tends to indicate non-literal usage as in (5). Non-literal usage can also be indicated by the occurrence of the prepositions *over* or *between* after *break the ice* as in (1) and (6). While such cues are not perfect they often make one usage more likely than the other. Unlike the semantically based global cues, many local clues are more rooted in syntax, i.e., local cues work because specific *constructions* tend to be more frequent for one or the other usage.

(5) The wiki includes a page of tasks suitable for those **just** getting their feet wet.

(6) Would the visit of the minister help break the ice **between** India and Pakistan?

Another type of local cues involves selectional preferences. For example, idiomatic usage is probable if the subject of *play with fire* is a country as in (7) or if *break the ice* is followed by a *with*-PP whose NP refers to a person (8).

(7) Dudayev repeated his frequent warnings that **Russia** was playing with fire.

(8) Edwards usually manages to break the ice with the taciturn **monarch**.

Based on those observations, we encode which words occur in a ten word window around the target expression, five pre-target words and five post-target words, as the locCont features.

4.3 Discourse Cohesion (dc)

We implemented two features, *related score* and *discourse connectivity*, which take into account the cohesive structure of an expression in its context as described by Li and Sporleder (2009). In addition, we also included the prediction of the cohesion graph proposed by Sporleder and Li (2009) as an additional feature. These features look at the lexical cohesion between an expression and the surrounding discourse, so they are more likely to generalize across different idioms.

4.4 Syntactic Structure (allSyn)

To capture syntactic effects, we encoded information of the **head node (heaSyn)** of the target expression in the dependency tree (e.g., *break* in the dependency tree in Figure 1). The syntactic features we encoded are the **parent node (parSyn)**, **sibling nodes (sibSyn)** and **children nodes (chiSyn)** of the head node. These nodes include the following type of syntactic information:

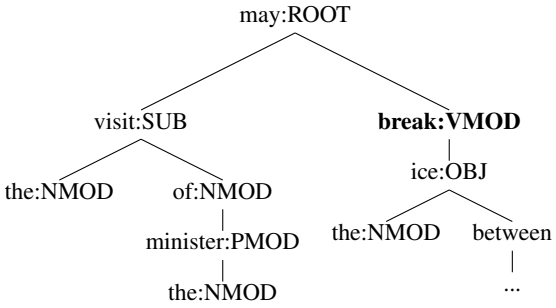


Figure 1: Dependency tree for a nonliteral example of *break the ice* (*The visit of the minister may break the ice between India and Pakistan.*)

Dependency Relation of the Verb Phrase The whole idiomatic expression used as an object of a preposition can be an indicative factor of idiomatic usage (see Example 9). This property is captured by the *heaSyn* feature.

(9) Ross headed back last week to Washington to brief president Bill Clinton on the Hebron talks after achieving a breakthrough **in** breaking the ice in the Hebron talks by arranging an Arafat-Netanyahu summit .

Modal Verbs usually appear in the parent position of the head verb (*parSyn*). Modals can be an indicator of idiomatic usage such as **may** in Figure 1. In contrast, the modal **had to** is indicative that the same phrase is used literally (Example 10).

(10) Dad **had to** break the ice on the chicken troughs.

Subjects can also provide clues about the usage of an expression, e.g., if selectional preferences are disobeyed. For instance, **visit** as a subject of the verb phrase *break the ice* is an indicator of idiomatic usage (see Figure 1). Subjects typically appear in the children position of the head verb (*chiSyn*), but sometimes may appear in the sibling position (*sibSyn*) as in Figure 1 .

Verb Subcat We also encode the arguments of the head verb of the target expression. These arguments can be, for example, additional PPs. This feature encodes syntactic constraints and attempts

to model selectional restrictions. The likelihood of subcategorisation frames may differ for the two usages of an expression, e.g., non-literal expressions often tend to have a shorter argument list. For instance, the subcat frame <PP-on, PP-for> intuitively seems more likely for literal usages of the expression *drop the ball* (see Example 11) than for non-literal ones, for which <PP-on> is more likely (12). To capture subcategorisation behaviour, we encode the children nodes of the head node (*chiSyn*).

(11) US defender Alexi Lalas twice went close to forcing an equaliser , first with a glancing equaliser from a Paul Caligiuri free kick and then from a Wynalda corner when Prunea dropped the ball **[on the ground]** only **[for Tibor Selyme to kick frantically clear]** .

(12) "Clinton dropped the ball **[on this]**," said John Parachini.

Modifiers of the verb can also be indicative of the usage of the target expression. For example, in 13, the fact that the phrase *get one's feet wet* is modified by the adverb *just* suggest that it is used idiomatically. Similar to verb subcat, modifiers are often appear in the children position (*chiSyn*).

(13) The wiki includes a page of tasks suitable for those **just** getting their feet wet.

Coordinated Verb Which verbs are coordinated with the target expression, if any, can also provide cues for the intended interpretation. For example, in (14), the fact that *break the ice* is coordinated with *fall* suggest that it is used literally. The coordinated verb can appear at the sibling position, children position, or some other position of the head verb depending on the parser. The Maltparser tends to put the coordinated verbs in the children position (*chiSyn*).

(14) They may break the ice and **fall** through.

4.5 Other Features

Named Entities (ne) can also indicate the usage of an expression. For instance, a country name in the subject position of the target expression *break the ice* is a strong indicator of this phrase being used idiomatically (see Example 7). Diab and Bhutada (2009) find that NE-features perform best. They used a commercial NE-tagger with 19 classes. We used the Stanford NE tagger (Finkel et al., 2005), and encoded three named entity classes ("person", "location", "organisation") in the feature vector.

Indicative Terms (iTerm) Some words such as *literally, proverbially* are also indicative of literal or idiomatic usages. We encoded the frequencies of those indicative terms as features.

Scare Quotes (quote) This feature encodes whether the idiom is marked off by scare quotes, which often indicates non-literal usage (15).

(15) Do consider "getting your feet wet" online, using some of the technology that is now available to us.

5 Experiments

In the previous section we discussed different linguistic cues for idiom usage. To determine which of these cues work best for the task and which ones generalize across different idioms, we carried out three experiments. In the first one (Section 5.1) we trained one model for each idiom (see Section 3) and tested the predictiveness of each feature type individually as well as all features together. In the second experiment (Section 5.2), we trained one generic model for all idioms and determined how the performance of this model differs from the idiom-specific models. Specifically we wanted to know whether the model would benefit from the additional training data available by combining information from several idioms. Finally (Section 5.3), we tested the generic model on *unseen* idioms to determine whether these could be classified based on generic properties even if training data for the target expressions had not been seen.

5.1 Idiom Specific Models

The first question we wanted to answer was how difficult token-based idiom classification is and which of the features we defined in the previous section work well for this task. We implemented a specific classifier for each of the idioms in the data set. We trained one model for all features in combination and one for each individual feature. Because the data set is not very big we decided to run these experiments in 10-fold stratified

cross-validation mode. We used the SVM classifier (SMO) from Weka.[3]

Table 1 shows the results. We report the precision (Prec.), recall (Rec.) and F-Score for the literal class, as well as the accuracy. Note that due to the imbalance in the data set, accuracy is not a very informative measure here; a classifier always predicting the majority class would already obtain a relatively high accuracy. The literal F-Score obtained for individual idioms varies from 38.10% for *bite one's tongue* to 96.10% for *bounce of the wall*. However, the data sets for the different idioms are relatively small and it is impossible to say whether performance differences on individual idioms are accidental, or due to differences in training set size or due to some inherent difficulty of the individual idiom. Thus we chose not to report the performance of our models on individual idioms but on the whole data set for which the numbers are much more reliable. The final performance confusion matrix is the sum over all individual idiom confusion matrices.

	Avg. literal			Avg.
feature	Prec.	Rec.	F-Score	Acc.
all	89.84	77.06	82.96	93.36
glc+dc	90.42	76.44	82.85	93.36
allSyn	76.30	86.13	80.92	91.48
heaSyn	76.64	85.77	80.95	91.53
parSyn	76.43	88.34	81.96	91.84
chiSyn	76.49	88.22	81.94	91.84
sibSyn	76.27	88.34	81.86	91.78
locCont	76.51	88.34	82.00	91.86
ne	76.49	88.22	81.94	91.84
iTerm	76.51	88.34	82.00	91.86
quote	76.51	88.34	82.00	91.86
Base_{maj}	76.71	88.34	82.00	91.86

Table 1: Performance of idiom-specific models (averaged over different idioms), 10-fold stratified cross-validation.

The Baseline (Base) is built based on predicting the majority class for each expression. This means predicting *literal* for the expressions which consist of more literal examples and *nonliteral* for the expressions consisting of more nonliteral examples. We notice the baseline gets a fairly high performance (Acc.=91.86%).

The results show that the expressions can be classified relatively reliably by the proposed features. The performance beats the majority baseline statistically significantly ($p = 0.01$, χ^2 test). We noticed that parSyn, chiSyn, locCont, iTerm and quote features are too sparse. These individual features cannot guide the classifier. As a result, the classifier only predicts the majority class which results in a performance similar to the baseline. Some of the syntactic features are less sparse and they get different results from the baseline classifier, however, the performances of these features are actually worse than the baseline. This may be due to the relatively small training size in each idiom specific model. When adding those features together with statistical-based features (glc+dc), the performance of the literal class can be improved slightly. However, we did not observe any performance increase on the accuracy.

5.2 Generic Models

Having verified that literal and idiomatic usages can be distinguished with some success by training expression-specific models, we carried out a second experiment in which we merged the data sets for different expressions and trained one generic model. We wanted to see whether a generic model, which has access to more training data, performs better and whether some features, e.g., the cohesion features profit more from this. The experiment was again run in 10-fold stratified cross-validation mode (using 10% from each idiom in the test set in each fold).

Table 2 shows the results. The baseline classifier always predict the majority class 'nonliteral'. Note that the result of this baseline is different from the majority baseline in the idiom specific model. In the idiom specific model, there are three expressions [4] for which the majority class is 'literal'.

Unsurprisingly, the F-Score and accuracy of the combined feature set drops a bit. However, the performance still statistically significantly beats the majority baseline classifier ($p << 0.01$, χ^2 test). Similar to previous observation, the

[3] http://www.cs.waikato.ac.nz/ml/weka/

[4] I.e., *bounce off the wall*, *drop the ball*, *pull the trigger*

feature	Avg. literal Prec.	Rec.	F-Score	Avg. Acc.
all	89.59	65.77	73.22	89.90
glc+dc	82.53	60.86	70.06	89.08
allSyn	50.83	59.88	54.99	79.42
heaSyn	50.57	59.88	54.83	79.29
sibSyn	33.33	0.86	1.67	78.83
ne	62.45	20.00	30.30	80.69
iTerm	40.00	0.25	0.49	78.99
Base_{maj}	–	–	–	79.01

Table 2: Performance of the generic model (averaged over different idioms), 10-fold stratified cross-validation.

statistical-based features (glc+dc) work the best, while the syntactic features are also helpful. However, the local context, iTerm, quote features are very sparse and, as in the idiom-specific experiments, the performances of these features are similar to the majority baseline classifier. We excluded them from the Table 2.

The numbers show that the syntactic features help more in this model compared with the idiom-specific model. When including these features, literal F-Score increases by 3.16% while accuracy increases by 0.9%. It seems that the syntactic features benefit from the increased training set. This is evidence that these features can generalize across idioms. For instance, the phrase "The US" on the subject position may be not only indicative of the idiomatic usage of *break the ice*, but also of idiomatic usage of *drop the ball*.

We found that the indicative terms are rare in our corpus. This is the reason why the recall rate of the indicative terms is very low (0.25%). The indicative terms are not very predictive of literal or non-literal usage, since the precision rate is also relatively low (40%), which means those words can be used in both literal and nonliteral cases.

5.3 Unseen Idioms

In our final experiment, we tested whether a generic model can also be applied to completely new expressions, i.e., expressions for which no instances have been seen in the data set. Such a behaviour would be desireable for practical purposes as it is unrealistic to label training data for each idiom the model might possibly encounter in a text. To test whether the generic model does indeed generalize to unseen expressions, we test it on all instances of a given expression while training on the rest of the expressions in the dataset. That is, we used a modified cross-validation setting, in which each fold contains instances from one expression in the test set. Since our dataset contains 13 expressions, we run a 13-fold cross validation. The final confusion matrix is the sum over each confusion matrix in each round.

feature	Avg. literal Prec.	Rec.	F-Score	Avg. Acc.
all	96.70	81.65	88.54	95.41
glc+dc	96.93	77.00	85.83	94.48
allSyn	52.54	58.77	55.48	79.52
heaSyn	51.35	59.47	55.11	78.96
sibSyn	55.56	2.32	4.46	78.38
ne	61.89	19.05	29.13	79.87
iTerm	66.67	0.7	1.38	78.36
Base_{maj}	–	–	–	79.01

Table 3: Performance of the generic model on unseen idioms (cross validation, instances from each idiom are chosen as test set for each fold)

The results are shown in Table 3. Similar to the generic model, we found that the cohesion features and syntactic features do generalize across expressions. Statistical features (glc+dc) perform well in this experiment. When including more linguistically orientated features, the performance can be further increased by almost 1%. In line with former observations, the sparse features mentioned in the former two experiments also do not work for this experiments. We also excluded them from the table.

One interesting finding about this experiment of this model is that the F-Score is higher than for the "generic model". This is counter-intuitive, since in the generic model, each idiom in the testing set has examples in the training set, thus, we might expect the performance to be better due to the fact that instances from the same expression appearing in the training set are more informative compared with instances from different idioms. Further analysis revealed that there are some expressions for which it may actually be beneficial to

train on other expressions, as the evidence of some features may be misleading.

	literal F-S.		Acc.	
feature	Spe.	Gen.	Spe.	Gen.
all	86.85	**91.79**	80.67	**88.37**
glc+dc	86.75	**88.84**	80.67	**84.61**
allSyn	**85.71**	71.94	**75.28**	61.13
heaSyn	**85.79**	71.94	**75.39**	61.13

Table 4: Comparing the performance of the idiom *drop the ball* on the idiom specific model (Spe.) and generic model (Gen.)

Table 4 shows the comparison of the performance of *drop the ball* on the idiom specific model and the generic model on unseen idioms. It can be seen that the statistical features (glc+dc) work better for the model that is trained on the instances from other idioms than the model which is trained on the instances of the target expression itself. We found this is due to the fact that *drop the ball* is especially difficult to classify with the discourse cohesion features (dc). The literal cases are often found in a context containing words, such as **fault**, **mistake**, **fail**, and **miss**, which are used to describe a scenario in a baseball game,[5] while, on the other hand, those context words are also closely semantically related to the idiomatic reading of *drop the ball*. This means the classifier can be mislead by the cohesion features of the literal instances of this idiom in the training set, since they exhibit strong idiomatic cohesive links with the target expression. When excluding *drop the ball* from the training set, the cohesive links in the training data are less noisy. Thus, the performance increases. Unsurprisingly, the performance of syntactic features works better for the idiom specific model compared with the unseen idiom model.

6 Conclusion

Idioms on the whole are frequent but instances of each particular idiom can be relatively infrequent (even for common idioms like "spill the beans"). The classes can also be fairly imbalanced, with one class (typically the nonliteral interpretation) being much more frequent than the other. This causes problems for training data generation. For idiom specific classifiers, it is difficult to obtain large data sets even when extracting from large corpora and it is even more difficult to find sufficient examples of the minority class. In order to address this problem, we looked for features which can generalize across idioms.

We found that statistical features (glc+dc) work best for distinguishing literal and nonliteral readings. Certain linguistically motivated features can further boost the performance. However, those linguistic features are more likely to suffer from data sparseness, as a result, they often only predict the majority class if used on their own. We also found that some of the features that we designed generalize well across idioms. The cohesion features have the best generalization ability, while syntactic features can also generalize to some extent.

[5]The corpus contains many sports news text

Acknowledgments

This work was funded by the DFG within the Cluster of Excellence MMCI.

References

Baldwin, Timothy, Emily M. Bender, Dan Flickinger, Ara Kim, and Stephen Oepen. 2004. Road-testing the English resource grammar over the British National Corpus. In *Proc. LREC-04*, pages 2047–2050.

Birke, Julia and Anoop Sarkar. 2006. A clustering approach for the nearly unsupervised recognition of nonliteral language. In *Proceedings of EACL-06*.

Boukobza, Ram and Ari Rappoport. 2009. Multi-word expression identification using sentence surface features. In *Proceedings of EMNLP-09*.

Briscoe, Ted and John Carroll. 2006. Evaluating the accuracy of an unlexicalized statistical parser on the PARC DepBank. In *Proceedings of the COLING/ACL on Main conference poster sessions*, pages 41–48.

Cook, Paul, Afsaneh Fazly, and Suzanne Stevenson. 2007. Pulling their weight: Exploiting syntactic forms for the automatic identification of idiomatic expressions in context. In *Proceedings of the ACL-07 Workshop on A Broader Perspective on Multiword Expressions*.

Diab, Mona and Pravin Bhutada. 2009. Verb noun construction mwe token classification. In *Proceedings of the Workshop on Multiword Expressions: Identification, Interpretation, Disambiguation and Applications*, pages 17–22.

Diab, Mona T. and Madhav Krishna. 2009. Unsupervised classification of verb noun multi-word expression tokens. In *CICLing 2009*, pages 98–110.

Fazly, Afsaneh, Paul Cook, and Suzanne Stevenson. 2009. Unsupervised type and token identification of idiomatic expressions. *Computational Linguistics*, 35(1):61–103.

Finkel, Jenny Rose, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of ACL-05*, pages 363–370.

Hashimoto, Chikara and Daisuke Kawahara. 2008. Construction of an idiom corpus and its application to idiom identification based on WSD incorporating idiom-specific features. In *Proceedings of EMNLP-08*, pages 992–1001.

Hashimoto, Chikara, Satoshi Sato, and Takehito Utsuro. 2006. Japanese idiom recognition: Drawing a line between literal and idiomatic meanings. In *Proceedings of COLING/ACL-06*, pages 353–360.

Katz, Graham and Eugenie Giesbrecht. 2006. Automatic identification of non-compositional multiword expressions using latent semantic analysis. In *Proceedings of the ACL/COLING-06 Workshop on Multiword Expressions: Identifying and Exploiting Underlying Properties*.

Li, Linlin and Caroline Sporleder. 2009. Contextual idiom detection without labelled data. In *Proceedings of EMNLP-09*.

Ratnaparkhi, Adwait. 1996. A maximum entropy part-of-speech tagger. In *Proceedings of EMNLP-96*.

Riehemann, Susanne. 2001. *A Constructional Approach to Idioms and Word Formation*. Ph.D. thesis, Stanford University.

Sag, Ivan A., Timothy Baldwin, Francis Bond, Ann Copestake, and Dan Flickinger. 2001. Multiword expressions: a pain in the neck for NLP. In *Lecture Notes in Computer Science*.

Sporleder, Caroline and Linlin Li. 2009. Unsupervised recognition of literal and non-literal use of idiomatic expressions. In *Proceedings of EACL-09*.

Contextual Recommendation based on Text Mining

Yize Li, Jiazhong Nie, Yi Zhang
School of Engineering
University of California Santa Cruz
{yize,niejiazhong,yiz}@soe.ucsc.edu

Bingqing Wang
School of Computer Science Technology
Fudan University
wbq@fudan.edu.cn

Baoshi Yan, Fuliang Weng
Research and Technology Center
Robert Bosch LLC
Baoshi.Yan@us.bosch.com
Fuliang.Weng@us.bosch.com

Abstract

The potential benefit of integrating contextual information for recommendation has received much research attention recently, especially with the ever-increasing interest in mobile-based recommendation services. However, context based recommendation research is limited due to the lack of standard evaluation data with contextual information and reliable technology for extracting such information. As a result, there are no widely accepted conclusions on how, when and whether context helps. Additionally, a system often suffers from the so called cold start problem due to the lack of data for training the initial context based recommendation model. This paper proposes a novel solution to address these problems with automated information extraction techniques. We also compare several approaches for utilizing context based on a new data set collected using the proposed solution. The experimental results demonstrate that 1) IE-based techniques can help create a large scale context data with decent quality from online reviews, at least for restaurant recommendations; 2) context helps recommender systems rank items, however, does not help predict user ratings; 3) simply using context to filter items hurts recommendation performance, while a new probabilistic latent relational model we proposed helps.

1 Introduction

In the information retrieval community, one major research focus is developing proactive retrieval agent that acts in anticipation of information needs of a user and recommends information to the user without requiring him/her to issue an explicit query. The most popular examples of such kind of proactive retrieval agent are recommender systems. Over the last several years, research in standard recommender systems has been improved significantly, largely due to the availability of large scale evaluation data sets such as Netflix. The current research focus goes beyond the standard user-item rating matrix. As researchers start to realize that the quality of recommendations depends on time, place and a range of other relevant users' context, how to integrate contextual information for recommendation is becoming an ever increasingly important topic in the research agenda (Adomavicius and Ricci, 2009).

One major challenge in context-aware recommendation research is the lack of large scale annotated data set. Ideally, a good research data set should contain contextual information besides users' explicit ratings on items. However, such kinds of data sets are not readily available for researchers. Previous research work in context based recommendation usually experiments on a small data set collected through user studies. Although undoubtedly useful, this approach is limited because 1) user studies are usually very expensive and their scales are small; 2) it is very hard for the research community to repeat such study; and 3) a personalized contextual system may not

Coling 2008: Poster Volume, pages 692–700,
Beijing, August 2010

1	I was very excited to try this place and **my wife** took me here on my **birthday**... We ordered a side of the brussell sprouts and they were the highlight of the **night**.
2	**A friend of mine** suggested we meet up here for a **night** of **drinks**... This actually a restaurant with a bar in it, but when we went it was **10pm** and ...

Table 1: Examples of the restaurant reviews

succeed until a user has interacted with it for a long period of time to enable context based recommendation models well trained.

On the other hand, a large amount of review documents from web sites such as tripadvisor.com, yelp.com, cnet.com, amazon.com, are available with certain contextual information, such as time and companion, implicitly in the reviews (see Table 1 for examples). This offers us an opportunity to apply information extraction techniques for obtaining contextual information from the review texts. Together with users' explicit ratings on items, this might lead to a large research data set for context based recommendation and consequently address the cold start issue in the recommender systems. This paper describes the methods that extract the contextual information from online reviews and their impact on the recommendation quality at different accuracy levels of the extraction methods.

Another challenge is how to integrate contextual information into existing recommendation algorithms. Existing approaches can be classified into three major categories: pre-filtering, post-filtering and the modeling based approaches (Oku et al., 2007; Adomavicius and Tuzhilin, 2008). Pre-filtering approaches utilize contextual information to select data for that context, and then predict ratings using a traditional recommendation method on the selected data (Adomavicius et al., 2005). Post-filtering approaches first predict ratings on the whole data using traditional methods, then use the contextual information to adjust results. Both methods separate contextual information from the rating estimation process and leads to unsatisfying findings. For example, Adomavicious et al. (2005) found neither standard collaborative filtering nor contextual reduction-based methods dominate each other in all the cases. In the modeling based approaches, contextual information is used directly in the rating prediction process. For example, Oku et al. (2007) propose a context-aware SVM-based predictive model to classify restaurants into "positive" and "negative" classes, and contextual information is included as additional input features for the SVM classifier. However, treating recommendation as classification is not a common approach, and does not take advantage of the state of art collaborative filtering techniques. In this paper, we propose a new probabilistic model to integrate contextual information into the state of art factorization based collaborative filtering approach, and compare it with several baselines.

2 Mining Contextual Information from Textual Opinions

The context includes any information that can be used to characterize the situation of entities. Examples of context are: location, identity and state of people, companions, time, activities of the current user, the devices being used etc. (Lee et al., 2005). Without loss of generality, we looked into widely available restaurant review data. More specifically, we investigated four types of contextual information for a dining event, as they might affect users' dining decisions, and they have not been studied carefully before. The four types of contextual information are: *Companion* (whether a dining event involves multiple people), *Occasion* (for what occasions the event is), *Time* (what time during the day) and *Location* (in which city the event happens).

2.1 Text Mining Approaches

We developed a set of algorithms along with existing NLP tools (GATE (Cunningham et al., 2002) etc.) for this task. More detailed description of these algorithms is given below.

Time: we classified the meal time into the following types: "breakfast", "lunch", "dinner", "brunch", "morning tea", "afternoon tea". We

compiled a list of lexicons for these different types of meal times, and used a string matching method to find the explicit meal times from reviews. Here, the meal time with an expression, such as "6pm", was extracted using ANNIE's time named entity recognition module from the GATE toolkit. For example, if a user says, "When we went there, it was 10pm", we infer that it was for dinner.

Occasion: The ANNIE's time named entity recognition module recognizes certain special days from text. We augmented ANNIE's lookup function with a list of holidays in the United States from Wikipedia[1] as well as some other occasions, such as birthdays and anniversaries.

Location: Ideally, a location context would be a user's departure location to the selected restaurant. However, such information rarely exists in the review texts. Therefore, we used the location information from a user's profile to approximate.

Companion: Extracting a companion's information accurately from review data is more difficult. We utilized two methods to address the challenge:

Companion-Baseline: This is a string matching based approach. First, we automatically generated a lexicon of different kinds of companion words/phrases by using prepositional patterns, such as "with my (our) NN (NNS)". We extracted the noun or noun phrases from the prepositional phrases as the companion terms, which were then sorted by frequency of occurrence and manually verified. This led to a lexicon of 167 entries. Next, we grouped these entries into 6 main categories of companions: "family", "friend", "couple", "colleague", "food-buddy" and "pet". Finally, the review is tagged as one or more of the companion categories if it contains a corresponding word/phrase in that lexicon.

Companion-Classifier: In order to achieve better precision, we sampled and annotated 1000 sentences with companion terms from the corpus and built three classifiers: 1) a MaxEnt classifier with bag-of-words features, 2) a rule-based classifier, 3) a hybrid classifier. For the rule-based classifier, we looked into the structural aspects of the window where companion terms occurred, specifically, the adjacent verbs and prepositions associated with those terms. We collected high frequency structures including verbs, verb-proposition combinations, and verb-genitive combinations from the whole corpus, and then constructed a list of rules to decide whether a companion context exists based on these structures. For the hybrid classifier, we used the patterns identified by the rule-based classifier as features for the MaxEnt model (Ratnaparkhi, 1998). To train the classifier, we also included features such as POS tags of the verb and of the candidate companion term, the occurrence of a meal term (e.g. "lunch", "dinner"), the occurrence of pronouns (e.g. "we" or "us") and the genitive of the companion term. Based on the evaluation results (using 5-fold cross validation) shown in Table 2, the hybrid classifier is the best performing classifier and it is used for the subsequent experiments in the paper.

	Words	Rule	Hybrid
Precision	0.7181	0.7238	0.7379
Recall	0.8962	0.8947	0.9143
F-Score	0.7973	0.8003	0.8167

Table 2: Evaluation results for the bag-of-words-based classifier (Words), the rule-based classifier (Rule) and the hybrid classifier (Hybrid)

3 Recommendation based on Contextual Information

Next we consider how to integrate various contextual information into recommender systems. Assume there are N items and M users. Each user reviews a set of items in the system. The data set can be represented as a set of quadruplet $D = (y, i, j, \mathbf{c})$, where i is the index of user, j is the index of item, $\mathbf{c}$ is a vector describing the context of this rating data, and y is the rating value. Let $\mathbf{c} = (c_1, ..., c_k)$, where each component c_k represents a type of context, such as "dinner time" or "location=San Jose". The observed features (meta data) of user i and item j are represented as vectors $\mathbf{f}_i$ and $\mathbf{f}_j$ respectively, where each component in the vector represents a type of feature, such as "gender of the user" or "price range of the restaurant". In the rest of this paper, we in-

[1] http://en.wikipedia.org/wiki/List_of_holidays_by_country#United_States_of_America

tegrate context $\mathbf{c}$ into the user's observed features $\mathbf{f}_i$. This makes $\mathbf{f}_i$ a dynamic feature vector, which will change with different context. The goal is to predict ratings for candidate items given user i and context $\mathbf{c}$, and recommend the top items. We present two recommendation models for integrating contextual information in this section.

3.1 Boolean Model

The Boolean Model filters out items that do not match the context. The Boolean model itself returns an item set instead of a ranked list. We further rank the items by predicted rating values. We score items by the Boolean model as follows:

$$s(j) = \begin{cases} s_m(j) & \text{if item } j \text{ matches the context} \\ -\infty & \text{otherwise} \end{cases} \quad (1)$$

where $s_m(j)$ is the predicted rating computed using a rating prediction method m, such as a Collaborative Filtering model without using context.

3.2 Probabilistic Latent Relational Model

We propose a novel Probabilistic Latent Relational Model (PLRM) for integrating contextual information. In a context-aware recommender system, a user's interest for item is influenced by two factors: (1) the user's long-term preference, which can be learned from users' rating history; (2) the current context (how the item matches the current context). To capture the two factors simultaneously, we introduce a new probabilistic model by assuming the rating value $y_{i,j,\mathbf{c}}$ follows a Gaussian distribution with mean $u_{i,j,\mathbf{c}}$ and variance $1/\lambda^{(y)}$:

$$y_{i,j,\mathbf{c}} \sim \mathcal{N}(u_{i,j,\mathbf{c}}, 1/\lambda^{(y)}) \quad (2)$$

$$u_{i,j,\mathbf{c}} = \mathbf{u}_i^T A \mathbf{v}_j + (W_u \mathbf{f}_i)^T (W_v \mathbf{f}_j) \quad (3)$$

where $\mathbf{u}_i$ and $\mathbf{v}_j$ are the hidden representations of user i and item j to be learned from rating data, and W_u and W_v are feature transformation matrices for users and items respectively. In Equation (3), the first term $\mathbf{u}_i^T A \mathbf{v}_j$ is the estimation based on user' long term preferences, where $A = \{a\}$ is a matrix modeling the interaction between $\mathbf{u}_i$ and $\mathbf{v}_j$.[2] The second term $(W_u \mathbf{f}_i)^T (W_v \mathbf{f}_j)$ is the estimation based on current context and the observed features of users and items, since the context $\mathbf{c}$ is integrated into user's observed features $\mathbf{f}_i$.

$\{U, V, A, W\}$ are the parameters of the model to be estimated from the training data set D, where $W = \{W_u, W_v\} = \{w\}$, $U = \{\mathbf{u}_1, \mathbf{u}_2, ...\mathbf{u}_N\}$ and $V = \{\mathbf{v}_1, \mathbf{v}_2, ...\mathbf{v}_M\}$. We assume the prior distribution of the parameters follow the Gaussian distributions centered on 0. We use $1/\lambda^{(u)}$,$1/\lambda^{(v)}$, $1/\lambda^{(w)}$ and $1/\lambda^{(a)}$ to represent the variance of the corresponding Gaussian distributions. The effect of the prior distribution is similar to the ridge regression (norm-2 regularizer) commonly used in machine learning algorithms to control model complexity and avoid overfitting.

The proposed model is motivated by well performing recommendation models in the literature. It generalizes several existing models. If we set A to the identity matrix and W_u, W_v to zero matrices, the model presented in Equation (3) is equivalent to the well known norm-2 regularized singular value decomposition, which performs well on the Netflix competition(Salakhutdinov and Mnih, 2007). If we set A to zero matrix and W_u to identity matrix, the Model (3) becomes the bilinear model that works well on Yahoo news recommendation task (Chu and Park, 2009).

Based on the above model assumption, the joint likelihood of all random variables (U, V, A, W and D) in the system is:

$$P(U, V, A, W, D) = \prod_{(i,j,c,y) \in D} P(y_{i,j,c} | \mathbf{u}_i, \mathbf{v}_j, \mathbf{f}_i, \mathbf{f}_j, A, W_u, W_v) \prod_i P(\mathbf{u}_i) \prod_j P(\mathbf{v}_j) P(A) P(W_u) P(W_v) \quad (4)$$

3.3 Parameter Estimation

We use a modified EM algorithm for parameter estimation to find the posterior distribution of (U, V) and max a posterior (MAP) of (A, W). The estimation can be used to make the final pre-

[2]We introduce A matrix so that the model can also be used to model multiple different types of relationships/interactions jointly, where each type of relationship corresponds to a different A matrix. For the task in this paper, A is not required and can be set to the identity matrix for simplicity. However, we leave A as parameters to be estimated in the rest of this paper for generality.

dictions as follows:

$$\hat{y}_{i,j,c} = \int_{\mathbf{u}_i,\mathbf{v}_j} P(\mathbf{u}_i)P(\mathbf{v}_j)(\mathbf{u}_i^T A\mathbf{v}_j + (W_u\mathbf{f}_i)^T W_v\mathbf{f}_j)d\mathbf{u}_i d\mathbf{v}_j$$

E Step: the Variational Bayesian approach is used to estimate the posterior distributions of U and V. Assuming (A, W) are known, based on Equation 4, we have

$$P(U,V|A,W,D) \propto \prod_{(y,i,j,\mathbf{c})\in D} \mathcal{N}(\mathbf{u}_i^T A\mathbf{v}_j + (W_u\mathbf{f}_i)^T W_v\mathbf{f}_j, 1/\lambda^{(y)}) \times \prod_{i=1}^{M}\mathcal{N}(\mathbf{u}_i|\mathbf{0}, 1/\lambda^{(u)}I)\prod_{j=1}^{N}\mathcal{N}(\mathbf{v}_j|\mathbf{0}, 1/\lambda^{(v)}I)$$

Deriving the exact distribution and use it to predict y will result in intractable integrals. Thus we approximate the posterior with a variational distribution $Q(U,V) = \prod_{i=1}^{M} Q(\mathbf{u}_i)\prod_{j=1}^{N} Q(\mathbf{v}_j)$. $Q(\mathbf{u}_i)$ and $Q(\mathbf{v}_j)$ are restricted to Gaussian distributions so that predicting y using Bayesian inference with $Q(U,V)$ will be straightforward. $Q(U,V)$ can be estimated by minimizing the KL-divergence between it and $P(U,V|A,W,D)$. Since $Q(U,V)$ is factorized into individual $Q(\mathbf{u}_i)$ and $Q(\mathbf{v}_j)$, we can first focus on one $Q(\mathbf{u}_i)$ (or $Q(\mathbf{v}_j)$) at a time by fixing/ignoring other factors. For space considerations, we omit the derivation in this paper. The optimal $Q(\mathbf{u}_i)$ is $\mathcal{N}(\bar{\mathbf{u}}_\mathbf{i}, \mathbf{\Sigma_i})$, where $\bar{\mathbf{u}}_\mathbf{i} = \Sigma_i\mathbf{d}_i$,

$$\begin{aligned}\Sigma_i^{-1} &= \sum_{(y,i,j,\mathbf{c})\in D}\lambda^{(y)}A(\bar{\mathbf{v}}_\mathbf{j}\bar{\mathbf{v}}_\mathbf{j}^T + \Psi_j)A^T \\ &+ \lambda^{(u)}I \\ \mathbf{d}_i &= \sum_{(y,i,j,\mathbf{c})\in D}\lambda^{(y)}\tilde{y}A\bar{\mathbf{v}}_\mathbf{j}\end{aligned}$$

Similarly, the optimal $Q(\mathbf{v}_j)$ is $\mathcal{N}(\bar{\mathbf{v}}_\mathbf{j}, \mathbf{\Psi_j})$, where $\bar{\mathbf{v}}_\mathbf{j} = \Psi_j\mathbf{e}_j$,

$$\begin{aligned}\Psi_j^{-1} &= \sum_{(y,i,j,\mathbf{c})\in D}\lambda^{(y)}A^T(\bar{\mathbf{u}}_\mathbf{i}\bar{\mathbf{u}}_\mathbf{i}^T + \Sigma_i)A \\ &+ \lambda^{(v)}I \\ \mathbf{e}_j &= \sum_{(y,i,j,\mathbf{c})\in D}\lambda^{(y)}\tilde{y}A^T\bar{\mathbf{v}}_\mathbf{j}\end{aligned}$$

M Step: Based on the approximate posterior estimation $Q(U,V)$ derived in the E step, the maximum a posteriori estimation of $\{A, W\}$ can be found by maximizing the expected posterior likelihood $\{\hat{A}, \hat{W}\} = arg\max_{A,W} E_{Q(U,V)}(\log P(A,W,U,V|D))$. This can be done using the conjugate gradient descent method, and the gradient of A, W_u, W_v can be calculated as follows:

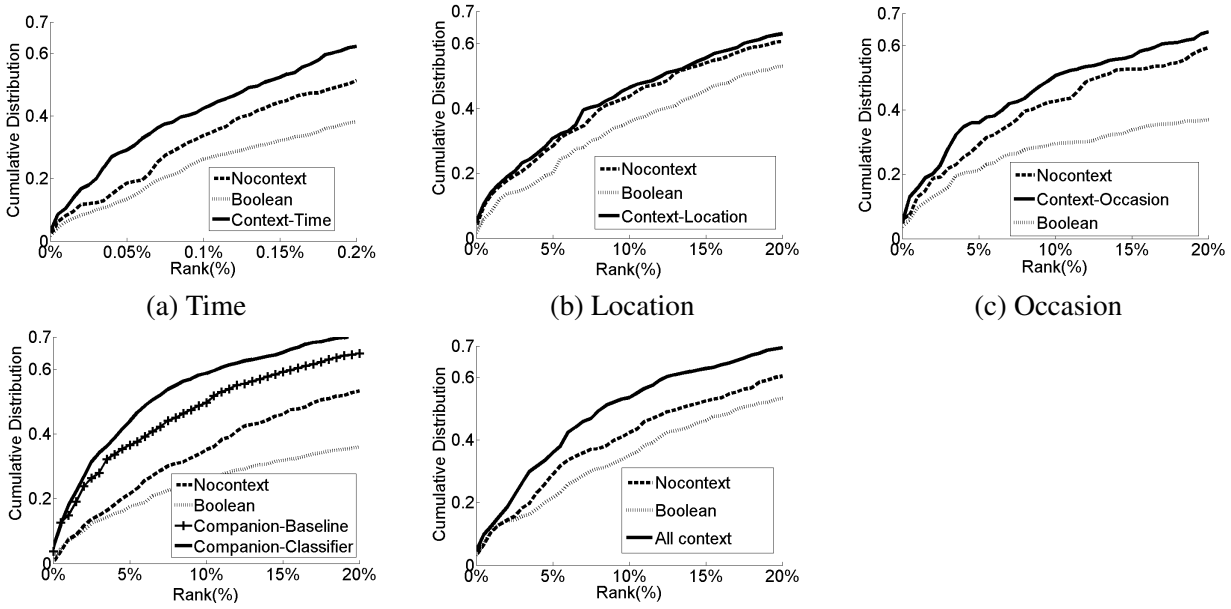
$$\begin{aligned}\frac{\partial\Phi}{\partial A} &= \sum_{(y,i,j,c)\in D}\lambda^{(y)}((\widehat{y}-y)\bar{\mathbf{u}}_\mathbf{i}\bar{\mathbf{v}}_\mathbf{j}^T \\ &+ \bar{\mathbf{u}}_\mathbf{i}\bar{\mathbf{u}}_\mathbf{i}^T A\Psi_j + \Sigma_i A\bar{\mathbf{v}}_\mathbf{j}\bar{\mathbf{v}}_\mathbf{j}^T + \Sigma_i A\Psi_j) \\ &+ \lambda^{(a)}A \\ \frac{\partial\Phi}{\partial W_u} &= \sum_{(y,i,j,\mathbf{c})\in D}\lambda^{(y)}(\widehat{y}-y)W_v\mathbf{f}_j\mathbf{f}_i^T \\ &+ \lambda^{(w)}W_u \\ \frac{\partial\Phi}{\partial W_v} &= \sum_{(y,i,j,\mathbf{c})\in D}\lambda^{(y)}(\widehat{y}-y)W_u\mathbf{f}_i\mathbf{f}_j^T \\ &+ \lambda^{(w)}W_v\end{aligned}$$

where $\Phi = E_{Q(U,V)}(\log P(A,W,U,V|D))$ and $\widehat{y} = \bar{\mathbf{u}}_\mathbf{i}^\mathbf{T}A\bar{\mathbf{v}}_\mathbf{j} + (W_u\mathbf{f}_i)^T W_v\mathbf{f}_j$.

4 Experimental Methodology

4.1 Data Collection

We collected an evaluation data set from a popular review web site where users review services/products and provide integer ratings from 1 to 5. The user profile and the description of items, such as user gender and the category of restaurants are also collected. The data set used in this paper includes the restaurants in Silicon Valley (Bay area) and the users who ever reviewed these restaurants. We extract context from the review texts. The four kinds of context considered in our paper are described in Section 2.1. For each type of context, we create a subset, in which all reviews contain the corresponding contextual information. Finally we construct four sub data sets and each data set is described by the corresponding context type: Time, Location, Occasion and Companion. We use "All" to represent the whole data set. Statistics about each data set are described in Table 3.



(a) Time (b) Location (c) Occasion

(d) Companion (e) All

Figure 1: Performance on the top-K recommendation task. The plots focus on the top 20% ranking region.

Dataset	#Ratings	#Users	#Items
All	756,031	82,892	12,533
Location	583,051	56,026	12,155
Time	229,321	49,748	10,561
Occasion	22,732	12,689	4,135
Companion	196,000	47,545	10,246

Table 3: Statistics of data

4.2 Experimental Setup

We design the experiments to answer the following questions: 1) Does including contextual information improve the recommendation performance? 2) How does the probabilistic latent relational modeling approach compare with pre-filtering or post-filtering approaches? 3) How does the extraction quality of the contextual information affect the recommendation performance?

To answer the first question, we compare the performance of the Probabilistic Latent Relational Model on a standard collaborative filtering setting where only rating information is considered, indicated by Nocontext. We also evaluate the performance of the Probabilistic Latent Relational Model when integrating contextual information, indicated by Context-X, where X represents the type of contextual information considered. To answer the second question, we compare the performance of Context-X with the pre-filtering Boolean Model, which first uses the context to select items and then ranks them using scores computed by Nocontext. To answer the third question, we compare the recommendation performance for different extraction precision. The performance on the following two recommendation tasks are reported in this paper:

Top-K Recommendation: We rank the items by the predicted rating values and retrieve the top K items. This task simulates the scenario where a real recommender system usually suggests a list of ranked K items to a user. To simulate the scenario that we only want to recommend the 5-star items to users, we treat 5-star rating data in testing data as relevant. Ideally, classic IR measures such as Precision and Recall are used to evaluate the recommendation algorithms. However, without complete relevance judgements, standard IR evaluation is almost infeasible. Thus we use a variation of the evaluation method proposed by Koren (Koren, 2008).

Rating Prediction: Given an active user i and a target item j, the system predicts the rating of user

	Training on Sub Data set			Training on the Whole Data set		
Testing Data	ItemAvg	Nocontext	Context	ItemAvg	Nocontext	Context
Time	1.1517	1.0067	1.0067	1.1052	0.9829	0.9822
Companion	1.2657	1.0891	1.0888	1.2012	1.0693	1.0695
Occasion	1.2803	1.1381	1.1355	1.2121	1.0586	1.0583
Location	1.1597	1.0209	1.0206	1.1597	1.0183	1.0183
All context	-	-	-	1.1640	1.0222	1.0219

Table 4: RMSE on the rating prediction task

	Time	CompanionBaseline	CompanionClassifier	Occasion
#Reviews	300	300	300	200
#Contexts	115	148	114	207
Precision	84.4%	62.2%	77.1%	-
Recall	80.2%	95.8%	91.7%	-
F-Score	82.2%	75.4%	83.8%	Accuracy 78.3%

Table 5: Performance of the context extraction module

i on item j. The prediction accuracy is measured by Root Mean Square Error (RMSE), which is commonly used in collaborative filtering research. This task simulates the scenario that we need to guess a user's rating about an item, given that the user has already purchased/selected the item.

For each data set (Time, Companion, Location, Occasion and All), we randomly sample 10% for testing, 80% for training and 10% for validation.

5 Experimental Results

5.1 Performance on Top-K Recommendation

Figure 1(a)-(e) shows the ranking performance on each data set. The x-axis is the rank and the y-axis is the portion of relevant products covered by this level of rank. The results across all data sets are consistent. With contextual information, PLRM Context-X outperforms Nocontext, whereas using context to pre-filter items (Boolean) does not help. It means that contextual information can help if used appropriately, however improperly utilizing context, such as simply using it as a boolean filter, may hurt the recommendation performance. Our proposed PLRM is an effective way to integrate contextual information.

5.2 Performance on Rating Prediction Task

Table 4 summaries the RMSE results of different approaches on the rating prediction task. The RMSE of simply using item's average rating value as the prediction is also reported as a reference since it is a commonly used approach by non personalized recommender systems. For each context, we can either train the model only on the subset that consists of rating data with related context, or train on a bigger data set by adding the rating data without related context. The results on both settings are reported here. Table 4 shows that utilizing context does not affect the prediction accuracy. We may wonder why the effects of adding context is so different on the rating task compared with the ranking task. One possible explanation is that the selection process of a user is influenced by context, while how the user rates an item after selecting it is less relevant to context. For example, when a user wants to have a breakfast, he may prefer a cafeteria rather than a formal restaurant. However, how the user rates this cafeteria is more based on user's experiences in the cafeteria, such as quality of services, food, price, environment, etc.

5.3 How does Text Mining Accuracy Affect Recommendation

To evaluate the extraction performance on "Companion", "Time" and "Occasion", we randomly sample some reviews and evaluate the perfor-

mance on the samples[3]. The results are shown in Table 5. Compared with other contexts, the extraction of companion context is more challenging and the string matching baseline algorithm produces significantly inferior results. However, by using a MaxEnt classifier with features selection, we can boost the precision of the companion context extraction to a level comparable to other contexts.

To further investigate the relationship between the quality of the extracted context and the performance of the recommender system, we compare the recommendation performance of Companion-Baseline and Companion-Classifier in Figure 1(d). It shows that improving the quality of the extraction task leads to a significant improvement on the recommender systems' top-K ranking task.

6 Conclusions

Reviews widely available online contain a large amount of contextual information. This paper proposes to leverage information extraction techniques to help recommender systems to train better context-aware recommendation models by mining reviews. We also introduce a probabilistic latent relation model for integrating the current context and the user's long term preferences. This model takes the advantages of traditional collaborative filtering approaches (CF). It also captures the interaction between contextual information and item characteristics. The experimental results demonstrate that context is an important factor that affects user choices. If properly used, contextual information helps ranking based recommendation systems, probably because context influences users' purchasing decisions. Besides, more accurate contextual information leads to better recommendation models. However, contextual information does not help the user rating prediction task significantly, probably because context doesn't matter much given the user has already chosen a restaurant.

As the first step towards using the information extraction techniques to help contextual recommendation, the techniques used in this paper are far from optimal. In the future, we will research more effective text mining techniques for contextual extraction(Mazur and Dale, 2008; McCallum et al., 2000; Lafferty et al., 2001) at the same time increasing the amount of annotated review data for better classifier performance through actively learning (Laws and Schütze, 2008). We also plan to work towards a better understanding of contextual information in recommender systems, and explore other types of contextual information in different types of recommendation tasks besides restaurant recommendations.

[3]We sample 300 reviews for "Time" and "Companion" evaluation. Due to the extremely low probability of occurrence of Occasion context, we futher sample 200 reviews containing Occasion-related expressions and only evaluate extraction accuracy on these samples

7 Acknowledgements

Part of this research is funded by National Science Foundation IIS-0713111 and the Institute of Education Science. Any opinions, findings, conclusions or recommendations expressed in this paper are the authors', and do not necessarily reflect those of the sponsors. Bingqing Wang's work is done during his stay in the Research and Technology Center, Robert Bosch LLC.

References

Adomavicius, Gediminas and Francesco Ricci. 2009. Recsys'09 workshop 3: workshop on context-aware recommender systems, cars-2009. In *Proceedings of the 3rd ACM Conference on Recommender Systems, RecSys 2009*, pages 423–424.

Adomavicius, Gediminas and Alexander Tuzhilin. 2008. Context-aware recommender systems. In *Proceedings of the 2nd ACM Conference on Recommender Systems, RecSys 2008*, pages 335–336.

Adomavicius, Gediminas, Ramesh Sankaranarayanan, Shahana Sen, and Alexander Tuzhilin. 2005. Incorporating contextual information in recommender systems using a multidimensional approach. *ACM Transactions on Information Systems (TOIS)*, 23(1):103–145.

Chu, Wei and Seung-Taek Park. 2009. Personalized recommendation on dynamic content using predictive bilinear models. In *Proceedings of the 18th International Conference on World Wide Web, WWW 2009*, pages 691–700.

Cunningham, Hamish, Diana Maynard, Kalina Bontcheva, and Valentin Tablan. 2002. A framework and graphical development environment for

robust nlp tools and applications. In *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics, ACL 2002*, pages 168–175.

Koren, Yehuda. 2008. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, SIGKDD 2008*, pages 426–434.

Lafferty, John D., Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning, ICML 2001*, pages 282–289.

Laws, Florian and Hinrich Schütze. 2008. Stopping criteria for active learning of named entity recognition. In *Proceedings of the 22nd International Conference on Computational Linguistics, Coling 2008*, pages 465–472, August.

Lee, Hong Joo, Joon Yeon Choi, and Sung Joo Park. 2005. Context-aware recommendations on the mobile web. In *On the Move to Meaningful Internet Systems 2005: OTM 2005 Workshops*, pages 142–151.

Mazur, Pawel and Robert Dale. 2008. What's the date? high accuracy interpretation of weekday names. In *Proceedings of the 22nd International Conference on Computational Linguistics, Coling 2008*, pages 553–560.

McCallum, Andrew, Dayne Freitag, and Fernando C. N. Pereira. 2000. Maximum entropy markov models for information extraction and segmentation. In *Proceedings of the 17th International Conference on Machine Learning, ICML 2000*, pages 591–598.

Oku, Kenta, Shinsuke Nakajima, Jun Miyazaki, and Shunsuke Uemura. 2007. Investigation for designing of context-aware recommendation system using svm. In *Proceedings of the International MultiConference of Engineers and Computer Scientists 2007, IMECS 2007*, pages 970–975.

Ratnaparkhi, A. 1998. *MAXIMUM ENTROPY MODELS FOR NATURAL LANGUAGE AMBIGUITY RESOLUTION*. Ph.D. thesis, University of Pennsylvania.

Salakhutdinov, Ruslan and Andriy Mnih. 2007. Probabilistic matrix factorization. In *Advances in Neural Information Processing Systems 20, Proceedings of the 21st Annual Conference on Neural Information Processing Systems, NIPS 2007*, pages 1257–1264.

Reexamination on Potential for Personalization in Web Search

Daren Li[1] Muyun Yang[1] Haoliang Qi[2] Sheng Li[1] Tiejun Zhao[1]

[1]School of Computer Science
Harbin Institute of Technology
{drli|ymy|tjzhao}@mtlab.hit.edu.cn, lisheng@hit.edu.cn

[2]School of Computer Science
Heilongjiang Institute of Technology
haoliang.qi@gmail.com

Abstract

Various strategies have been proposed to enhance web search through utilizing individual user information. However, considering the well acknowledged recurring queries and repetitive clicks among users, it is still an open issue whether using individual user information is a proper direction of efforts in improving the web search. In this paper, we first quantitatively demonstrate that individual user information is more beneficial than common user information. Then we statistically compare the benefit of individual and common user information through Kappa statistic. Finally, we calculate potential for personalization to present an overview of what queries can benefit more from individual user information. All these analyses are conducted on both English AOL log and Chinese Sogou log, and a bilingual perspective statistics consistently confirms our findings.

1 Introduction

Most of traditional search engines are designed to return identical result to the same query even for different users. However, it has been found that majority of queries are quite ambiguous (Cronen-Townsend et al., 2002) as well as too short (Silverstein et al., 1999) to describe the exact informational needs of users. Different users may have completely different information needs under the same query (Jansen et al., 2000). For example, when users issue a query "Java" to a search engine, their needs can be something ranging from a programming language to a kind of coffee.

In order to solve this problem, personalized search is proposed, which is a typical strategy of utilizing individual user information. Pitkow et al. (2002) describe personalized search as the contextual computing approach which focuses on understanding the information consumption patterns of each user, the various information foraging strategies and applications they employ, and the nature of the information itself. After that, personalized search has gradually developed into one of the hot topics in information retrieval. As for various personalization models proposed recently, Dou et al. (2007), however, reveal that they actually harms the results for certain queries while improving others. This result based on a large-scale experiment challenges not only the current personalization methods but also the motivation to improve web search by the personalized strategies.

In addition, the studies on query logs recorded by search engines consistently report the prevailing repeated query submissions by large number of users (Silverstein et al., 1999; Spink et al., 2001). It is reported that the 25 most frequent queries from the AltaVista cover 1.5% of the total query submissions, despite being only 0.00000016% of unique queries (Silverstein et al., 1999). As a result, the previous users' activities may serve as valuable information, and technologies focusing on common

Coling 2008: Poster Volume, pages 701–709,
Beijing, August 2010

user information, such as collaborative filtering (or recommendation) may be a better resolution to web search. Therefore, the justification of utilizing individual user information deserves further discussion.

To address this issue, this paper conducts a bilingual perspective of survey on two large-scale query logs publically available: the AOL in English and the Sogou[1] in Chinese. First we quantitatively investigate the evidences for exploiting common user information and individual user information in these two logs. After that we introduce Kappa statistic to measure the consistency of users' implicit relevance judgment inferred from clicks. It is tentatively revealed that using individual user information is what requires web search to face with after common user information is well exploited. Finally, we study the distribution of potential for personalization over the whole logs to generally disclose what kind of query deserves for individual user information.

The remainder of this paper is structured as follows. Section 2 introduces previous methods employing individual and common user information. In Section 3, we quantitatively compare the evidences for exploiting common user information and individual user information. In Section 4, we introduce Kappa statistic to measure the consistency of users' clicks on the same query and try to statistically present the development direction of current web search. Section 5 figures out utilizing individual user information as a research issue after well exploiting common user information. Section 6 presents the potential for personalization curve, trying to outline which kind of queries benefit the most from individual user information. Conclusions and future work are detailed in Section 7.

2 Related Work

With the rapid expansion of World Wide Web, it becomes more and more difficult to find relevant information through one-size-fits-all information retrieval service provided by classical search engines. Two kinds of user information are mainly used to enhance search engines: common user information and individual user information. We separately review the previous works focusing on using these two kinds of information.

Among various attempts to improve the performance of search engine, collaborative web search is the one to take advantage of the repetition of users' behaviors, which we call common user information. Since there is no unified definition on collaborative web search, in this paper, we believe that the collaborative web search assumes that community search activities can provide valuable search knowledge, and sharing this knowledge facilitates improving traditional search engine results (Smyth, 2007). An important technique of collaborative web search is Collaborative Filtering (CF, also known as collaborative recommendation), in which, items are recommended to an active user based on historical co-occurrence data between users and items (Herlocker et al., 1999). A number of researchers have explored algorithms for collaborative filtering and the algorithms can be categorized into two classes: memory-based CF and model-based CF. Memory-based CF methods apply a nearest-neighbor-like scheme to predict a user's ratings based on the ratings given by like-minded users (Yu et al., 2004). The model-based approaches expand memory-based CF to build a descriptive model of group-based user preferences and use the model to predict the ratings. Examples of model-based approaches include clustering models (Kohrs et al., 1999) and aspect models (J. Canny, 2002).

The other way to improve web search is personalized web search, focusing on learning the individual preferences instead of others' behaviors, which is called individual user information. Early works learn user profiles from the explicit description of users to filter search results (Chirita et al., 2005). However, most of users are not willing to provide explicit feedback on search results and describe their interests (Carroll et al., 1987). Therefore, recent researches on the personalized search focus on modeling user preference from different types of implicit data, such as query history (Speretta et al., 2005), browsing history (Sugiyama et al., 2004), clickthrough data (Sun et al., 2005), immediate search context (Shen et al., 2005) and other personal information (Teevan et al.,

[1] A famous Chinese search engine with a large number of Chinese web search users.

2005). So far, there is still no proper comparison between the two solutions. It is still an open question which kind of information is more effective to build the web search model.

Considering the difficulty in collecting private information, using individual user information seems less promising as the cost-effective solution to web search. To address this issue, some researches about the value of personalization have been conducted. Teevan et al. (2007) have done a ground breaking job to quantify the benefit for the search engines if search results were tailored to satisfy each user. The possible improvement by the personalized search, named potential for personalization, is measured by a gap between the relevance of individualized rankings and group ranking based on NDCG. However, it is less touched for the position of individual user information in contrast with common user information in large scale query log and how to balance the usage of common and individual information in information retrieval model.

This paper tentatively examines individual user information against common user information on two large-scale search engine logs in following aspects: the evidence from clicks on the same query, Kappa statistic for the whole queries, and overall distribution of queries in terms of number of submissions and Kappa value. The bilingual statistics consistently reveals the tendency of using individual user information as an equally important issue as (if not more than) using common user information) issue for researches on web search.

3 Quantitative Evidences for Using Common or Individual User Information

To quantitatively investigate the value of common user information and individual user information in query log, we discriminate the evidence for using the two different types of user information as follows:

(1) Evidence for using common user information: if there were multiple users who have exactly the same click sets on one query, we suppose those clicks sets, together with the query, as the evidence for exploiting common user information. It is clear that such queries are able to be better responded with other's search results. Note that common user information is hard to be clearly defined, in order to simplify the quantitative statistics we give a strict definition. Further analysis will be shown in following sections.

(2) Evidence for using individual user information: if a user's click set on a query was not the same as any other's, for that query, the search intent of the user who issue that query can be better inferred from his/her individual information than common user information. We suppose this kind of clicks, together with the related queries, as the evidence for exploiting individual user information.

Since users may have different search intents when they issue the same query, a query can be an evidence for using both common and individual user information. In our statistics, if a query has both duplicate click sets and unique click set, the query is not only counted by the first category but also the second category.

The statistics of the two categories are conducted in the query log of both English and Chinese search engines. We use a subset of AOL Query Log from March 1, 2006 to May 31, 2006 and Sogou Query Log from March 1, 2007 to March 31, 2007. The basic statistics of AOL and Sogou log are shown in Table 1. Notice that the queries in raw AOL and Sogou log without clicks are removed in this study.

Item	AOL	Sogou
#days	92	31
#users	6,614,960	7,488,754
#queries	7,840,348	8,019,229
#unique queries	4,811,649	4,580,836
#clicks	12,984,610	17,607,808

Table 1: Basic statistics of AOL & Sogou log

Table 2 summarizes the statistics of different evidence categories over AOL and Sogou log. Note that click set refers to the set of clicks related to a query submission instead of a unique query. As for evidence for using common and individual user information, there is no clear distinction in terms of number of records, number of users in two logs. However, in terms of unique query and distinct click set, one can't fail to find that evidence for using individual user information clearly exceeds

Log	The Condition		Number			
	Repeated queries	Click	Records	User	Unique Query	Distinct Click Set
AOL	3,745,088 (47.77% of total query submissions)	Same	2,438,284	277,416	382,267	461,460
		Different	2,563,245	343,846	542,593	1,349,892
Sogou	4,252,167 (53.02% of total query submissions)	Same	2,469,363	1,380,951	228,315	358,346
		Different	5,481,832	1,545,817	752,047	2,171,872

Table 2: Different click behaviors on repeated queries

that for using common user information, especially in Sogou log. Therefore, though making use of common and individual user information can address equally well for half users and half visits to the search engine, the fact that much more unique queries and click sets actually claims the significance of needing individual user information to personalize web results. And methods exploiting individual user information provide a much more challenging task in terms of problem space, though one may argue utilizing common user information is much easier to attack.

4 Kappa Statistics for Individual and Common user information

Section 3 has shown the evidence for using individual user information is prevailing than common user information in quantity for the unique queries in search engines. However, these counts deserve a further statistical characterization. In this section, we introduce Kappa statistic to depict the overall consistency of users' clicks in query logs.

4.1 Kappa

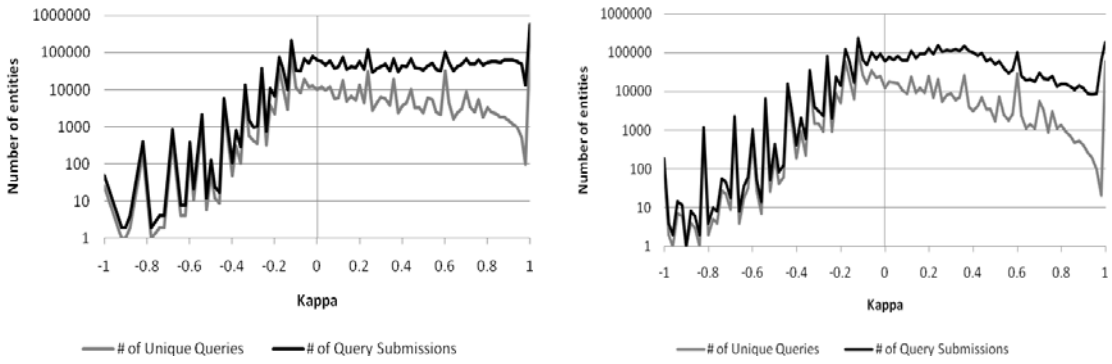
Kappa is a statistical measure introduced to access the agreement among different raters. There are two types of Kappa. One is Cohen's Kappa (Cohen, 1960), which measures only the degree of agreement between two raters. The other is Fleiss's Kappa (Fleiss, 1971), which generalizes Cohen's Kappa to measure agreement among more than two raters, denoted as:

$$\kappa = \frac{\overline{P} - \overline{P_e}}{1 - \overline{P_e}}$$

where, $\overline{P}$ is the probability that a randomly selected rater agree with another on a randomly selected subject. $\overline{P_e}$ is the expected probability of agreement if all raters made ratings by chance. If we use Kappa to measure the consistency of relevance judgment by different raters, $\overline{P}$ can be interpreted as the probability that two random selected raters consistently rate a random selected search result as relevant or non-relevant one. Similarly, $\overline{P_e}$ can also be construed as the expected probability of identical relevance judgment rated by different raters all by chance.

Teevan et al. (2008) used Fleiss's Kappa to measure the inter-rater reliability of different raters' explicit relevance judgments. We expand their work and employ Fleiss's Kappa to measure the consistency of implicit relevance judgments by users on the same query[2]. Here clicks are treated as a proxy for relevance: documents clicked by a user are judged as relevant and those not clicked as non-relevant (Teevan et al., 2008). As we all know that the result set of one query may change over time, so we select the longest time span to calculate Kappa value of a query, during which the result set of it preserves unchanged. From Kappa value of each query, we can statistically interpret to which extent users share consistent intent on the same query according to Table 3 (Landis and Koch, 1977). Though the interpretation in Table 3 is not accepted with no doubt, it can give us an intuition about what extent of agreement consistency is. In other words, Kappa is a measure with statistical sense. Meanwhile, Kappa values of queries with

[2] There may be more than two users who submitted the same query.



(a). AOL **(b). Sogou**

Figure 1: Number of unique queries and query submissions as a function of Kappa value.

κ	Interpretation
< 0	No agreement
0.0 – 0.20	Slight agreement
0.21 – 0.40	Fair agreement
0.41 – 0.60	Moderate agreement
0.61 – 0.80	Substantial agreement
0.81 – 1.00	Almost perfect agreement

Table 3: Kappa Interpretation

various sizes of click sets are also comparable. That is also the reason we choose Kappa to measure consistency.

4.2 Distribution of Kappa

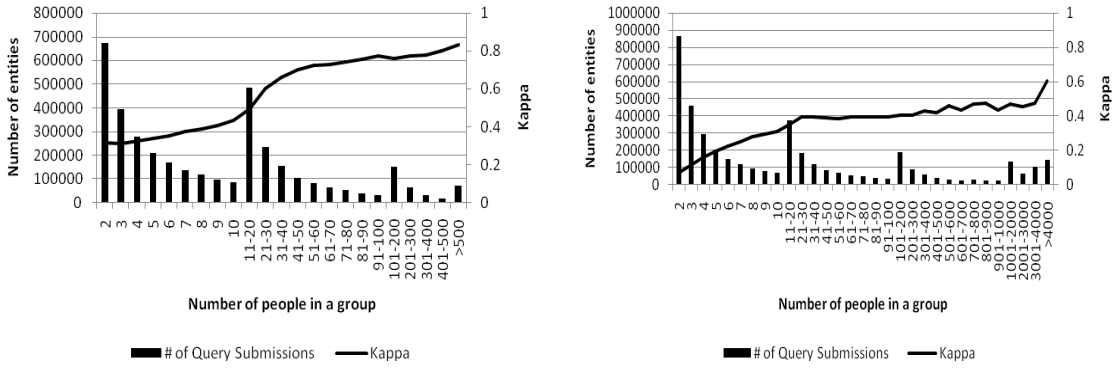
As introduced in Section 2, common user information is supposed to be the repetition of users' behaviors. We consider that the amount of repetition of users' clicks on one query is quantified by the consistency of its clicks. To statistically present the scale of repetition in current query log, we try to give an overview of consistency level of two commercial query logs.

Figure 1 plots distribution of Kappa value of the two logs in the coordinate with logarithmic Y-axis. About 34.5% unique queries (44.0% query submissions) in AOL log and only 13.9% unique queries (15.2% query submissions) in Sogou log have high Kappa values above 0.6. According to Table 3, click sets of these queries can be regarded as somewhat consistent. These queries can be roughly resolved by using common user information. On the other hand, for the rest of queries which constitute majority of the logs, users' click sets are rather diversified, which are hard to be satisfied by returning the same result list to them. As a whole, the queries in both AOL and Sogou can be characterized as less consistently in the clicks according to Kappa value, which is a statistical support for exploiting individual user information.

5 Individual or Common user information: A Tendency View

The above analyses quantitative analyses have shown that the repetition of search is not the statistically dominant factor, with the impression that employing individual user information is equally, if not more, important than common user information. This section tries to further reveal this issue so as to balance the position of individual user information and common user information from a research point.

Intuitively, a query can be characterized by the number of people issuing it, i.e. query frequency if we remove the resubmissions of one query by the same people. We try to depict the above mentioned query submissions and Kappa values as a function of number of people who issue the queries in Figure 2. In Figure 2, different numbers of users who issue the same query are shown on the x-axis, and the y-axis represents the number of different entities (left scale) and the average Kappa value (right scale) of the queries. We find that the number of queries becomes very small when the number of users in a group grows over 10, so we set a variant step length for them: with the length step of the group size falling between 2 and 10 set as 1, between 11 and 100 as 10, between 101 and 1000 as 100 and above 1000 as 1000.



(a). AOL **(b). Sogou**

Figure 2: Average Kappa value of queries as a function of number of people in a group who issue the same query (line) and the number of submissions of the queries issued by the same size of group (dark columns).

According to Figure 2(a), Kappa values of the queries in AOL log with more than 20 users are above 0.6, which indicates rather consistent clicks for them, accounting for about 29.4% of all query submissions. While for those queries visited by less than 20 users, the Kappa value declines gradually from 0.6 with the drop of users. For these queries occupying majority of query submissions, exploiting individual user information is supposed to be a better solution since the clicks on them are rather individualized.

According to Figure 2(b), though Kappa values of queries increase similarly with people submitting them in AOL, the overall consistency of the queries in Sogou log is much lower: with a Kappa value below 0.6 even for the queries visited by a large number of users. This fact indicates that Chinese users may be less consistent in their search intents, or partially reflects that the Chinese as a non-inflection language has more ambiguity, which can also be implied from Table 2. Therefore, individual user information may be more effective than common user information in Sogou log.

Summarized from Figure 2, it is sensible that common user information is appropriate for the queries in the right-most of X-axis. With most number of visiting people, such queries bear rather consistent clicks though covering only a small proportion of the distinct query set. Moving from the right to the left, we can find the majority of queries yield a less Kappa value, for which the individualized clicks require individual user information to meet the needs of each user. In this sense, how to exploit individual user information is predestined as the next issue of information retrieval if common user information was to be well utilized.

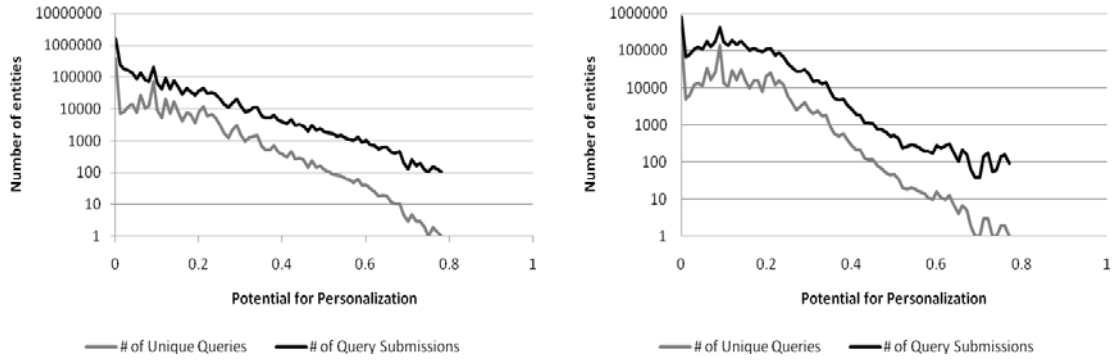
6 Queries for Personalization

Since using individual user information is a non-negligible issue in IR research, a subsequent issue is what queries can benefit in what extent from individual user information. In this section, we try to give an overview for this issue via a measure named potential for personalization.

6.1 Potential for Personalization

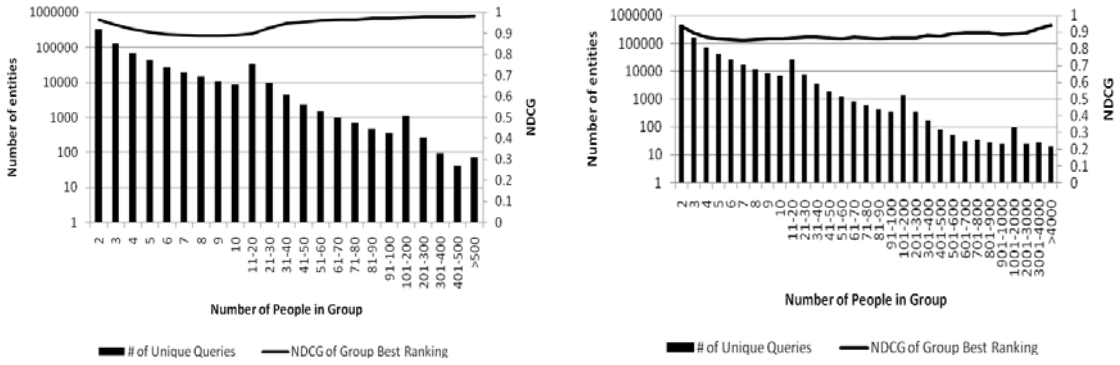
Potential for personalization proposed by Teevan et al. (2007) is used to measure the normalized Discounted Cumulative Gain (NDCG) improvement between the best ranking of the results to a group and individuals. NDCG is a well-known measure of the quality of a search result (Järvelin and Kekäläinen, 2000).

The best ranking of the results to a group is the ranking with highest NDCG based on relevance judgments of the users in the group. For the queries with explicit judgments, the best ranking can be generated as follows: results that all raters thought were relevant are ranked first, followed by those that most people thought were relevant but a few people thought were irrelevant, until the results most people though were irrelevant. In other word,



(a) AOL **(b) Sogou**

Figure 3: Number of unique queries and query submissions as a function of potential for personalization



(a) AOL **(b) Sogou**

Figure 4: The average NDCG of group best ranking as a function of number of people in group (solid line), combining with the distribution of the number of unique queries issued by the same size of group (dark columns)

the best ranking always tries to put the results that have the highest collective gain first to get the highest NDCG.

The previous work has shown that the implicit click-based potential for personalization is strongly related to variation in explicit judgments (J. Teevan et al., 2008). In this paper, we continue using click-based potential for personalization to measure the variation. Assuming the clicked results as relevant, we can calculate the potential for personalization of each query over the web search query log to present what kind of query can benefit more from personalization.

6.2 Potential for Personalization Distribution over Query Logs

Teevan et al. (2007) have depicted a potential for personalization curve based on explicit judgment to characterize the benefit that could be obtained by personalizing search results for each user. We continue using potential for personalization based on click-through to roughly reveal what kind of query can benefit more from personalization.

First we investigate the number of unique queries with different potential for personalization, which is shown in Figure 3. We find that there are about 53.9% unique queries in AOL log and 32.4% unique queries in Sogou log, whose potential for personalization is 0. For these queries, current web search is able to return perfect results to all users. However, for the rest of queries, even the best group ranking of results can't satisfy everyone who issues the query. So these queries should be better served by individual user information, covering

46.1% unique queries in AOL and 67.6% in Sogou.

Then, in order to further interpret what kind of query individual user information is needed most, we further relate potential for personalization to the number of users who submit the queries over AOL and Sogou query log as shown in Figure 4. For clarity's sake, we also set the same step length as in Figure 2.

According to Figure 4, the curve of potential for personalization is approximately U-shaped in both AOL log and Sogou Log. As the number of users in one group increases, performance of the best non-personalized rankings first declines, then flattens out and finally promotes[3]. Note that the left part of the curve is very similar to what Teevan et al. (2007) showed in their work.

Again in Figure 4, the queries which have the most potential for personalization are the ones which are issued by more than 6 and less than 20 users in AOL log. While in Sogou log, the queries issued by more than 6 and less than 4000 users have the most potential for personalization. Such different findings are probably caused by the content of query. There are many recommended queries in the homepage of Sogou search engine, most of which are informational query and clicked by a large number of users. Even when the size of group who issue the same query becomes very big, the query still has a wide variation of users' behaviors. So the consistency level of queries in Sogou log is much lower than the queries in AOL log at the same size of group.

7 Conclusion and Future Work

In this paper, we try to justify the position of individual user information comparing with common user information. It is shown that exploiting individual user information is a non-trivial issue challenging the IR community through the analysis of both English and Chinese large scale search logs.

We first classify the repetitive queries into 2 categories according to whether the corresponding clicks are unique among different users. We find that quantitatively the queries and clicks deserving for individual user information is much bigger than those deserving for common user information.

After that we use Kappa statistic to present that the overall consistency of query clicks recoded in search logs is pretty low, which statistically reveals that the repetition is not the dominant factor and individual user information is more desired to enhance most queries in current query log.

We also explore the distribution of Kappa values over different numbers of users in the group who issue the same query, concluding that how to utilize individual user information to improve the performance of web search engine is the next research issue confronted by the IR community when the repeated search of users are properly exploited.

Finally, potential for personalization is calculated over the two query logs to present an overview of what kind of queries that the optimal group-based retrieval model fails, which is supposed to benefit most from individual user information.

One possible enrichment to this work may come from the employment of content analysis based on text processing techniques. The different clicks, which are the basis of our examination, may have similar or even exact content in their web pages. Though the manual check for a small scale sampling from the Sogou log yields less than 1% probability for such case, the content based examination will be definitely more convincing than simple click counts. In addition, the queries for the two types of user information are not examined for their contents or the related information needs. Content analysis or linguistic view to these queries would be more informative. Both of these issues are to be addressed in our future work.

Acknowledgement

This work is supported by the Key Project of Natural Science Foundation of China (Grant No.60736044), and National 863 Project (Grant No.2006AA010108). The authors are grateful for the anonymous reviewers for their valuable comments.

[3] Note that the different step length dims the actual U-shape in the figure.

References

Canny John. 2002. Collaborative filtering with privacy via factor analysis. In *Proceedings of SIGIR' 02*, pages 45-57.

Carroll M. John and Mary B. Rosson. 1987. Paradox of the active user. *Interfacing thought: cognitive aspect of human-computer interaction*, pages 80-111.

Chirita A. Paul, Wofgang Nejdl, Raluca Paiu, and Christian Kohlschutter. 2005. Using odp metadata to personalize search. In *Proceedings of SIGIR '05*, pages 178-185.

Cohen Jacob. 1960. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20: 37-46

Dou Zhicheng, Ruihua Song, and Ju-Rong Wen. 2007. A Large-scale Evaluation and Analysis of Personalized Search Strategies. In *Proceedings of WWW '07*, pages 581-590.

Fleiss L. Joseph. 1971. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5): 378-382.

Herlocker L. Jonathan, Joseph A. Konstan, Al Borchers, and John Riedl. 1999. An algorithmic framework for performing collaborative filtering. In *Proceedings of SIGIR '99*, pages 230-237.

Jansen J. Bernard, Amanda Spink, and Tefko Saracevic. 2000. Real life, real users, and real needs: a study and analysis of user queries on the web. *Information Processing and Management*, pages 207-227.

Järvelin Kalervo and Jaana Kekäläinen. 2000. IR evaluation methods for retrieving highly relevant documents. In *Proceedings of SIGIR '00*, pages 41-48.

Kohrs Arnd and Bernard Merialdo. 1999. Clustering for collaborative filtering applications. In *Proceedings of CIMCA '99*, pages 199-204.

Landis J. Richard and Gary. G. Koch. 1977. The mea-surement of observer agreement for categorical data. *Biometrics* 33: 159-174.

Pitkow James, Hinrich Schutze, Todd Cass, Rob Cooley, Don Turnbull, Andy Edmonds, Eytan Adar and Thomas Breuel. 2002. Personalized search. *ACM*, 45(9):50-55.

Shen Xuehua, Bin Tan and ChengXiang Zhai. 2005 Implicit user modeling for personalized search. In *Proceedings of CIKM '05*, pages 824-831.

Silverstein Craig, Monika Henzinger, Hannes Marais and Michael Moricz. 1999. Analysis of a very large web search engine query log. *SIGIR Forum*, 33(1):6-12.

Smyth Barry. 2007. A Community-Based Approach to Personalizing Web Search. *IEEE Computer*, 40(8): 42-50.

Speretta Mirco and Susan Gauch. Personalized Search based on user search histories. 2005. In *Proceedings of WI '05*, pages 622-628.

Spink Amanda, Dietmar Wolfram, Major Jansen, Tefko Saracevic. 2001. Searching the web: The public and their queries. *Journal of the American Society for Information Science and Technology*, 52(3), 226-234

Sugiyama Kazunari, Kenji Hatano, and Masatoshi Yoshikawa. 2004. Adaptive web search based on user profile constructed without any effort from users. In *Proceedings of WWW '04*, pages 675-684.

Sun Jian-Tao, Hua-Jun Zeng, Huan Liu, Yuchang Lu and Zheng Chen. 2005. CubeSVD: a novel approach to personalized web search. In *Proceedings of WWW'05*, pages 382-390.

Teevan Jaime, Susan T. Dumais, and Eric Horvitz. 2005. Personalizing search via automated analysis of interests and activities. In *Proceedings of SIGIR '05*, pages 449-456.

Teevan Jaime, Susan T. Dumais and Eric Horvitz. 2007. Characterizing the value of personalizing search. In *Proceedings of SIGIR '07*, pages 757-758.

Teevan Jaime, Susan T. Dumais and Daniel J. Liebling. 2008. To personalize or Not to Personalize: Modeling Queries with Variation in User Intent. In *Proceedings of SIGIR '08*, pages 163-170.

Townsend Steve Cronen and W. Bruce Croft. 2002. Quantifying query ambiguity. In *Proceedings of HLT '02*, pages 613-622.

Yu Kai, Anton Schwaighofer, Volker Tresp, Xiaowei Xu, Hans-Peter Kriegel. 2004. Probabilistic Memory-based Collaborative Filtering. In *IEEE Transactions on Knowledge and Data Engineering*, pages 56-59.

Fast-Champollion: A Fast and Robust Sentence Alignment Algorithm

Peng Li and **Maosong Sun**
Department of Computer Science and Technology
State Key Lab on Intelligent Technology and Systems
National Lab for Information Science and Technology
pengli09@gmail.com, sms@tsinghua.edu.cn

Ping Xue
The Boeing Company
ping.xue@boeing.com

Abstract

Sentence-level aligned parallel texts are important resources for a number of natural language processing (NLP) tasks and applications such as statistical machine translation and cross-language information retrieval. With the rapid growth of online parallel texts, efficient and robust sentence alignment algorithms become increasingly important. In this paper, we propose a fast and robust sentence alignment algorithm, i.e., Fast-Champollion, which employs a combination of both length-based and lexicon-based algorithm. By optimizing the process of splitting the input bilingual texts into small fragments for alignment, Fast-Champollion, as our extensive experiments show, is 4.0 to 5.1 times as fast as the current baseline methods such as Champollion (Ma, 2006) on short texts and achieves about 39.4 times as fast on long texts, and Fast-Champollion is as robust as Champollion.

1 Introduction

Sentence level aligned parallel corpora are very important resources for NLP tasks including machine translation, cross-language information retrieval and so on. These tasks typically require support by large aligned corpora. In general, the more aligned text we have, the better result we achieve. Although there is a huge amount of bilingual text on the Internet, most of them are either only aligned at article level or even not aligned at all. Sentence alignment is a process mapping sentences in the source text to their corresponding units in the translated text. Manual sentence alignment operation is both expensive and time-consuming, and thus automated sentence alignment techniques are necessary. A sentence alignment algorithm for practical use should be (1) fast enough to process large corpora, (2) robust enough to tackle noise commonly present in the real data, and (3) effective enough to make as few mistakes as possible.

Various sentence alignment algorithms have been proposed, which generally fall into three types: length-based, lexicon-based, and the hybrid of the above two types. Length-based algorithms align sentences according to their length (measured by character or word). The first length-based algorithm was proposed in (Brown et al., 1991). This algorithm is fast and has a good performance if there is minimal noise (e.g., sentence or paragraph omission) in the input bilingual texts. As this algorithm does not use any lexical information, it is not robust. Lexicon-based algorithms are usually more robust than the length-based algorithm, because they use the lexical information from source and translation lexicons instead of solely sentence length to determine the translation relationship between sentences in the source text and the target text. However, lexicon-based algorithms are slower than length-based sentence alignment algorithms, because they require much more expensive computation. Typical lexicon-based algorithms include (Ma, 2006; Chen, 1993; Utsuro et al., 1994; Melamed, 1996). Sentence length and lexical information are also combined to achieve more efficient algorithms in two ways. One way is to use both sentence length and lex-

Coling 2008: Poster Volume, pages 710–718,
Beijing, August 2010

ical information together to determine whether two sentences should be directly aligned or not (Simard et al., 1993; Wu, 1994). The other way is to produce a rough alignment based on sentence length (and possibly some lexical information at the same time), and then build more precise alignment by using more effective lexicon-based algorithms (Moore, 2002; Varga et al., 2005). But both of the two ways suffer from high computational cost and are not fast enough for processing large corpora.

Lexical information is necessary for improving robustness of a sentence alignment algorithm, but use of lexical information will introduce higher computational cost and cause a lower speed. A common fact is that the shorter the text is, the less combination possibilities it would introduce and the less computational cost it would need. So if we can first split the input bilingual texts into small aligned fragments reliably with a reasonable amount of computational cost, and then further align these fragments one by one, we can speed up these algorithms remarkably. This is the main idea of our algorithm Fast-Champollion.

The rest of this paper is organized as follows: Section 2 presents formal definitions of sentence alignment problem, and briefly reviews the length-based sentence alignment algorithm and Champollion algorithm; Section 3 proposes the Fast-Champollion algorithm. Section 4 shows the experiment results; and Section 5 is the conclusion.

2 Definitions and Related Work

2.1 Definitions and Key Points

A **segment** is one or more consecutive sentence(s). A **fragment** consists of one segment of the source text (denoted by S) and one segment of the target text (denoted by T), and a fragment can be further divided into one or more beads. A **bead** represents a group of one or more sentences in the source text and the corresponding sentence(s) in the target text, denoted by $A_i = (S_{Ai}; T_{Ai}) = (S_{a_{i-1}+1}, S_{a_{i-1}+2}, \cdots, S_{a_i}; T_{b_{i-1}+1}, T_{b_{i-1}+2}, \cdots, T_{b_i})$, where S_i and T_j are the i$^{\text{th}}$ and j$^{\text{th}}$ sentence of S and T respectively.

In practice, we rarely encounter crossing alignment, e.g., sentences S_i and S_j of the source language are aligned to the sentences T_j and T_i of the target language respectively. But much more effort has to be taken for an algorithm to process crossing alignment well. So we do not consider crossing alignment here.

In addition, only a few type of beads are frequently observed in the real world. As it can save significantly in terms of computational cost and it would not do significant harm to algorithm without considering rare bead types, a common practice for designing sentence alignment algorithms is to only consider the frequently observed types of beads. Following this practice, we only consider beads of 1-to-0, 0-to-1, 1-to-1, 1-to-2, 2-to-1, 1-to-3, 3-to-1, 1-to-4, 4-to-1 and 2-to-2 types in our algorithm, where n-to-m means the bead consists of n sentence(s) of the source language and m sentence(s) of the target language.

2.2 Length-based Sentence Alignment Algorithm

Length-based sentence alignment algorithm was first proposed in (Brown et al., 1991). This algorithm captures the idea that long or short sentences tend to be translated into long or short sentences. A probability is produced for each bead based on the sentence length, and a dynamic programming algorithm is used to search for the alignment with the highest probability, which is treated as the best alignment.

This algorithm is fast and can produce good alignment when the input bilingual texts do not contain too much noise, but it is not robust, because it only uses the sentence length information. When there is too much noise in the input bilingual texts, sentence length information will be no longer reliable.

2.3 Champollion Aligner

Champollion aligner was proposed in (Ma, 2006). It borrows the idea of *tf-idf* value, which is widely used in information retrieval, to weight term[1] pair similarity. Greater weight is assigned to the less frequent translation term pairs, because these term

[1]Here terms are not limited to linguistic words, but also can be tokens like "QX6800"

pairs have much stronger evidence for two segments to be aligned. For any two segments, a similarity is assigned based on the term pair weight, sentence number and sentence length. And the dynamic programming algorithm is used to search for the alignment with the greatest total similarity. This alignment is treated as the best alignment.

Champollion aligner can produce good alignment even on noisy input as reported in (Ma, 2006). Its simplicity and robustness make it a good candidate for practical use. But this aligner is slow. Because its time complexity is $O(n^2)$ and it has to look up the dictionary multiple times in each step of the dynamic programming algorithm, which needs higher computational cost.

3 Fast-Champollion Algorithm

In this section we propose a new sentence alignment algorithm: Fast-Champollion. Its basis is splitting the input bilingual texts into small aligned fragments and then further aligning them one by one to reduce running time while maintaining Champollion-equivalent (or better) alignment quality; it takes the advantages of both length-based and lexicon-based algorithms to the maximum extent. The outline of the algorithm is that first the *length-based splitting module* is used to split the input bilingual texts into aligned fragments, and then the components of each of these fragments will be identified and aligned by a Champollion-based algorithm. The details are described in the following sections.

3.1 Length-based Splitting Module

Although length-based sentence alignment algorithm is not robust enough, it can produce rough alignment very fast with a certain number of reliably translated beads. Length-based splitting module is designed to select these reliably translated beads to be used for delimiting and splitting the input bilingual texts into fragments. These beads will be referred to as *anchor beads* in the remaining sections.

There are four steps in this module as described below in detail.

Step 1: decide whether to skip step 2-4 or not

When there is too much noise in the input bilingual texts, the percentage of reliably translated beads in the alignment produced by the length-based algorithm will be very low. In this case, we will skip step 2 through 4.

An evidence for such a situation is that the difference between the sentence numbers of the source and target language is too big. Suppose N_S and N_T are the number of sentences of the source and target language respectively. We specify $r = |N_S - N_T| / \min\{N_S, N_T\}$ as a measure of the difference, where min means minimum. If r is bigger than a threshold, we say the difference is too big. In our experiments, the threshold is set as 0.4 empirically.

Step 2: align the input texts using length-based algorithm

In this step, length-based sentence alignment algorithm is used to align the input bilingual texts. Brown, et al. (1991) models the process of sentence alignment as two steps. First, a bead is generated according to a fixed probability distribution over bead types, and then sentence length in the bead is generated according to this model: for the 0-to-1 and 1-to-0 type of beads, it is assumed that the sentence lengths are distributed according to a probability distribution estimated from the data. For other type of beads, the lengths of sentences of the source language are generated independently from the probability distribution for the 0-to-1 and 1-to-0 type of beads, and the total length of sentences of the target language is generated according to a probability distribution conditioned on the total length of sentences of the source language. For a bead $A_i = (S_{Ai}; T_{Ai})$, $l_{S_{Ai}}$ and $l_{T_{Ai}}$ are the total lengths of sentences in S_{Ai} and T_{Ai} respectively, which are measured by word[2]. Brown, et al. (1991) assumed this conditioned probability distribution is

$$Prob(l_{T_{Ai}}|l_{S_{Ai}}) = \alpha \exp\left(-\frac{(\lambda_i - \mu)^2}{2\sigma^2}\right),$$

where $\lambda_i = \log(l_{T_Ai}/l_{S_Ai})$ and α is a normalization factor. Moore (2002) assumed the condi-

[2]For Chinese, word segmentation should be done first to identify words.

tioned probability distribution is

$$Prob(l_{T_{Ai}}|l_{S_{Ai}}) = \frac{\exp(-l_{S_{Ai}}r)(l_{S_{Ai}}r)^{l_{T_{Ai}}}}{l_{T_{Ai}}!},$$

where r is the ratio of the mean length of sentences of the target language to the mean length of sentences of the source language. We tested the two models on our development corpus and the result shows that the first model performs better, so we choose the first one.

Step 3: determine the anchor beads

In this step, the reliably translated beads in the alignment produced by the length-based algorithm in Step 2 will be selected as anchor beads.

The length-based algorithm can generate a probability for each bead it produces. So a trivial way is to choose the beads with a probability above certain threshold as anchor beads. But as pointed out before, when there is too much noise, the alignment produced by the length-based algorithm is no longer reliable, and so is it with the probability. A fact is that if we select a non-translated bead as an anchor bead, we will split the input bilingual texts into wrong fragments and may cause many errors. So we have to make decision conservatively in this step and we decide to use lexical information instead of the probability to determine the anchor beads.

For a bead $A_i = (S_{Ai}; T_{Ai})$, the proportion of translation term-pairs is a good measure for determine whether this bead is reliably translated or not. In addition, use of *local information* will also be greatly helpful. To explain the use of "local information", let's define the fingerprint of a sentence first. Suppose we have a sequence of sentences $S_1, S_2, \cdots, S_m$, and $W(S_i)$is the set of distinct words in S_i, then the fingerprint of S_i is

$$f(S_i) = W(S_i) - W(S_{i-1}) - W(S_{i+1}),$$

and specially

$$f(S_1) = W(S_1) - W(S_2),$$

$$f(S_m) = W(S_m) - W(S_{m-1}).$$

The fingerprints of S_{Ai} and T_{Ai}, denoted by $f(S_{Ai})$ and $f(T_{Ai})$, are the unions of all the fingerprints of sentences in S_{Ai} and T_{Ai} respectively. As you can see, the fingerprint of a sentence is the set of words in the sentence that do not appear in the adjacent sentence(s), and thus can distinguish this sentence from its neighbors. So fingerprint is also a good measure. By combining these two measures together, we can select out more reliably translated beads.

For a word w, we use $d_D(w)$ to denote the set of all its translations in a bilingual dictionary D, and use $t_D(w)$ to denote the union of $\{w\}$ and $d_D(w)$, i.e., $t_D(w) = \{w\} \cup d_D(w)$. Given two sets of words A and B. We say a word w of A is translated by B if either one of its translations in the dictionary D or the word itself appears in B, i.e., $t_D(w) \cap B \neq \emptyset$. The set of all the words of A that are translated by B is:

$$h_D(A, B) = \{w \in A \; and \; t_D(w) \cap B \neq \emptyset\}.$$

Then the proportion of terms in A that are translated by B is

$$r_D(A, B) = \frac{|h_D(A, B)|}{|A|}.$$

We specify the proportion of translation term pairs in a bead, denoted as $ar_D(A_i)$, to be $\min\{r_D(W(S_{Ai}), W(T_{Ai})), r_D(W(T_{Ai}), W(S_{Ai}))\}$, where $W(S_{Ai})$ and $W(T_{Ai})$ are the sets of distinct words in S_{Ai} and T_{Ai} respectively. Also we specify the proportion of translation term-pairs in the fingerprint, denoted as $fr_D(A_i)$, to be $\min\{r_D(f(S_{Ai}), f(T_{Ai})), r_D(f(T_{Ai}), f(S_{Ai}))\}$. Given thresholds TH_{ar} and TH_{fr}, a bead is selected as an anchor bead when $ar_D(A_i)$ and $fr_D(A_i)$ are not smaller than TH_{ar} and TH_{fr} respectively. We will show that Fast-Champollion algorithm is not sensitive to TH_{ar} and TH_{fr} to some extent in Section 4.2.

Step 4: split the input bilingual texts

The anchor beads determined in Step 3 are used to split the input texts into fragments. The ending location of each anchor bead is regarded as a splitting point, resulting in two fragments.

3.2 Aligning Fragments with Champollion Aligner

The similarity function used by Champollion aligner is defined as follows. Given two (source

and target language) groups of sentences in a fragment, denoted by G_S=S_1, S_2,$\cdots$,S_m and G_T=T_1, T_2,$\cdots$,T_n, suppose there are k pairs of translated terms in G_S and G_T denoted by (ws_1, wt_1),(ws_2, wt_2),$\cdots$,(ws_k, wt_k), where ws_i is in G_S and wt_i is in G_T. For each pair of the translated terms (ws_i, wt_i), define $idtf(ws_i)$ to be

$$\frac{Total\ \#\ of\ terms\ in\ the\ whole\ document}{\#\ occurrences\ of\ ws_i\ in\ G_S},$$

and define

$$stf(ws_i, wt_i) = \min\{stf(ws_i), stf(wt_i)\},$$

where $stf(ws_i)$ and $stf(wt_i)$ are the frequency of ws_i and wt_i in G_S and G_T respectively. The similarity between G_S and G_T is defined as

$$\sum_{i=1}^{k} \log\left(idtf(ws_i) \times stf(ws_i, wt_i)\right) \times alignment\_penalty \times length\_penalty,$$

where $alignment\_penalty$ is 1 for 1-to-1 alignment type of beads and a number between 0 and 1 for other type of beads, $length\_penalty$ is a function of the total sentence lengths of G_S and G_T.

The reason for choosing Champollion aligner instead of other algorithms will be given in Section 4.2. And another question is how $idtf$ values should be calculated. $idtf$ is used to estimate how widely a term is used. An intuition is that $idtf$ will work better if the texts are longer, because if the texts are short, most words will have a low frequency and will seem to only appear locally. In Fast-Champollion, we calculate $idtf$ according to the whole document instead of each fragment. In this way, a better performance is achieved.

3.3 Parameter Estimation

A development corpus is used to estimate the parameters needed by Fast-Champollion.

For the length-based algorithm, there are five parameters that need to be estimated. The first one is the probability distribution over bead types. The ratio of different types of beads in the development corpus is used as the basis for the estimation. The second and third parameters are the probability distributions over the sentence length of the source language and the target language. These distributions are estimated as the distributions observed from the input bilingual texts. That is to say, these two distributions will not be the same for different bilingual input texts. The forth and fifth are μ and σ. They are estimated as the mean and variance of λ_i over the development corpus.

For Champollion aligner, $alignment\_penalty$ and $length\_penalty$ need to be determined. Because the Perl version of Champollion aligner[3] is well developed, we borrow the two definitions from it directly.

4 Experiments

4.1 Datasets and Evaluation Metrics

We have two English-Chinese parallel corpora, one for the development purpose and one for the testing purpose. Both of the two corpora are collected from the Internet and are manually aligned.

The development corpus has 2,004 beads. Given the space constraint, detailed information about the development corpus is omitted here.

The testing corpus contains 26 English-Chinese bilingual articles collected from the Internet, including news reports, novels, science articles, television documentary subtitles and the record of government meetings. There are 9,130 English sentences and 9,052 Chinese sentences in these articles[4]. The number of different type of beads in the golden standard answer is shown in Table 1.

Type	Number	Percentage(%)
1:1	7275	83.19
1:2 2:1	846	9.67
1:3 3:1	77	0.88
1:4 4:1	16	0.18
2:2	32	0.37
1:0 0:1	482	5.51
others	17	0.19
total	8745	100.00

Table 1: Types of beads in the golden standard

Both the Fast-Champollion algorithm and the Champollion aligner need a bilingual dictionary and we supply the same bidirectional dictionary to

[3] http://champollion.sourceforge.net

[4] The definition of "sentence" is slightly different from the common sense here. We also treat semicolon and colon as the end of a sentence.

them in the following evaluations. This dictionary contains 45,439 pair of English-Chinese translation terms.

We use four commonly used measures for evaluating the performance of a sentence alignment algorithm, which are the *running time*,

$$\textit{Precision} = \frac{|GB \cap PB|}{|PB|},$$

$$\textit{Recall} = \frac{|GB \cap PB|}{|GB|},$$

and

$$\textit{F1-measure} = \frac{2 \times Precision \times Recall}{Precision + Recall},$$

where GB is the set of beads in the golden standard, and PB is the set of beads produced by the algorithm.

All the following experiments are taken on a PC with an Intel QX6800 CPU and 8GB memory.

4.2 Algorithm Design Issues

Why Choose Champollion?

We compared Champollion aligner with two other sentence alignment algorithms which also make use of lexical information. And the result is shown in Table 2. "Moore-1-to-1" and "Moore-all" are corresponding to the algorithm proposed in (Moore, 2002). The difference between them is how Recall is calculated. Moore's algorithm can only output 1-to-1 type of beads. For "Moore-1-to-1", we only consider beads of 1-to-1 type in the golden standard when calculating Recall, but all types of beads are considered for "Moore-all". The result suggests that ignoring the beads that are not of 1-to-1 type does have much negative effect on the overall performance of Moore's algorithm. Our goal is to design a general purpose sentence alignment algorithm that can process frequently observed types of beads. So Moore's algorithm is not a good choice. Hunalign refers to the hunalign algorithm proposed in (Varga et al., 2005). The resources provided to Champollion aligner and hunalign algorithm are the same in the test, but hunalign algorithm's performance is much lower. So hunalign algorithm is not a good choice either. Champollion algorithm is simple and has a high overall performance. So it is a better choice for us.

Aligner	Precision	Recall	F1-measure
Champollion	0.9456	0.9546	0.9501
Moore-1-to-1	0.9529	0.9436	0.9482
Moore-all	0.9529	0.7680	0.8505
Hunalign	0.8813	0.9037	0.8923

Table 2: The performance of different aligners on the development corpus

The Effect of TH_{ar} and TH_{fr}

TH_{ar} and TH_{fr} are two thresholds for selecting anchor beads in Step 3 of length-based splitting module. In order to investigate the effect of these two thresholds on the performance of Fast-Champollion, we run Fast-Champollion on the development corpus with different TH_{ar} and TH_{fr}. Both TH_{ar} and TH_{fr} vary from 0 to 1 with step 0.05. And the running time and F1-measure are shown in Figure 1 and Figure 2 respectively.

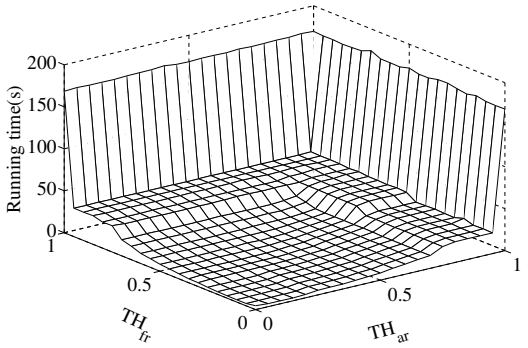


Figure 1: The running time corresponding to different TH_{ar} and TH_{fr}

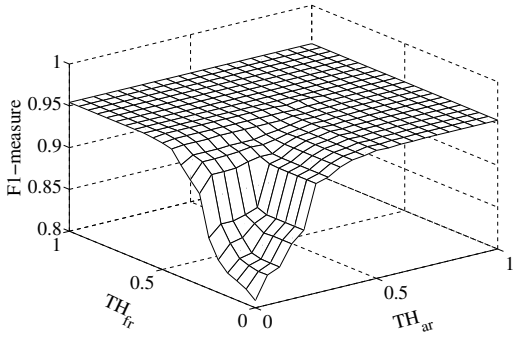


Figure 2: The F1-measure corresponding to different TH_{ar} and TH_{fr}

From Figure 1 and Figure 2, we see that for a large range of the possible values of TH_{ar} and TH_{fr}, the running time of Fast-Champollion increases slowly while F1-measure are nearly the same. In other words, Fast-Champollion are not sensitive to TH_{ar} and TH_{fr} to some extent. So making choice for the exact values of TH_{ar} and TH_{fr} becomes simple. And we use 0.5 for both of them in the following experiments.

4.3 Performance of Fast-Champollion

We use three baselines in the following evaluations. One is an implementation of the length-based algorithm in Java, one is a re-implemented Champollion aligner in Java according to the Perl version, and the last one is Fast-Champollion-Recal. Fast-Champollion-Recal is the same as Fast-Champollion except that it calculates $idtf$ values according to the fragments themselves independently instead of the whole document, and the Java versions of the length-based algorithm and Champollion aligner are used for evaluation.

Performance on Texts from the Internet

Table 3 shows the performance of Fast-Champollion and the baselines on the testing corpus. The result shows that Fast-Champollion achieves slightly better performance than Fast-Champollion-Recal. The running time of Champollion is about 2.6 times longer than Fast-Champollion with lower Precision, Recall and F1-measure. It should be pointed out that Fast-Champollion achieves better Precision, Recall and F1-measure than Champollion does because the splitting process may split the regions hard to align into different fragments and reduces the chance for making mistakes. Because of the noise in the corpus, the F1-measure of the length-based algorithm is low. This result suggests that Fast-Champollion is fast, robust and effective enough for aligning texts from the Internet.

Robustness of Fast-Champollion

In order to make a more precise investigation on the robustness of Fast-Champollion against noise, we made the following evaluation. First we manually removed all the 1-to-0 and 0-to-1 type of beads from the testing corpus to produce a clean corpus. This corpus contains 8,263 beads. Then we added 8263×n% 1-to-0 or 0-to-1 type of beads to this corpus at arbitrary positions to produce a series of noisy corpora, with n having the values of 5,10,...,100. Finally we ran Fast-Champollion algorithm and the baselines on these corpora respectively and the results are shown in Figure 3 and Figure 4, which indicate that for Fast-Champollion, when n increases 1, Precision drops 0.0021, Recall drops 0.0038 and F1-measure drops 0.0030 on average, which are very similar to those of Champollion, but Fast-Champollion is 4.0 to 5.1 times as fast as Champollion. This evaluation proves that Fast-Champollion is robust against noise and is a more reasonable choice for practical use.

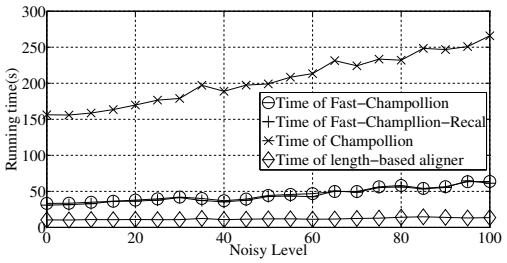


Figure 3: Running Time of Fast-Champollion, Fast-Champollion-Recal, Champollion and the length-based algorithm

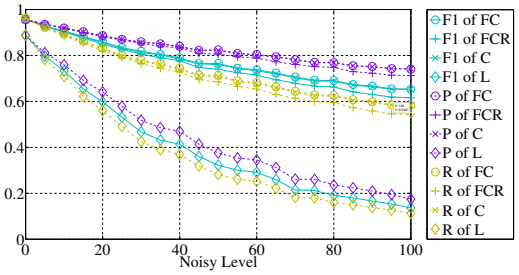


Figure 4: Precision (P), Recall (R) and F1-measure (F1) of Fast-Champollion (FC), Fast-Champollion-Recall (FCR), Champollion (C) and the length-base algorithm (L)

Performance on Long Texts

In order to test the scalability of Fast-Champollion algorithm, we evaluated it on long texts. We merged all the articles in the testing cor-

Aligner	Precision	Recall	F1-measure	Running time(s)
Fast-Champollion	0.9458	0.9408	0.9433	48.0
Fast-Champollion-Recal	0.9470	0.9373	0.9421	45.4
Champollion	0.9450	0.9385	0.9417	173.5
Length-based	0.8154	0.7878	0.8013	11.3

Table 3: Performance on texts from the Internet

Aligner	Precision	Recall	F1-measure	Running time(s)
Fast-Champollion	0.9457	0.9418	0.9437	51.5
Fast-Champollion-Recall	0.9456	0.9362	0.9409	50.7
Champollion	0.9464	0.9412	0.9438	2029.0
Length-based	0.8031	0.7729	0.7877	23.8

Table 4: Performance on long text

pus into a single long "article". Its length is comparable to that of the novel of Wuthering Heights. Table 4 shows the evaluation results on this long article. Fast-Champollion is about 39.4 times as fast as Champollion with slightly lower Precision, Recall and F1-measure, and is just about 1.2 times slower than the length-based algorithm, which has much lower Precision, Recall and F1-measure. So Fast-Champollion is also applicable for long text, and has a significantly higher speed.

4.4 Evaluation of the Length-based Splitting Module

The reason for Fast-Champollion can achieve relatively high speed is that the length-based splitting module can split the bilingual input texts into many small fragments reliably. We investigate the fragments produced by the length-based splitting module when aligning the long article used in Section 4.3. The length-based splitting module splits the long article at 1,993 places, and 1,972 segments are correct. The numbers of Chinese and English segments with no more than 30 Chinese and English sentences are shown in Figure 5. As there are only 27 and 29 segments with more than 30 sentences for Chinese and English respectively, we omit them in the figure. We can conclude that although the length-based splitting module is simple, it is efficient and reliable.

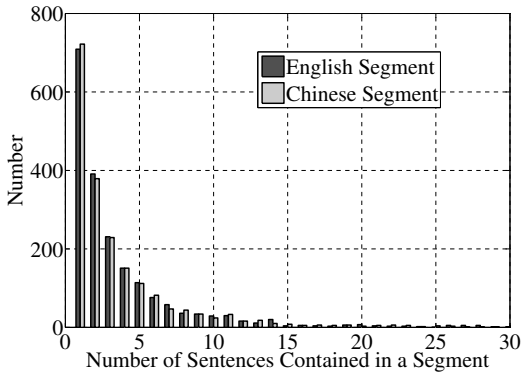


Figure 5: Numbers of Chinese/English segments with no more than 30 Chinese/English sentences

5 Conclusion and Future Work

In this paper we propose a new sentence alignment algorithm Fast-Champollion. It reduces the running time by first splitting the bilingual input texts into small aligned fragments and then further aligning them one by one. The evaluations show that Fast-Champollion is fast, robust and effective enough for practical use, especially for aligning large amount of bilingual texts or long bilingual texts.

Fast-Champollion needs a dictionary for aligning sentences, and shares the same problem of Champollion aligner as indicated in (Ma, 2006), that is the precision and recall will drop as the size of the dictionary decreases. So how to build bilingual dictionaries automatically is an important task for improving the performance of Fast-Champollion in practice, and is a critical problem for applying Fast-Champollion on language pairs without a ready to use dictionary.

Acknowledgement

This research is supported by the Boeing-Tsinghua Joint Research Project "Robust Chinese Word Segmentation and High Performance English-Chinese Bilingual Text Alignment".

References

Brown, Peter F., Jennifer C. Lai, and Robert L. Mercer. 1991. Aligning sentences in parallel corpora. In *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics*, pages 169–176, Berkeley, California, USA, June. Association for Computational Linguistics.

Chen, Stanley F. 1993. Aligning sentences in bilingual corpora using lexical information. In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*, pages 9–16, Columbus, Ohio, USA, June. Association for Computational Linguistics.

Ma, Xiaoyi. 2006. Champollion: A robust parallel text sentence aligner. In *Proceedings of LREC-2006: Fifth International Conference on Language Resources and Evaluation*, pages 489–492.

Melamed, I. Dan. 1996. A geometric approach to mapping bitext correspondence. In *Proceedings of the First Conference on Empirical Methods in Natural Language Processing*, pages 1–12.

Moore, Robert C. 2002. Fast and accurate sentence alignment of bilingual corpora. In *Proceedings of the 5th Conference of the Association for Machine Translation in the Americas on Machine Translation: From Research to Real Users*, pages 135–144, London, UK. Springer-Verlag.

Simard, Michel, George F. Foster, and Pierre Isabelle. 1993. Using cognates to align sentences in bilingual corpora. In *Proceedings of the 1993 Conference of the Centre for Advanced Studies on Collaborative Research*, pages 1071–1082. IBM Press.

Utsuro, Takehito, Hiroshi Ikeda, Masaya Yamane, Yuji Matsumoto, and Makoto Nagao. 1994. Bilingual text matching using bilingual dictionary and statistics. In *Proceedings of the 15th Conference on Computational Linguistics*, pages 1076–1082, Morristown, NJ, USA. Association for Computational Linguistics.

Varga, D., L. Nmeth, P. Halcsy, A. Kornai, V. Trn, and Nagy V. 2005. Parallel corpora for medium density languages. In *Proceedings of the RANLP 2005*, pages 590–596.

Wu, Dekai. 1994. Aligning a parallel english-chinese corpus statistically with lexical criteria. In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics*, pages 80–87, Las Cruces, New Mexico, USA, June. Association for Computational Linguistics.

Tibetan Number Identification Based on Classification of Number Components in Tibetan Word Segmentation

Huidan Liu
Institute of Software, Chinese Academy of Sciences, Graduate University of the Chinese Academy of Sciences
huidan@iscas.ac.cn

Weina Zhao
Beijing Language and Culture University, Qinghai Normal University
weina@iscas.ac.cn

Minghua Nuo
Institute of Software, Chinese Academy of Sciences, Graduate University of the Chinese Academy of Sciences
minghua@iscas.ac.cn

Li Jiang
Institute of Software, Chinese Academy of Sciences
jiangli@iscas.ac.cn

Jian Wu
Institute of Software, Chinese Academy of Sciences
wujian@iscas.ac.cn

Yeping He
Institute of Software, Chinese Academy of Sciences
yeping@iscas.ac.cn

Abstract

Tibetan word segmentation is essential for Tibetan information processing. People mainly use the basic machine matching method which is based on dictionary to segment Tibetan words at present, because there is no segmented Tibetan corpus which can be used for training in Tibetan word segmentation. But the method based on dictionary is not fit to Tibetan number identification. This paper studies the characteristics of Tibetan numbers, and then, proposes a method to identify Tibetan numbers based on classification of number components. The method first tags every number component according to the class it belongs to while segmenting, and then updates the tag series according to some predefined rules. At last adjacent number components are combined to form a Tibetan number if they meet a certain requirement. In the testing result from 7938K Tibetan corpus, the identification accuracy is 99.21%.

1 Introduction

As a phonetic writing script, Tibetan syllables are separated with syllable dots. But like Chinese, there is no separator between Tibetan words. Tibetan word segmentation is essential for Tibetan information processing. In recent years, many experts did much work on Tibetan word segmentation. CHEN Yuzhong (2003) proposed a method based on case auxiliary words and continuous features to segment Tibetan text. Based on this method, using reinstallation rules to identify Abbreviated Words, CAI Zhijie (2009) designed and implemented the Banzhida Tibetan word segmentation system. QI (2006) proposed a three level method to segment Tibetan text. Dolha (2007), Zhaxijia (2007), CAI Rangjia (2009) and TASHI (2009) researched the word categories and annotation scheme for Tibetan corpus and the parts-of-speech and tagging set standards. At present, there is no corpus for Tibetan word segmentation. However, models which are used in Chinese word segmentation, such as HMM, ME, CRF, have to be trained with segmented corpus. As a result, we can't use them in Tibetan word segmentation. So people mainly use machine matching method based on dictionary in Tibetan word segmentation. But machine matching can not be used to identify Tibetan numbers because we can not include all numbers in the dictionary.

In Tibetan text, numbers have 3 different representations. The first is Arabic numbers, such as "2010". The second is Tibetan alphabet numbers composed with Tibetan digital characters: ཀ(0), ཁ(1), ག(2), ང(3), ཅ(4), ཆ(5), ཇ(6), ཉ(7),

Coling 2008: Poster Volume, pages 719–724,
Beijing, August 2010

ཉ(8), ཊ(9), such as “༢༠༡༠”(2010). The third is Tibetan syllable numbers(“Tibetan numbers” in short) which are composed with Tibetan syllables, such as བཅོ་ལྔ(fifteen).The former two classes of numbers can be identified by combining adjacent number characters. However, this method is not fit to the third class, because some Tibetan syllables are used not only in numbers but also in other common words.

According to papers written by Dolha (2007), Zhaxijia (2007), CAI Rangjia (2009) and TASHI (2009), Tibetan numbers should be taken as single words in Tibetan word segmentation, however, we haven’t found any paper on the issue of the identification of Tibetan numbers in Tibetan word segmentation.

In this paper, we propose a method which is based on classification of number components to identify the third class of numbers.

2 Composition of Tibetan numbers

In Tibetan, we use the following syllables (words) to express the meanings of number one to nine: གཅིག གཉིས། གསུམ། བཞི། ལྔ། དྲུག བདུན། བརྒྱད། དགུ།, and the following different syllables for ten, hundred, thousand, ten thousand, million, ten million and so on: བཅུ། བརྒྱ། སྟོང་། ཁྲི། འབུམ། ས་ཡ། ཨ་བ། དུང་ཕྱུར།…..Generally, Tibetan syllable numbers are composed by these syllables, but some syllables have variants, and sometimes we have to use different conjunctions according to the context. The composition of Tibetan syllable numbers has the following rules.

1. Number 1-10 are expressed with the syllables mentioned above, but sometimes variants are used: ཆིག(1),ཉིས།(2),སུམ།(3).
2. Number “tens” (20, 30, 40 …) have the form of “(2-9)+བཅུ”. but in “20”, “30”, variants of “2” and “3” are used, while in “60”, “70”, “80”, variant of “ten”(ཅུ།) is used.
3. Number 11-19 have the form of “བཅུ(10)+(1-9)”, but in “13” and “15” variant of “10”(བཅོ།) is used.
4. Number 21-99, except “tens”, have the form of “(tens)+conjunction+(1-9)”. Different conjunctions are used according to different “tens”: རྩ། སོ། ཞེ། ང་། རེ། དོན། ཙྭ། ཙོ. Sometimes, this form is abbreviated to “conjunction+(1-9)”.
5. In number which is larger then 100, conjunction (དང་) may be used, just like “and” in the reading of English number “115”. Sometimes, (མེད་) is used to express the meaning of vacancy. For example, number “507” is “ལྔ་བརྒྱ་བཅུ་མེད་བདུན”: ལྔ་(five) བརྒྱ་(hundred)བཅུ་ (ten)མེད་(has no)བདུན (seven).
6. Composition of numbers larger than 1000 can be deduced.
7. Ordinal numeral has the form of “(cardinal numeral)+(པor་པོ)”.
8. Multiples have the form of “ལྷག+ (cardinal numeral)”.
9. Fractions have the form of “(cardinal numeral) +ཆ+ (cardinal numeral)”.
10. Decimals have the form of “(cardinal numeral) +དོ+(གངས་རྫོགorཚེག)+ (cardinal numeral)”. “གངས་རྫོད” or “ཚེག” means the decimal point.
11. Approximate numbers have the form of “(cardinal numeral)+(suffix)”. Suffix can be one of (ཙོ། ཙམ། ཡས་མས། ལྷག་ཙམ། གངས་ཀ་ཤས། ཐུག་ཀ་ཤས།…) according to the meaning to be expressed.
12. Some Tibetan numbers don’t obey the above rules. They have no form of number, but have meanings of number, such as “དང་པོ” (first).

3 Tibetan number identification

In this paper, we call all syllables mentioned in the previous section “number components” in general. For some of these number components, we can take it as a part of number when we meet one of them. For others, we can’t, because they can be used to express non-number meanings. So we have to check whether it is a part of a number according to the context when we meet a number component.

Tibetan number identification is a part of Tibetan word segmentation. In Tibetan word seg-

mentation system, Tibetan text is segmented into words by maximum matching method. In this procedure, every Tibetan number is segmented into number components. Then, identification module combines adjacent number components when they meet a certain predefined rules.

In this section, we first briefly introduce the whole procedure of Tibetan word segmentation, then the classification of number components and the tagging method to identify Tibetan number.

3.1 Flow of Tibetan word segmentation

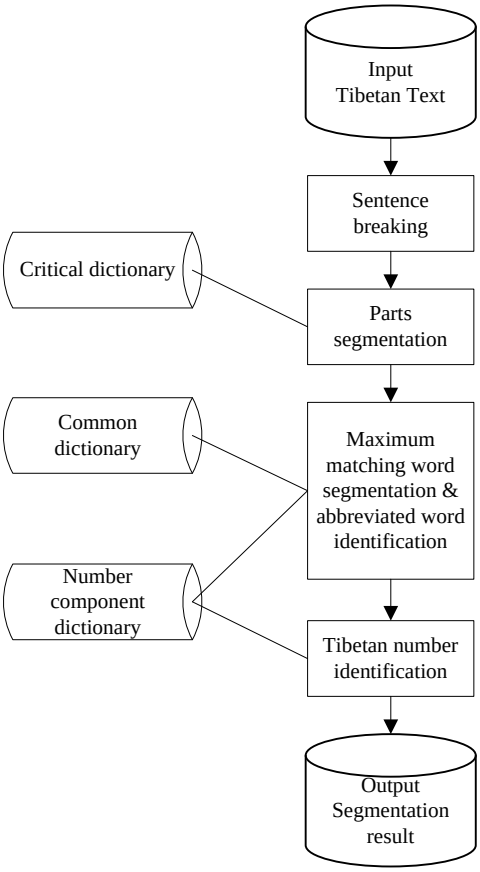


Figure 1. The flow chart of Tibetan word segmentation

As shown in Figure 1, for the input Tibetan text, we first segment it into sentences. Then we segment each sentence into parts with case-auxiliary words. In this procedure, a critical dictionary is used because case-auxiliary words can be a part of some Tibetan words (critical words). When we meet a critical word in Tibetan text, we should not segment it into shorter words. Next, we segment each part into words and identify the abbreviated word (CAI Zhijie, 2009) by maximum matching method with a common dictionary and the number components dictionary. At last, we identify Tibetan numbers and output word segmentation result.

In the procedure of segmenting a part into words, a Tibetan number is segmented into words; we must ensure every one of them is a number component. To do this, both common dictionary and number components dictionary are used. As we use maximum matching method, all Tibetan number words in the common dictionary should be obsolete.

Identification module tags each number component with a tag according to the class which it belongs to, then updates the tags and combines adjacent number components when they meet a certain predefined rules.

3.2 Classification of number components

We classify number components into the following five classes according to their functions and ambiguity.

Basic number: these number components are the basis of Tibetan numbers. Every one of them can be an independent number. If we meet it in context, we should take it as a part of a Tibetan number. Including: Tibetan number 1-9(གཅིག་གཉིས། གསུམ། བཞི། ལྔ། དྲུག་བདུན། བརྒྱད། དགུ།); ten, hundred, thousand, ten thousand, million, ten million and so on(བཅུ། བརྒྱ། སྟོང་། ཁྲི། འབུམ། སར་ཡ། ཉི་ག་དུད་ཕྱུར།…); and their variants.

Number prefix: when it is used as a part of Tibetan number, the next word must be a basic number, while the previous word may be or may not be a number component. Including: abbreviations of "(tens)+conjunction" (སོ། ཞེ། དཾ་རེ། དོན། ཐ། གོ); variants of 1, 2, 3(ཆིག ཉིས། སུམ); decimal point(གདངས་རྫུད་and ཚེག).

Number linker: when it is used as a part of Tibetan number, both the previous word and the next word must be number components. These include (དང་ ཚ མེད་). Conjunctions (སོ། ཞེ། དཾ་རེ། དོན། ཐ། གོ) belong to number prefix class, so we don't include them in this class. But Conjunction (ཙོ) doesn't belong to number prefix class, we include it in this class.

Number suffix: these number components are used to express the meaning of "total number", "approximate number", and "ordinal number" and so on. They follow basic number and should be taken as a part of Tibetan number word. Including: རྒྱ། ཙམ། ཡས་མས། ལྷག་ཙམ། གྲངས་ཀ་ཤས། ཐུག་ཀ་ཤས།…

Independent number: these number components have no form of number, but have meanings of number, such as "དང་པོ" (first).

The difference between "basic number" and "Independent number" is: a basic number can be a Tibetan number itself or a part of a Tibetan number, while an independent number is a Tibetan number itself, but it can't be a part of a Tibetan number.

3.3 Number identification

As shown in Figure 2, identification module tags each number component with a tag according to the class which it belongs to, then updates the tags and combines adjacent number components when they meet a certain predefined rules.

Figure 2. The flow of number identification

Class	Tag
Basic number	N **(**Number**)**
Number prefix	P **(**Prefix**)**
Number linker	L **(**Linker**)**
Number suffix	S**(**Suffix**)**
Independent number	I**(**Independent**)**
Other(non-number)	O **(**Other**)**

Table 1. Classes and their tags

We assign every class with a tag, as shown in Table 1. The tagging procedure screens every segmented part of Tibetan sentences, and tags every word with a tag according to the class which the word belongs to. If the word is not a number component, we tag it with "O" (Other).

As some number components can be used to express non-number meanings, (the cases exist in both number prefix class and number linker class), we have to check whether it is a part of a number according to the context. For number prefix, we take it as a part of number only if it is followed by a basic number, while for number linker, only if it follows a basic number and it is followed by another basic number. We define two rules to do this work.

Rule 1: update tag series "PN" to "NN".
Rule 2: update tag series "NLN" to "NNN".

The tags updating algorithm applies the rules to the current word series until no tag is updated. After tags updating, the tag of a number prefix ("P") is updated to "N" when it is a part of Tibetan number in the context, but the tag will still be "P" when it is not a part of Tibetan number. It is the same for number linkers.

Combination algorithm combines adjacent number components to form a Tibetan number word. It mainly combines continuous number components with tags "NN…N", and the following word is combined too if it has a tag "S". The tag of the number is updated to "N". All words with tag "N" or "I" are taken as Tibetan numbers after combination.

Then the segmentation result is output.

For example, for the following Tibetan sentence:

ལས་འཛོལ་མང་པོ་ཞིག་ནི་བརྒྱ་ཚ་གཅིག་གམ་ཐ་ན་བརྒྱ་ཚ་གྲངས་རྒྱུད་ལྷའི་ནང་ཁོངས་ཀྱི་ལྷ་ལག་གཅིག་ལ་སྟོན་མོར་ནས་བྱུང་འདུག (A considerable parts of accidents were due to the faults of 1% or even 0.5% of components.)

After parts segmentation and maximum matching word segmentation, it is segmented to:

ལས་འཛོལ/ མང་པོ/ ཞིག/ ནི/ བརྒྱ/ ཚ/ གཅིག/ གམ/ ཐ་ན/ བརྒྱ/ ཚ/ གྲངས་རྒྱུད/ ལྷ/ འི/ ནང་ཁོངས/ ཀྱི/ ལྷ་ལག་གཅིག/ ལ/ སྟོན་མོར/ ནས/ བྱུང/ འདུག/

After tagging:

ལས་འཛོལ/(O) མང་པོ/(O) ཞིག/(O) ནི/(O) བརྒྱ/(N) ཚ/(L) གཅིག/(N) གམ/(O) ཐ་ན/(O) བརྒྱ/(N) ཚ/(L) གྲངས་རྒྱུད/(P) ལྷ/(N) འི/(O) ནང་ཁོངས/(O) ཀྱི/(O) ལྷ་ལག་གཅིག/(O) ལ/(O) སྟོན་མོར/(O) ནས/(O) བྱུང/(O) འདུག/(O)

The corresponding tag series is:

OOOONLNOONLPNOOOOOOOOO

After the first run of tags updating, the tag series is changed to:

OOOONNNOONLNNOOOOOOOOO

After the second run of tags updating, the tag series is changed to:

OOOONNNOONNNNOOOOOOOOO

In the third run of tags updating, no tag is updated. Then, combination algorithm combines adjacent number components corresponding to the continuous "N" tags. The result is:

ལས་འཛིན་/ མང་པོ་/ ཞིག་/ ནི་/ བརྒྱ་ཚ་གཅིག་/ གམ་/ ཐ་ན་/ བརྒྱ་ཆག་ལས་ལྷུང་ལྷ་/ འི་/ ནང་ཁོངས་/ ཀྱི་/ ལྷུ་ལག་གཅིག་/ ལ་/ སྐྱོན་མོར་/ ནས་/ ཟྱང་/ འདུག/

The corresponding tag series is:

OOOONOONOOOOOOOOO

It has two "N" tags, which means two Tibetan numbers are identified.

4 Experiment

Corpus	Byte	Sentence	BNS	TNS
Corpus 1	1624K	13957	2590	1667
Corpus 2	1334K	11441	1748	1076
Corpus 3	1408K	11923	1751	969
Corpus 4	1015K	8453	1212	672
Corpus 5	1311K	10445	1613	897
Corpus 6	1246K	10009	1474	880
Total	7938K	66228	10388	6161

Table 2. Information about the 6 corpuses

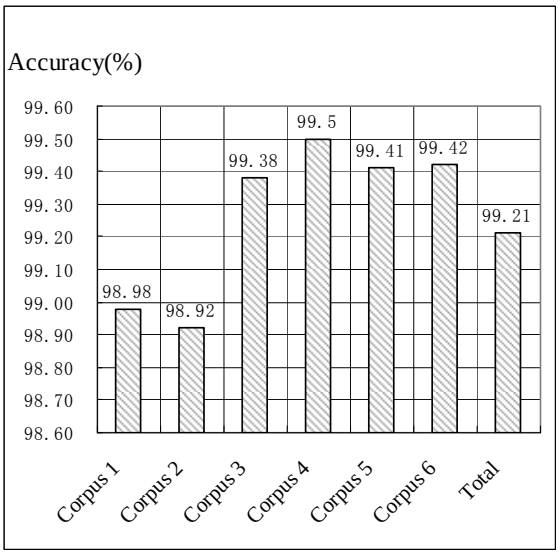


Figure 3. Accuracy of Tibetan number identification on 6 corpuses

As there is no corpus for Tibetan word segmentation, we have to make experiment on original Tibetan texts. We make use of several books which are written in Tibetan, and collect many web pages from several Tibetan web sites. After preprocessing, we get six corpuses. The basic information about the corpuses is shown in Table 2. Note that, in Table 2, the column "BNS" includes all sentences which have in it at least one number component belonging to basic number class, while the column "TNS" includes all sentences which have at least one Tibetan number in it. The count of the former is significantly larger than the count of the later because some basic numbers are used in idioms and proverbs which should be segmented as single words, thus we don't take them as number components under this circumstance. Figure 3 shows the results of our experiment. As we can see, the total identification accuracy is 99.21%. As we have included all basic numbers in our method, theoretically the recall is 100%.

After analyzing the results, we find that wrongly identified words can be divided into two classes. One is that there is a conjunction (དང) between two Tibetan numbers, but is taken as one Tibetan number, such as "བཅུ་དང་ཉི་ཤུ།" (ten and twenty), "ཁྲི་གཅིག་དང་ཁྲི་གཉིས" (ten thousand and twenty thousand). The other is that some Tibetan numbers has other non-number meanings in the context, but our algorithm takes them as numbers. For instance, "ཞེ་གཅིག" means 41 when it is used as a number, but it has another meaning of "similarly"; "དོན་ལྔ" means 75 when it is used as a number, but it has the meaning of "the five internal organs".

5 Conclusion

Tibetan syllables are separated with syllable dots. But like Chinese, there is no separator between Tibetan words. Tibetan word segmentation is essential for Tibetan information processing. People mainly use machine matching in Tibetan word segmentation base on dictionary. But machine matching can not be used to identify Tibetan numbers because we can not include all numbers in our dictionary. This paper proposes a method to tag number components according to the classes they belong to, and then apply predefined rules to update tag series, and next combine adjacent number components to

form a Tibetan number. In the testing result from 7938K Tibetan corpus, the identification accuracy is 99.21%, which means that this method is feasible to be applied to Tibetan word segmentation.

Acknowledgement

We thank the anonymous reviewers for their insightful comments that helped us improve the quality of the paper.

References

CHEN Yuzhong, LI Baoli, YU Shiwen, LAN Cuoji. 2003. An Automatic Tibetan Segmentation Scheme Based on Case Auxiliary Words and Continuous Features, *Applied Linguistics*, 2003(01): 75-82.

CHEN Yuzhong, LI Baoli, YU Shiwen. 2003. The Design and Implementation of a Tibetan Word Segmentation System, *Journal of Chinese Information Processing*, 17(3): 15-20.

CAI Rangjia. 2009. Research on the Word Categories and Its Annotation Scheme for Tibetan Corpus, *Journal of Chinese Information Processing*, 23(04):107-112

CAI Zhijie. 2009. Identification of Abbreviated Word in Tibetan Word Segmentation, *Journal of Chinese Information Processing*, 23(01):35-37.

CAI Zhijie. 2009. The Design of Banzhida Tibetan word segmentation system, *the 12th Symposium on Chinese Minority Information Processing*.

Dolha, Zhaxijia, Losanglangjie, Ouzhu. 2007. The parts-of-speech and tagging set standards of Tibetan information process, *the 11th Symposium on Chinese Minority Information Processing*.

QI Kunyu. 2006. On Tibetan Automatic Participate Research with the Aid of Information Treatment *Journal of Northwest University for Nationalities (Philosophy and Social Science)*, 2006(04):92-97.

SUN Yuan, LUO Sangqiangba, YANG Rui and ZHAO Xiaobing. 2009. Design of a Tibetan Automatic Segmentation Scheme, *the 12th Symposium on Chinese Minority Information Processing*.

TASHI Gyal, ZHU Jie. 2009. Research on Tibetan Segmentation Scheme for Information Processing, *Journal of Chinese Information Processing*, 23(04):113-117.

Zhaxijia, Dolha, Losanglangjie, Ouzhu. 2007. The theoretical explanation on "the parts-of-speech and tagging set standards of Tibetan information process", *the 11th Symposium on Chinese Minority Information Processing*.

Collective Semantic Role Labeling on Open News Corpus by Leveraging Redundancy

[1,2]Xiaohua Liu, [3]Kuan Li[*], [4]Bo Han[*], [2]Ming Zhou,
[2]Long Jiang, [5]Daniel Tse[*] and [3]Zhongyang Xiong
[1]School of Computer Science and Technology
Harbin Institute of Technology
[2]Microsoft Research Asia
[3]College of Computer Science
Chongqing University
[4]School of Software
Dalian University of Technology
[5]School of Information Technologies
The University of Sydney
{xiaoliu, v-kuli, v-bohan, mingzhou, longj}
@microsoft.com
dtse6695@it.usyd.edu.au
zyxiong@cqu.edu.cn

Abstract

We propose a novel MLN-based method that collectively conducts SRL on groups of news sentences. Our method is built upon a baseline SRL, which uses no parsers and leverages redundancy. We evaluate our method on a manually labeled news corpus and demonstrate that news redundancy significantly improves the performance of the baseline, e.g., it improves the F-score from 64.13% to 67.66%.

1 Introduction

Semantic Role Labeling (SRL, Màrquez, 2009) is generally understood as the task of identifying the arguments of a given predicate and assigning them semantic labels describing the roles they play. For example, given a sentence *The luxury auto maker sold 1,214 cars.*, the goal is to identify the arguments of *sold* and produce the following output: [A0 *The luxury auto maker*] [V *sold*] [A1 *1,214 cars*]. Here *A0* represents the *seller*, and *A1* represents the things *sold* (CoNLL 2008 shared task, Surdeanu et al., 2008).

Gildea and Jurafsky (2002) first tackled SRL as an independent task, which is divided into several sub-tasks such as argument identification, argument classification, global inference, etc. Some researchers (Xue and Palmer, 2004; Koomen et al., 2005; Cohn and Blunsom, 2005; Punyakanok et al., 2008; Toutanova et al., 2005; Toutanova et al., 2008) used a pipelined approach to attack the task. Some others resolved the sub-tasks simultaneously. For example, some work (Musillo and Merlo, 2006; Merlo and Musillo, 2008) integrated syntactic parsing and SRL into a single model, and another (Riedel and Meza-Ruiz, 2008; Meza-Ruiz and Riedel, 2009) jointly handled all sub-tasks using Markov Logic Networks (MLN, Richardson and Domingos, 2005).

All the above methods conduct sentence level SRL, and rely on parsers. Parsers have showed great effects on SRL performance. For example, Xue and Palmer (2004) reported that SRL performance dropped more than 10% when they used syntactic features from an automatic parser instead of the gold standard parsing trees. Even worse, parsers are not robust and cannot always analyze any input, due to the fact that some inputs are not in the language described by the parser's formal grammar, or adequately represented within the parser's training data.

[*] This work has been done while the author was visiting Microsoft Research Asia.

Coling 2008: Poster Volume, pages 725–729,
Beijing, August 2010

We propose a novel MLN-based method that collectively conducts SRL on groups of news sentences to leverage the content redundancy in news. To isolate the negative effect of noise from parsers and thus focus on the study of the contribution of redundancy to SRL, we use no parsers in our approach. We built a baseline SRL, which depends on no parsers, and use the MLN framework to exploit redundancy. Our intuition is that SRL on one sentence can help that on other differently phrased sentences with similar meaning. For example, consider the following sentence from a news article:

> *A suicide **bomber** blew himself up Sunday in market in Pakistan's **northwest** crowded with shoppers ahead of a Muslim holiday, **killing** 12 people, including a mayor who once supported but had turned against the Taliban, officials said.*

The state-of-art MLN-based system (Meza-Ruiz and Riedel, 2009), hereafter referred to as MLNBS for brevity, incorrectly labels *northwest* instead of *bomber* as *A0* of *killing*. Now consider another sentence from another news article:

> *Police in northwestern Pakistan say that a suicide **bomber** has **killed** at least 13 people and wounded dozens of others.*

Here MLNBS correctly identify *bomber* as A0 of *killing*. When more sentences are observed where *bomber* as *A0* of *killing* is correctly identified, we will be more confident that *bomber* should be labeled as *A0* of *killing*, and that *northwest* should not be the *A0* of *killing* according to the constraint that one predicate has at most one A0.

We manually construct a news corpus to evaluate our method. In the corpus, semantic role information is annotated and sentences with similar meanings are grouped together. Experimental results show that news redundancy can significantly improve the performance of the baseline system.

Our contributions can be summarized as follows:

1. We present a novel method that conducts SRL on a set of sentences collectively, instead of on a single sentence, by extending MLNBS to leverage redundancy.
2. We show redundancy can significantly improve the performance of the baseline system, indicating a promising research direction towards open SRL.

In the next section, we introduce news sentence extraction and clustering. In Section 3, we describe our collective inference method. In Section 4, we show our experimental results. Finally, in Section 5 we conclude our paper with a discussion of future work.

2 Extraction and Clustering of News Sentences

To construct a corpus to evaluate our method, we extract sentences from clustered news articles returned by news search engines such as Bing and Google, and divide them into groups so that sentences in a group have similar meaning.

News articles in the same cluster are supposed to report the same event. Thus we first group sentences according to the news cluster they come from. Then we split sentences in the same cluster into several groups according to the similarity of meaning. We assume that two sentences are more similar in meaning if they share more synonymous proper nouns and verbs. The synonyms of verbs, like *plod* and *trudge*, are mainly extracted from the Microsoft Encarta Dictionary[1], and the proper nouns thesaurus, containing synonyms such as *U.S.* and *the United States*, is manually compiled.

As examples, below are two sentence groups which are extracted from a news cluster describing Hurricane Ida.

Group 1:

- *Hurricane Ida, the first Atlantic hurricane to target the U.S. this year, plodded yesterday toward the Gulf Coast…*
- *Hurricane Ida trudged toward the Gulf Coast…*
- …

Group 2:

- *It could make landfall as early as Tuesday morning, although it was forecast to weaken further.*

[1] http://uk.encarta.msn.com/encnet/features/dictionary/dictionaryhome.aspx

- *Authorities said Ida could make landfall as early as Tuesday morning, although it was forecast to weaken by then.*
- *…*

3 Collective Inference Based on MLN

Our method includes two core components: a baseline system that conducts SRL on every sentence; and a collective inference system that accepts as input a group of sentences with preliminary SRL information provided by the baseline.

We build the baseline by removing formulas involving syntactic parsing information from MLNBS (while keeping other rules) and retraining the system using the tool and scripts provided by Riedel and Meza-Ruiz (2008) on the manually annotated news corpus described in Section 4.

A collective inference system is constructed to leverage redundancy in the SRL information from the baseline.

We first redefine the predicate *role* and treat it as observed:

predicate ***role****: Int x Int x Int x Role;*

role has four parameters: the first one stands for the number of sentence in the input, which is necessary to distinguish the sentences in a group; the other three are taken from the arguments of the *role* predicate defined by Riedel and Meza-Ruiz (2008), which denote the positions of the predicate and the argument in the sentence and the role of the argument, respectively. If the predication holds, it returns 1, otherwise 0.

A hidden predicate *final_role* is defined to present the final output, which has the same parameters as the predicate *role*:

predicate ***final_role****: Int x Int x Int x Role;*

We introduce the following formula, which directly passes the semantic role from the baseline to the final output:

$$role(s, p, a, +r) \Rightarrow final\_role(s, p, a, +r) \quad (1)$$

Here *s* is the sentence number in a group; *p* and *a* denote the positions of the predicate and argument in *s,* respectively*; r* stands for the role of the argument; the "+" before the variable *r* indicates that different *r* has different weight.

Then we define another formula for collective inference:

$$s1 \neq s2 \wedge lemma(s1,p1,p\_lemma) \wedge lemma(s2,p2,p\_lemma) \wedge lemma(s1,a1,a\_lemma) \wedge lemma(s2,a2,a\_lemma) \wedge role(s2,p2,a2,+r) \Rightarrow final\_role(s1,p1,a1,+r) \quad (2)$$

Here *p_lemma(a_lemma)* stands for the lemma of the predicate(argument), which is obtained from the lemma dictionary. This dictionary is extracted from the dataset of CoNLL 2008 shared task and is normalized using synonym dictionary described in Section 2; *lemma* is an observed predicate that states whether or not the word has the lemma.

Formula 2 encodes our basic ideas about collective SRL: given several sentences expressing similar meaning, if one sentence has a predicate *p* with an argument *a* of role *r*, the other sentences would be likely to have a predicate *p'* with an argument *a'* of role *r*, where *p'* and *a'* are the same or synonymous with *p* and *a,* respectively, as illustrated by the example in Section 1.

Besides, we also apply structural constraints (Riedel and Meza-Ruiz, 2008) to *final_role*.

To learn parameters of the collective inference system, we use *thebeast* (Riedel and Meza-Ruiz, 2008), which is an open Markov Logic Engine, and train it on manually annotated news corpus described in Section 4.

4 Experiments

To train and test the collective inference system, we extract 1000 sentences from news clusters, and group them into 200 clusters using the method described in Section 2. For every sentence, POS tagging is conducted with the OpenNLP toolkit (Jason Baldridge et al., 2009), lemma of each word is obtained through the normalized lemma dictionary described in Section 3, and SRL is manually labeled. To reduce human labeling efforts, we retrain our baseline on the WSJ corpus of CoNLL 2008 shared task and run it on our news corpus, and then edit the SRL outputs by hand.

We implement the collective inference system with the *thebeast* toolkit. Precision, recall, and F-score are used as evaluation metrics. In both training and evaluation, we follow the CoNLL 2008 shared task and regard only heads of phrases as arguments.

Table 1 shows the averaged 10-fold cross validation results of our systems and the baseline, where the third and second line report the results of using and not using Formula 1 in our collective inference system, respectively.

Systems	Pre. (%)	Rec. (%)	F-score (%)
Baseline	**69.87**	59.26	64.13
CI-1	62.99	**72.96**	67.61
CI	67.01	68.33	**67.66**

Table 1. Averaged 10-fold cross validation results (Pre.: precision; Rec.: recall).

Experimental results show that the two collective inference engines (CI-1 and CI) perform significantly better than the baseline in terms of the recall and F-score, though a little worse in the precision. We observe that predicate-argument relationships in sentences with complex syntax are usually not recognized by the baseline, but some of them are correctly identified by the collective inference systems. This, we guess, explains in large part the difference in performance. For instance, consider the following sentences in a group, where *order* and *tell* are synonyms:

- Colombia said on Sunday it will appeal to the U.N. Security Council and the OAS after Hugo *Chavez*, the fiery leftist president of neighboring Venezuela, *ordered* his army to prepare for war in order to assure peace.
- President Hugo *Chavez ordered* Venezuela's military to prepare for a possible armed conflict with Colombia, saying yesterday that his country's soldiers should be ready if the U.S. tries to provoke a war between the South American neighbors.
- Venezuelan President Hugo *Chavez told* his military and civil militias to prepare for a possible war with Colombia as tensions mount over an agreement giving U.S. troops access to Colombian military bases.

The baseline cannot label (*ordered, Chavez, A0*) for the first sentence, partially owing to the syntactic complexity of the sentence, but can identify the relationship for the second and third sentence. In contrast, the collective inference systems can identify *Chavez* in the first sentence as A0 of *order* because of its occurrence in the other sentences of the same group.

As Table 1 shows, the CI system achieves the highest F-score (67.66%), and a higher precision than the CI-1 system, indicating the effectiveness of Formula 1. Consider the above three sentences. CI-1 mislabels (*ordered, Venezuela, A1*) for the first sentence because the baseline labels it for the second sentence. In contrast, CI does not label it for the first sentence because the baseline does not and (*ordered, Venezuela, A1*) rarely occurs in the outputs of the baseline for this sentence group.

We also find cases where the collective inference systems do not but should help. For example, consider the following group of sentences:

- A Brazilian *university* expelled a woman who was heckled by hundreds of fellow students when she wore a short, pink dress to class, *taking* out newspaper ads Sunday to publicly accuse her of immorality.
- The *university* also *published* newspaper ads accusing the student, Geisy Arruda, of immorality.

The baseline has identified *(published, university, A0)* for the second sentence. But neither the baseline nor our method labels *(taking, university, A0)* for the first one. This happens because *publish* is not considered as a synonym of *take*, and thus *(published, university, A0)* in the second provides no evidence for *(taking, university, A0)* in the first. We plan to develop a context based synonym detection component to address this issue in the future.

5 Conclusions and Future Work

We present a novel MLN-based method that collectively conducts SRL on groups of sentences. To help build training and test corpora, we design a method to collect news sentences and to divide them into groups so that sentences of similar meaning fall into the same cluster. Experimental results on a manually labeled news corpus show that collective inference, which leverages redundancy, can effectively improve the performance of the baseline.

In the future, we plan to evaluate our method on larger news corpora, and to extend our method to other genres of corpora, such as tweets.

References

Baldridge, Jason, Tom Morton, and Gann. 2009. *OpenNLP*, http://opennlp.sourceforge.net/

Cohn, Trevor and Philip Blunsom. 2005. Semantic role labelling with tree conditional random fields. *Proceedings of the Ninth Conference on Computational Natural Language Learning*, pages: 169-172.

Gildea, Daniel and Daniel Jurafsky. 2002. Automatic Labeling of Semantic Roles. *Journal of Computational Linguistics*, 28(3):245–288.

Koomen, Peter, Vasin Punyakanok, Dan Roth, and Wen-tau Yih. 2005. Generalized inference with multiple semantic role labeling systems. *Proceedings of the Ninth Conference on Computational Natural Language Learning*, pages: 181-184.

Màrquez, Lluís. 2009. *Semantic Role Labeling Past, Present and Future*, Tutorial of ACL-IJCNLP 2009.

Merlo, Paola and Gabriele Musillo. 2008. Semantic parsing for high-precision semantic role labelling. *Proceedings of the Twelfth Conference on Computational Natural Language Learning*, pages: 1-8.

Meza-Ruiz, Ivan and Sebastian Riedel. 2009. Jointly Identifying Predicates, Arguments and Senses using Markov Logic. *Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the ACL*, pages: 155-163.

Musillo, Gabriele and Paola Merlo. 2006. Accurate Parsing of the proposition bank. *Proceedings of the Human Language Technology Conference of the NAACL*, pages: 101-104.

Punyakanok, Vasin, Dan Roth and Wen-tau Yih. 2008. The importance of syntactic parsing and inference in semantic role labeling. *Journal of Computational Linguistics*, 34(2), 257-287.

Richardson, Matthew and Pedro Domingos. 2005. Markov logic networks. *Technical Report, University of Washington,* 2005.

Riedel, Sebastian and Ivan Meza-Ruiz. 2008. Collective semantic role labelling with Markov Logic. *Proceedings of the Twelfth Conference on Computational Natural Language Learning,* pages: 193-197.

Surdeanu, Mihai, Richard Johansson, Adam Meyers, Lluís Màrquez, and Joakim Nivre. 2008. The conll 2008 shared task on joint parsing of syntactic and semantic dependencies. *Proceedings of the Twelfth Conference on Computational Natural Language Learning,* pages: 159-177.

Toutanova, Kristina, Aria Haghighi and Christopher D. Manning. 2005. Joint learning improves semantic role labeling. *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics,* pages: 589-596.

Toutanova, Kristina, Aria Haghighi and Christopher D. Manning. 2008. A global joint model for semantic role labeling. *Journal of Computational Linguistics*, 34(2), 161-191.

Xue, Nianwen and Martha Palmer. 2004. Calibrating features for semantic role labeling. *Proceedings of the Conference on Empirical Methods in Natural Language Processing,* pages: 88-94.

Improved Discriminative ITG Alignment using Hierarchical Phrase Pairs and Semi-supervised Training

[†]Shujie Liu[*], [‡]Chi-Ho Li and [‡]Ming Zhou
[†] School of Computer Science and Technology
Harbin Institute of Technology
shujieliu@mtlab.hit.edu.cn
[‡]Microsoft Research Asia
{chl, mingzhou}@microsoft.com

Abstract

While ITG has many desirable properties for word alignment, it still suffers from the limitation of one-to-one matching. While existing approaches relax this limitation using phrase pairs, we propose a ITG formalism, which even handles units of non-contiguous words, using both simple and hierarchical phrase pairs. We also propose a parameter estimation method, which combines the merits of both supervised and unsupervised learning, for the ITG formalism. The ITG alignment system achieves significant improvement in both word alignment quality and translation performance.

1 Introduction

Inversion transduction grammar (ITG) (Wu, 1997) is an adaptation of CFG to bilingual parsing. It does synchronous parsing of two languages with phrasal and word-level alignment as by-product. One of the merits of ITG is that it is less biased towards short-distance reordering compared with other word alignment models such as HMM. For this reason ITG has gained more and more attention recently in the word alignment community (Zhang et al., 2005; Cherry et al., 2006; Haghighi et al., 2009) .

The basic ITG formalism suffers from the major drawback of one-to-one matching. This limitation renders ITG unable to produce certain alignment patterns (such as many-to-many alignment for idiomatic expression). For this reason those recent approaches to ITG alignment introduce the notion of *phrase* (or *block*), defined as sequence of contiguous words, into the ITG formalism (Cherry and Lin, 2007; Haghighi et al., 2009; Zhang et al., 2008). However, there are still alignment patterns which cannot be captured by phrases. A simple example is connective in Chinese/English. In English, two clauses are connected by merely one connective (like "*although*", "*because*") but in Chinese we need two connectives (e.g. There is a sentence pattern "虽然 X_1 但是 $X_2 \rightarrow X_2$ although X_1", where X_1 and X_2 are variables for clauses). The English connective should then be aligned to two non-contiguous Chinese connectives, and such alignment pattern is not available in either word-level or phrase-level ITG. As hierarchical phrase-based SMT (Chiang, 2007) is proved to be superior to simple phrase-based SMT, it is natural to ask, why don't we further incorporate hierarchical phrase pairs (henceforth h-phrase pairs) into ITG? In this paper we propose a ITG formalism and parsing algorithm using h-phrase pairs.

The ITG model involves much more parameters. On the one hand, each phrase/h-phrase pair has its own probability or score. It is not feasible to learn these parameters through discriminative/supervised learning since the repertoire of phrase pairs is much larger than the size of human-annotated alignment set. On the other hand, there are also a few useful features which cannot be estimated merely by unsupervised learning like EM. Inspired by Fraser et al. (2006), we propose a semi-supervised learning algorithm which combines the merits of both discrimina-

[*] This work has been done while the first author was visiting Microsoft Research Asia.

Coling 2008: Poster Volume, pages 730–738,
Beijing, August 2010

tive training (error minimization) and approximate EM (estimation of numerous parameters).

The ITG model augmented with the learning algorithm is shown by experiment results to improve significantly both alignment quality and translation performance.

In the following, we will explain, step-by-step, how to incorporate hierarchical phrase pairs into the ITG formalism (Section 2) and in ITG parsing (Section 3). The semi-supervised training method is elaborated in Section 4. The merits of the complete system are illustrated with the experiments described in Section 5.

2 ITG Formalisms

2.1 W-ITG : ITG with only word pairs

The simplest formulation of ITG contains three types of rules: terminal unary rules $X \rightarrow e/f$, where e and f represent words (possibly a null word, ε) in the English and foreign language respectively, and the binary rules $X \rightarrow [X, X]$ and $X \rightarrow \langle X, X \rangle$, which refer to that the component English and foreign phrases are combined in the same and inverted order respectively. From the viewpoint of word alignment, the terminal unary rules provide the links of word pairs, whereas the binary rules represent the reordering factor. Note also that the alignment between two phrase pairs is always composed of the alignment between word pairs (c.f. Figure 1(a) and (b)). The Figure 1 also shows ITG can handle the cases where two languages share the same (Figure 1(a)) and different (Figure 1(b)) word order

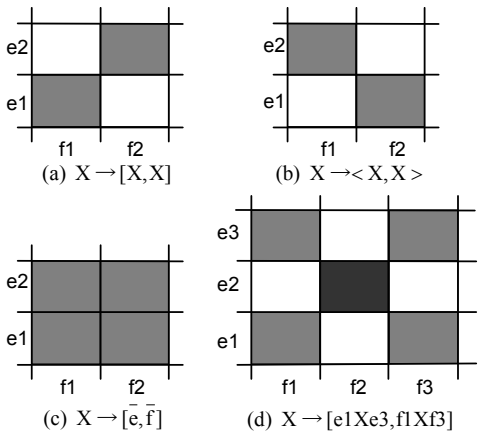


Figure 1. Four ways in which ITG can analyze a multi-word span pair.

Such a formulation has two drawbacks. First of all, the simple ITG leads to redundancy if word alignment is the sole purpose of applying ITG. For instance, there are two parses for three consecutive word pairs, viz. [a/a' [b/b' c/c']] and [[a/a' b/b'] c/c']. The problem of redundancy is fixed by adopting ITG normal form. The ITG normal form grammar as used in this paper is described in Appendix A.

The second drawback is that ITG fails to produce certain alignment patterns. Its constraint that a word is not allowed to align to more than one word is indeed a strong limitation as no idiom or multi-word expression is allowed to align to a single word on the other side. Moreover, its reordering constraint makes it unable to produce the 'inside-out' alignment pattern (c.f. Figure 2).

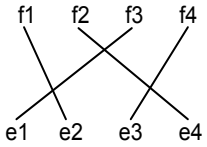


Figure 2. An example of inside-out alignment.

2.2 P-ITG : ITG with Phrase Pairs

A single word in one language is not always on a par with a single word in another language. For example, the Chinese word "白宫" is equivalent to two words in English ("white house"). This problem is even worsened by segmentation errors (i.e. splitting a single word into more than one word). The one-to-one constraint in W-ITG is a serious limitation as in reality there are always segmentation or tokenization errors as well as idiomatic expressions. Therefore, researches like Cherry and Lin (2007), Haghighi et al. (2009) and Zhang et al. (2009) tackle this problem by enriching ITG, in addition to word pairs, with pairs of phrases (or blocks). That is, a sequence of source language word can be aligned, as a whole, to one (or a sequence of more than one) target language word.

These methods can be subsumed under the term phrase-based ITG (P-ITG), which enhances W-ITG by altering the definition of a terminal production to include phrases: $X \rightarrow \bar{e}/\bar{f}$ (c.f. Figure 1(c)). $\bar{e}$ stands for English phrase and $\bar{f}$ stands for foreign phrase. As an example, if there is a simple phrase pair <white house, 白

宫>, then it is transformed into the ITG rule $C \rightarrow$"white house"/"白宫".

An important question is how these phrase pairs can be formulated. Marcu and Wong (2002) propose a joint probability model which searches the phrase alignment space, simultaneously learning translations lexicons for words and phrases without consideration of potentially sub-optimal word alignments and heuristic for phrase extraction. This method suffers from computational complexity because it considers all possible phrases and all their possible alignments. Birch et al. (2006) propose a better and more efficient method of constraining the search space which does not contradict a given high confidence word alignment for each sentence. Our P-ITG collects all phrase pairs which are consistent with a word alignment matrix produced by a simpler word alignment model.

2.3 HP-ITG : P-ITG with H-Phrase pairs

P-ITG is the first enhancement of ITG to capture the linguistic phenomenon that more than one word of a language may function as a single unit, so that these words should be aligned to a single unit of another language. But P-ITG can only treat contiguous words as a single unit, and therefore cannot handle the single units of non-contiguous words. Apart from sentence connectives as mentioned in Section 1, there is also the example that the single word "since" in English corresponds to two non-adjacent words "自" and "以来" as shown the following sentence pair:

自 上周末 ***以来***， 我 一直 在 生病 .

I have been ill ***since*** last weekend .

No matter whether it is P-ITG or phrase-based SMT, the very notion of phrase pair is not helpful because this example is simply handled by enumerating all possible contiguous sequences involving the words "自" and "以来", and thus subject to serious data sparseness. The lesson learned from hierarchical phrase-based SMT is that the modeling of non-contiguous word sequence can be very simple if we allow rules involving h-phrase pairs, like:

$C \rightarrow$"since X"/自 X 以来"

where X is a placeholder for substituting a phrase pair like "上周末/last weekend".

H-phrase pairs can also perform reordering, as illustrated by the well-known example from Chiang (2007), $C \rightarrow$ "have X_2 with X_1" / "与 X_1 有 X_2", for the following bilingual sentence fragment:

与 北韩 ***有*** 邦交

have diplomatic relations ***with*** North Korea

The potential of intra-phrase reordering may also help us to capture those alignment patterns like the 'inside-out' pattern.

All these merits of h-phrase pairs motivate a ITG formalism, viz. hierarchical phrase-based ITG (HP-ITG), which employs not only simple phrase pairs but also hierarchical ones. The ITG grammar is enriched with rules of the format: $X \rightarrow \bar{e}/\bar{f}$ where $\bar{e}$ and $\bar{f}$ refer to either a phrase or h-phrase (c.f. Figure 1(d)) pair in English and foreign language respectively[2]. Note that, although the format of HP-ITG is similar to P-ITG, it is much more difficult to handle rules with h-phrase pairs in ITG parsing, which will be elaborated in the next section.

It is again an important question how to formulate the h-phrase pairs. Similar to P-ITG, the h-phrase pairs are obtained by extracting the h-phrase pairs which are consistent with a word alignment matrix produced by some simpler word alignment model.

3 ITG Parsing

Based on the rules, W-ITG word alignment is done in a similar way to chart parsing (Wu, 1997). The base step applies all relevant terminal unary rules to establish the links of word pairs. The word pairs are then combined into span pairs in all possible ways. Larger and larger span pairs are recursively built until the sentence pair is built.

Figure 3(a) shows one possible derivation for a toy example sentence pair with three words in each sentence. Each node (rectangle) represents a pair, marked with certain phrase category, of

[2] Haghighi et al. (2009) impose some rules which look like h-phrase pairs, but their rules are essentially h-phrase pairs with at most one 'X' only, added with the constraint that each 'X' covers only one word.

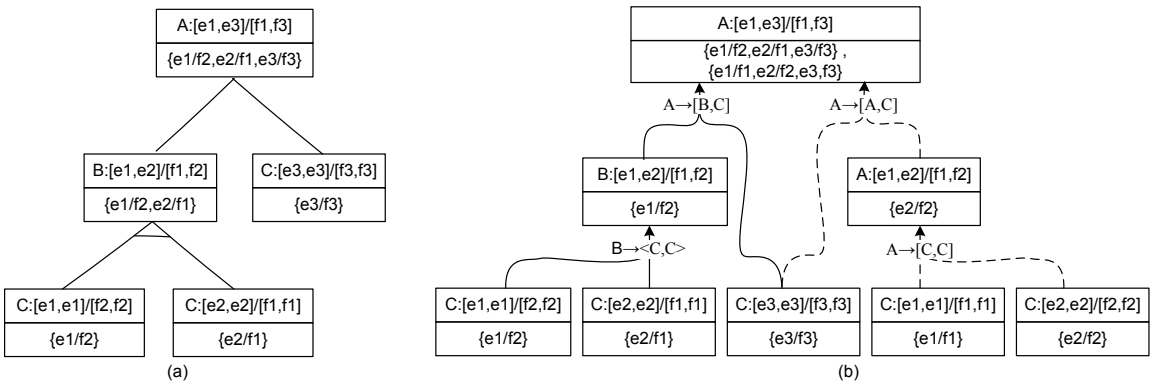


Figure 3. Example ITG parses in graph (a) and hypergraph (b).

foreign span (F-span) and English span (E-span) (the upper half of the rectangle) and the associated alignment hypothesis (the lower half). Each graph like Figure 3(a) shows only one derivation and also only one alignment hypothesis.

The various derivations in ITG parsing can be compactly represented in hypergraph (Klein et al., 2001) like Figure 3(b). Each hypernode (rectangle) comprises both a span pair (upper half) and the list of possible alignment hypotheses (lower half) for that span pair. The hyperedges show how larger span pairs are derived from smaller span pairs. Note that hypernode may have more than one alignment hypothesis, since a hypernode may be derived through more than one hyperedge (e.g. the topmost hypernode in Figure 3(b)). Due to the use of normal form, the hypotheses of a span pair are different from each other.

In the case of P-ITG parsing, each span pair does not only examine all possible combinations of sub-span pairs using binary rules, but also checks if the yield of that span pair is exactly the same as that phrase pair. If so, then this span pair is treated as a valid leaf node in the parse tree. Moreover, in order to enable the parse tree produce a complete word aligned matrix as by-product, the alignment links within the phrase pair (which are recorded when the phrase pair is extracted from a word aligned matrix produced by a simpler model) are taken as an alternative alignment hypothesis of that span pair.

In the case of HP-ITG parsing, an ITG rule like $C \rightarrow$ "have X_2 with X_1"/"与 X_1有 X_2" (originated from the hierarchical rule like $X \rightarrow$ <与 X_1 有 X_2, have X_2 with X_1>), is processed in the following manner: 1) Each span pair checks if it contains the lexical anchors: "have", "with","与" and "有"; 2) each span pair checks if the remaining words in its yield can form two sub-span pairs which fit the reordering constraint among X_1 and X_2 (Note that span pairs of any category in the ITG normal form grammar can substitute for X_1 or X_2). 3) If both conditions hold, then the span pair is assigned an alignment hypothesis which combines the alignment links among the lexical anchors and those links among the sub-span pairs.

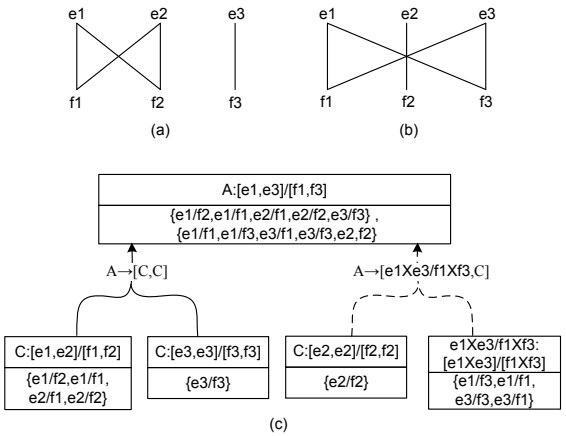


Figure 4. Phrase/h-phrase in hypergraph.

Figure 4(c) shows an example how to use phrase pair and h-phrase pairs in hypergraph. Figure 4(a) and Figure 4(b) refer to alignment matrixes which cannot be generated by W-ITG, because of the one-to-one assumption. Figure 4(c) shows how the span pair [e1,e3]/[f1,f3] can be generated in two ways: one is combining a phrase pair and a word pair directly, and the other way is replacing the *X* in the h-phrase pair with a word pair. Here we only show how h-phrase pairs with one variable be used during the

parsing, and h-phrase pairs with more than one variable can be used in a similar way.

The original (unsupervised) ITG algorithm has complexity of $O(n^6)$. When extended to supervised/discriminative framework, ITG runs even more slowly. Therefore all attempts to ITG alignment come with some pruning method. Zhang and Gildea (2005) show that Model 1 (Brown et al., 1993) probabilities of the word pairs inside and outside a span pair are useful. Tic-tac-toe pruning algorithm (Zhang and Gildea, 2005) uses dynamic programming to compute inside and outside scores for a span pair in $O(n^4)$. Tic-tac-toe pruning method is adopted in this paper.

4 Semi-supervised Training

The original formulation of ITG (W-ITG) is a generative model in which the ITG tree of a sentence pair is produced by a set of rules. The parameters of these rules are trained by EM. Certainly it is difficult to add more non-independent features in such a generative model, and therefore Cherry et al. (2006) and Haghighi et al. (2009) used a discriminative model to incorporate features to achieve state-of-art alignment performance.

4.1 HP-DITG : Discriminative HP-ITG

We also use a discriminative model to assign score to an alignment candidate for a sentence pair $(\bar{f}, \bar{e})$ as probability from a log-linear model (Liu et al., 2005; Moore, 2006):

$$P(a|\bar{e},\bar{f}) = \frac{exp(\sum_i \lambda_i \Psi_i(a,\bar{f},\bar{e}))}{\sum_{a' \in A} exp(\sum_i \lambda_i \Psi_i(a',\bar{f},\bar{e}))} \quad (1)$$

where each $\Psi_i(a,\bar{f},\bar{e})$ is some feature about the alignment matrix, and each λ is the weight of the corresponding feature. The discriminative version of W-ITG, P-ITG, and HP-ITG are then called W-DITG, P-DITG, and HP-DITG respectively.

There are two kinds of parameters in (1) to be learned. The first is the values of the features Ψ. Most features are indeed about the probabilities of the phrase/h-phrase pairs and there are too many of them to be trained from a labeled data set of limited size. Thus the feature values are trained by approximate EM. The other kind of parameters is feature weights λ, which are trained by an error minimization method. The discriminative training of λ and the approximate EM training of Ψ are integrated into a semi-supervised training framework similar to EMD3 (Fraser and Marcu, 2006).

4.2 Discriminative Training of λ

MERT (Och, 2003) is used to train feature weights λ. MERT estimates model parameters with the objective of minimizing certain measure of translation errors (or maximizing certain performance measure of translation quality) for a development corpus. Given an SMT system which produces, with model parameters λ_1^M, the K-best candidate translations $\hat{e}(f_s;\lambda_1^M)$ for a source sentence f_s, and an error measure $E(r_s, e_{s,k})$ of a particular candidate $e_{s,k}$ with respect to the reference translation r_s, the optimal parameter values will be:

$$\hat{\lambda}_1^M = \underset{\lambda_1^M}{argmin}\left\{\sum_{s=1}^{S} E(r_s, \hat{e}(f_s;\lambda_1^M))\right\}$$
$$= \underset{\lambda_1^M}{argmin}\left\{\sum_{s=1}^{S}\sum_{k=1}^{K} E(r_s, e_{s,k})\delta(\hat{e}(f_s;\lambda_1^M), e_{s,k})\right\}$$

MERT for DITG applies the same equation for parameter tuning, with different interpretation of the components in the equation. Instead of a development corpus with reference translations, we have a collection of training samples, each of which is a sentence pair with annotated alignment result. The ITG parser outputs for each sentence pair a K-best list of alignment result $\hat{e}(f_s;\lambda_1^M)$ based on the current parameter values λ_1^M. The MERT module for DITG takes alignment F-score of a sentence pair as the performance measure. Given an input sentence pair and the reference annotated alignment, MERT aims to maximize the F-score of DITG-produced alignment.

4.3 Approximate EM Training of Ψ

Three kinds of features (introduced in section 4.5 and 4.6) are calculated from training corpus given some initial alignment result: conditional probability of word pairs and two types of conditional probabilities for phrase/h-phrase.

[3] For simplicity, we will also call our semi-supervised framework as EMD.

The initial alignment result is far from perfect and so the feature values thus obtained are not optimized. There are too many features to be trained in supervised way. So, unsupervised training like EM is the best solution.

When EM is applied to our model, the E-step corresponds to calculating the probability for all the ITG trees, and the M-step corresponds to re-estimate the feature values. As it is intractable to handle all possible ITG trees, instead we use the Viterbi parse to update the feature values. In other words, the training is a kind of approximate EM rather than EM.

Word pairs are collected over Viterbi alignment and their conditional probabilities are estimated by MLE. As to phrase/h-phrase, if they are handled in a similar way, then there will be data sparseness (as there are much fewer phrase/h-phrase pairs in Viterbi parse tree than needed for reliable parameter estimation). Thus, we collect all phrase/h-phrase pairs which are consistent with the alignment links. The conditional probabilities are then estimated by MLE.

4.4 Semi-supervised training

Algorithm EMD (semi-supervised training)

input development data *dev*, test data *test*, training data with initial alignment (*train, align_train*)

output feature weights λ and features Ψ.

estimate initial features Ψ^0 with (*train, align_train*)
get an initial weights λ^0 by MERT with the initial features Ψ^0 on *dev*.
get the F-Measure e^0 for λ^0 and Ψ^0 on *test*.
for(t=1;; t++)
    get the Viterbi alignment *align_train* for *train* using λ^{t-1} and Ψ^{t-1}
    estimate Ψ^t with (*train, align_train*)
    get new feature weights λ^t by MERT with Ψ^t on *dev*.
    get the F-Measure e^t for λ^t and Ψ^t on *test*.
    if $e^t \leq e^{t-1} + 0.1$ **then**
        break
end for
return λ^{t-1} and Ψ^{t-1}

Figure 5. Semi-supervised training for HP-DITG.

The discriminative training (error minimization) of feature weights λ and the approximate EM learning of feature values Ψ are integrated in a single semi-supervised framework. Given an initial estimation of Ψ (estimated from an initial alignment matrix by some simpler word alignment model) and an initial estimation of λ, the discriminative training process and the approximate EM learning process are alternatively iterated until there is no more improvement. The sketch of the semi-supervised training is shown in Figure 5.

4.5 Features for word pairs

The following features about alignment link are used in W-DITG:

1) Word pair translation probabilities trained from HMM model (Vogel et al., 1996) and IBM model 4 (Brown et al., 1993).
2) Conditional link probability (Moore, 2006).
3) Association score rank features (Moore et al., 2006).
4) Distortion features: counts of inversion and concatenation.

4.6 Features for phrase/h-phrase pairs

For our HP-DITG model, the rule probabilities in both English-to-foreign and foreign-to-English directions are estimated and taken as features, in addition to those features in W-DITG, in the discriminative model of alignment hypothesis selection:

1) $P(\bar{e}_i/\bar{f}_i)$: The conditional probability of English phrase/h-phrase given foreign phrase/h-phrase.
2) $P(\bar{f}_i/\bar{e}_i)$: The conditional probability of foreign phrase/h-phrase given English phrase/h-phrase.

The features are calculated as described in section 4.3.

5 Evaluation

Our experiments evaluate the performance of HP-DITG in both word alignment and translation in a Chinese-English setting, taking GIZA++, BerkeleyAligner (henceforth BERK) (Haghighi, et al., 2009), W-ITG as baselines. Word alignment quality is evaluated by recall, precision, and F-measure, while translation performance is evaluated by case-insensitive BLEU4.

5.1 Experiment Data

The small human annotated alignment set for discriminative training of feature weights is the same as that in Haghighi et al. (2009). The 491

sentence pairs in this dataset are adapted to our own Chinese word segmentation standard. 250 sentence pairs are used as training data and the other 241 are test data. The large, un-annotated bilingual corpus for approximate EM learning of feature values is FBIS, which is also the training set for our SMT systems.

In SMT experiments, our 5-gram language model is trained from the Xinhua section of the Gigaword corpus. The NIST'03 test set is used as our development corpus and the NIST'05 and NIST'08 test sets are our test sets. We use two kinds of state-of-the-art SMT systems. One is a phrase-based decoder (PBSMT) with a MaxEnt-based distortion model (Xiong, et al., 2006), and the other is an implementation of hierarchical phrase-based model (HPBSMT) (Chiang, 2007). The phrase/rule table for these two systems is not generated from the terminal node of HP-DITG tree directly, but extracted from word alignment matrix (HP-DITG generated) using the same criterion as most phrase-based systems (Chiang, 2007).

5.2 HP-DITG without EMD

Our first experiment isolates the contribution of the various DITG alignment models from that of semi-supervised training. The feature values of the DITG models are estimated simply from IBM Model 4 using GIZA++. Apart from DITG, P-ITG, and HP-ITG as introduced in Section 2, we also include a variation, known as H-DITG, which covers h-phrase pairs but no simple phrase pairs at all. The experiment results are shown in Table 1.

	Precision	Recall	F-Measure
GIZA++	0.826	0.807	0.816
BERK	**0.917**	**0.814**	**0.862**
W-DITG	0.912	0.745	0.820
P-DITG	0.913	0.788	0.846
H-DITG	0.913	0.781	0.842
HP-DITG	**0.915**	**0.795**	**0.851**

Table 1. Performance gains with features for HP-DITG.

It is obvious that any form of ITG achieves better F-Measure than GIZA++. Without semi-supervised training, however, our various DITG models cannot compete with BERK. Among the DITG models, it can be seen that precision is roughly the same in all cases, while W-ITG has the lowest recall, due to the limitation of one-to-one matching. The improvement by (simple) phrase pairs is roughly the same as that by h-phrase pairs. And it is not surprising that the combination of both kinds of phrases achieve the best result.

Even HP-DITG does not achieve as high recall as BERK, it does produce promising alignment patterns that BERK fails to produce. For instance, for the following sentence pair:

自 上周末 以来， 我 一直 在 生病 .

I have been ill ***since*** last weekend .

Both GIZA++ and BERK produce the pattern in Figure 6(a), while HP-DITG produces the better pattern in Figure 6(b) as it learns the h-phrase pair "since *X*"/"自 *X* 以来".

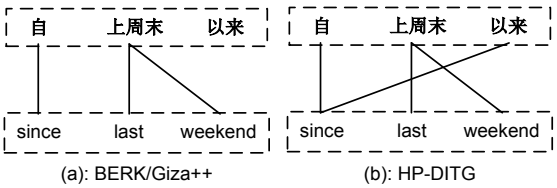


Figure 6. Partial alignment results.

5.3 Alignment Quality of HP-DITG with EMD

	Precision	Recall	F- Measure
GIZA++	0.826	0.807	0.816
BERK	0.917	0.814	0.862
EMD0	0.915	0.795	0.851
EMD1	0.923	0.814	0.865
EMD2	0.930	**0.821**	0.872
EMD3	**0.935**	0.819	**0.873**

Table 2. Semi-supervised Training Task on F-Measure.

The second experiment evaluates how the semi-supervised method of EMD improves HP-DITG with respect to word alignment quality. The results are shown in Table 2. In the table, EMD0 refers to the HP-DITG model before any EMD training; EMD1 refers to the model after the first iteration of training, and so on. It is empirically found that F-Measure is not improved after the third EMD iteration.

It can be observed that EMD succeeds to help HP-DITG improves feature value and weight estimation iteratively. When semi-supervised

training converges, the new HP-DITG model is better than before training by 2%, and better than BERK by 1%.

5.4 Translation Quality of HP-DITG with EMD

The third experiment evaluates the same alignment models in the last experiment but with respect to translation quality, measured by case-insensitive BLEU4. The results are shown in Table 3. Note that the differences between EMD3 and the two baselines are statistically significant.

	PBSMT		HPBSMT	
	05	08	05	08
GIZA++	33.43	23.89	33.59	24.39
BERK	33.76	24.92	34.22	25.18
EMD0	34.02	24.50	34.30	24.90
EMD1	34.29	24.80	34.77	25.25
EMD2	*34.25*	25.01	**35.04**	25.43
EMD3	**34.42**	**25.19**	*34.82*	**25.56**

Table 3. Semi-supervised Training Task on BLEU.

It can be observed that EMD improves SMT performance in most iterations in most cases. EMD does not always improve BLEU score because the objective function of the discriminative training in EMD is about alignment F-Measure rather than BLEU. And it is well known that the correlation between F-Measure and BLEU (Fraser and Marcu, 2007) is itself an intriguing problem.

The best HP-DITG leads to more than 1 BLEU point gain compared with GIZA++ on all datasets/MT models. Compared with BERK, EMD3 improves SMT performance significantly on NIST05 and slightly on NIST08.

6 Conclusion and Future Work

In this paper, we propose an ITG formalism which employs the notion of phrase/h-phrase pairs, in order to remove the limitation of one-to-one matching. The formalism is proved to enable an alignment model to capture the linguistic fact that a single concept is expressed in several non-contiguous words. Based on the formalism, we also propose a semi-supervised training method to optimize feature values and feature weights, which does not only improve the alignment quality but also machine translation performance significantly. Combining the formalism and semi-supervised training, we obtain better alignment and translation than the baselines of GIZA++ and BERK.

A fundamental problem of our current framework is that we fail to obtain monotonic increment of BLEU score during the course of semi-supervised training. In the future, therefore, we will try to take the BLEU score as our objective function in discriminative training. That is to certain extent inspired by Deng et al. (2008).

Appendix A. The Normal Form Grammar

Table 4 lists the ITG rules in normal form as used in this paper, which extend the normal form in Wu (1997) so as to handle the case of alignment to null.

1	$S \rightarrow A \mid B \mid C$
2	$A \rightarrow [A\ B] \mid [A\ C] \mid [B\ B] \mid [BC] \mid [C\ B] \mid [C\ C]$
3	$B \rightarrow \langle A\ A\rangle \mid \langle A\ C\rangle \mid \langle B\ A\rangle \mid \langle B\ C\rangle$
	$B \rightarrow \langle C\ A\rangle \mid \langle C\ C\rangle$
4	$C \rightarrow C_w \mid C_{fw} \mid C_{ew}$
5	$C \rightarrow [C_{ew}\ C_{fw}]$
6	$C_w \rightarrow u/v$
7	$C_e \rightarrow \varepsilon/v;\ C_f \rightarrow u/\varepsilon$
8	$C_{em} \rightarrow C_e \mid [C_{em}\ C_e];\ C_{fm} \rightarrow C_f \mid [C_{fm}\ C_f]$
9	$C_{ew} \rightarrow [C_{em}\ C_w];\ C_{fw} \rightarrow [C_{fm}\ C_w]$

Table 4. ITG Rules in Normal Form.

In these rules, S is the Start symbol; A is the category for concatenating combination whereas B for inverted combination. Rules (2) and (3) are inherited from Wu (1997). Rules (4) divide the terminal category C into subcategories. Rule schema (6) subsumes all terminal unary rules for some English word u and foreign word v, and rule schemas (7) are unary rules for alignment to null. Rules (8) ensure all words linked to null are combined in left branching manner, while rules (9) ensure those words linked to null combine with some following, rather than preceding, word pair. (Note: Accordingly, all sentences must be ended by a special token $\langle end\rangle$, otherwise the last word(s) of a sentence cannot be linked to null.) If there are both English and foreign words linked to null, rule (5) ensures that those English words linked to null precede those foreign words linked to null.

References

Birch, Alexandra, Chris Callison-Burch, Miles Osborne and Phillipp Koehn. 2006. Constraining the Phrase-Based, Joint Probability Statistical Translation Model. *Proceedings of the Workshop on Statistical Machine Translation*.

Brown, Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Peitra, Robert L. Mercer. 1993. The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics,* 19(2):263-311.

Cherry, Colin and Dekang Lin. 2006. Soft Syntactic Constraints for Word Alignment through Discriminative Training. Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics.

Cherry, Colin and Dekang Lin. 2007. Inversion Transduction Grammar for Joint Phrasal Translation Modeling. *Proceedings of the Second Workshop on Syntax and Structure in Statistical Translation*, Pages:17-24.

Chiang, David. 2007. Hierarchical Phrase-based Translation. *Computational Linguistics*, 33(2).

Deng, Yonggang, Jia Xu and Yuqing Gao. 2008. Phrase Table Training For Precision and Recall: What Makes a Good Phrase and a Good Phrase Pair?. *Proceedings of the 7th International Conference on Human Language Technology Research and 46th Annual Meeting of the Association for Computational Linguistics*, Pages:1017-1026.

Fraser, Alexander, Daniel Marcu. 2006. Semi-Supervised Training for StatisticalWord Alignment. Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, Pages:769-776.

Fraser, Alexander, Daniel Marcu. 2007. Measuring Word Alignment Quality for Statistical Machine Translation. *Computational Linguistics*, 33(3).

Haghighi, Aria, John Blitzer, John DeNero, and Dan Klein. 2009. Better Word Alignments with Supervised ITG Models. Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language, Pages: 923-931.

Klein, Dan and Christopher D. Manning. 2001. Parsing and Hypergraphs. *Proceedings of the 7th International Workshop on Parsing Technologies*, Pages:17-19

Liu, Yang, Qun Liu and Shouxun Lin. 2005. Log-linear models for word alignment. *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, Pages: 81-88.

Marcu, Daniel, William Wong. 2002. A Phrase-Based, Joint Probability Model for Statistical Machine Translation. *Proceedings of 2002 Conference on Empirical Methods in Natural Language Processing*, Pages:133-139.

Moore, Robert, Wen-tau Yih, and Andreas Bode. 2006. Improved Discriminative Bilingual Word Alignment. *Proceedings of the 44rd Annual Meeting of the Association for Computational Linguistics*, Pages: 513-520.

Och, Franz Josef. 2003. Minimum error rate training in statistical machine translation. *Proceedings of the 41rd Annual Meeting of the Association for Computational Linguistics*, Pages:160-167.

Och, Franz Josef and Hermann Ney. 2004. The Alignment Template Approach to Statistical Machine Translation. *Computational Linguistics*, 30(4) : 417-449.

Vogel, Stephan, Hermann Ney, and Christoph Tillmann. 1996. HMM-based word alignment in statistical translation. *Proceedings of 16th International Conference on Computational Linguistics*, Pages: 836-841.

Wu, Dekai. 1997. Stochastic Inversion Transduction Grammars and Bilingual Parsing of Parallel Corpora. *Computational Linguistics*, 23(3).

Xiong, Deyi, Qun Liu and Shouxun Lin. 2006. Maximum entropy based phrase reordering model for statistical machine translation. *Proceedings of the 44rd Annual Meeting of the Association for Computational Linguistics*, Pages: 521-528.

Zhang, Hao and Daniel Gildea. 2005. Stochastic Lexicalized Inversion Transduction Grammar for Alignment. Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics.

Zhang, Hao, Chris Quirk, Robert Moore, and Daniel Gildea. 2008. Bayesian learning of non-compositional phrases with synchronous parsing. *Proceedings of the 46rd Annual Meeting of the Association for Computational Linguistics*, Pages: 314-323.

Visually and Phonologically Similar Characters in Incorrect Simplified Chinese Words

Chao-Lin Liu[†] Min-Hua Lai[‡] Yi-Hsuan Chuang[↑] Chia-Ying Lee[↕]
[†‡↑]Department of Computer Science; [†↕]Center for Mind, Brain, and Learning
National Chengchi University
[↕]Institute of Linguistics, Academia Sinica
{[†]chaolin, [‡]g9523, [↑]g9804}@cs.nccu.edu.tw, [↕]chiaying@gate.sinica.edu.tw

Abstract

Visually and phonologically similar characters are major contributing factors for errors in Chinese text. By defining appropriate similarity measures that consider extended Cangjie codes, we can identify visually similar characters within a fraction of a second. Relying on the pronunciation information noted for individual characters in Chinese lexicons, we can compute a list of characters that are phonologically similar to a given character. We collected 621 incorrect Chinese words reported on the Internet, and analyzed the causes of these errors. 83% of these errors were related to phonological similarity, and 48% of them were related to visual similarity between the involved characters. Generating the lists of phonologically and visually similar characters, our programs were able to contain more than 90% of the incorrect characters in the reported errors.

1 Introduction

In this paper, we report the experience of our studying the errors in simplified Chinese words. Chinese words consist of individual characters. Some words contain just one character, but most words comprise two or more characters. For instance, "卖" (mai4)[1] has just one character, and "语言" (yu3 yan2) is formed by two characters. Two most common causes for writing or typing incorrect Chinese words are due to visual and phonological similarity between the correct and the incorrect characters. For instance, one might use "划" (hwa2) in the place of "画"(hwa4) in "刻画形象" (ke1 hwa4 xing2 xiang4) partially because of phonological similarity; one might replace "拙" (zhuo2) in "心劳力拙" (xin1 lao2 li4 zhuo2) with "绌" (chu4) partially due to visual similarity. (We do not claim that the visual or phonological similarity alone can explain the observed errors.)

Similar characters are important for understanding the errors in both traditional and simplified Chinese. Liu et al. (2009a-c) applied techniques for manipulating correctness of Chinese words to computer assisted test-item generation. Research in psycholinguistics has shown that the number of neighbor characters influences the timing of activating the mental lexicon during the process of understanding Chinese text (Kuo et al. 2004; Lee et al. 2006). Having a way to compute and find similar characters will facilitate the process of finding neighbor words, so can be instrumental for related studies in psycholinguistics. Algorithms for optical character recognition for Chinese and for recognizing written Chinese try to guess the input characters based on sets of confusing sets (Fan et al. 1995; Liu et al., 2004). The confusing sets happen to be hand-crafted clusters of visually similar characters.

It is relatively easy to judge whether two characters have similar pronunciations based on their records in a given Chinese lexicon. We will discuss more related issues shortly.

To determine whether two characters are visually similar is not as easy. Image processing techniques may be useful but is not perfectly feasible, given that there are more than fifty thousand Chinese characters (HanDict, 2010) and that many of them are similar to each other in special ways. Liu et al. (2008) extend the Cangjie codes (Cangjie, 2010; Chu, 2010) to encode the layouts and details about traditional

[1] We show simplified Chinese characters followed by their Hanyu pinyin. The digit that follows the symbols for the sound is the tone for the character.

Coling 2008: Poster Volume, pages 739–747,
Beijing, August 2010

Chinese characters for computing visually similar characters. Evidence observed in psycholinguistic studies offers a cognition-based support for the design of Liu et al.'s approach (Yeh and Li, 2002). In addition, the proposed method proves to be effective in capturing incorrect traditional Chinese words (Liu et al., 2009a-c).

In this paper, we work on the errors in simplified Chinese words by extending the Cangjie codes for simplified Chinese. We obtain two lists of incorrect words that were reported on the Internet, analyze the major reasons that contribute to the observed errors, and evaluate how the new Cangjie codes help us spot the incorrect characters. Results of our analysis show that phonological and visual similarities contribute similar portions of errors in simplified and traditional Chinese. Experimental results also show that, we can catch more than 90% of the reported errors.

We go over some issues about phonological similarity in Section 2, elaborate how we extend and apply Cangjie codes for simplified Chinese in Section 3, present details about our experiments and observations in Section 4, and discuss some technical issues in Section 5.

2 Phonologically Similar Characters

The pronunciation of a Chinese character involves a sound, which consists of the nucleus and an optional onset, and a tone. In Mandarin Chinese, there are four tones. (Some researchers include the fifth tone.)

In our work, we consider four categories of phonological similarity between two characters: same sound and same tone (**SS**), same sound and different tone (**SD**), similar sound and same tone (**MS**), and similar sound and different tone (**MD**).

We rely on the information provided in a lexicon (Dict, 2010) to determine whether two characters have the same sound or the same tone. The judgment of whether two characters have similar sound should consider the language experience of an individual. One who live in the southern and one who live in the northern China may have quite different perceptions of "similar" sound. In this work, we resort to the confusion sets observed in a psycholinguistic study conducted at the Academic Sinica.

Some Chinese characters are heteronyms. Let C_1 and C_2 be two characters that have multiple pronunciations. If C_1 and C_2 share one of their pronunciations, we consider that C_1 and C_2 belong to the SS category. This principle applies when we consider phonological similarity in other categories.

One challenge in defining similarity between characters is that the pronunciations of a character can depend on its context. The most common example of tone sandhi in Chinese (Chen, 2000) is that the first third-tone character in words formed by two adjacent third-tone characters will be pronounced in the second tone. At present, we ignore the influences of context when determining whether two characters are phonologically similar.

Although we have confined our definition of phonological similarity to the context of the Mandarin Chinese, it is important to note the influence of sublanguages within the Chinese language family will affect the perception of phonological similarity. Sublanguages used in different areas in China, e.g., Shanghai, Min, and Canton share the same written forms with the Mandarin Chinese, but have quite different though related pronunciation systems. Hence, people living in different areas in China may perceive phonological similarity in very different ways. The study in this direction is beyond the scope of the current study.

3 Visually Similar Characters

Figure 1 shows four groups of visually similar characters. Characters in group 1 and group 2 differ subtly at the stroke level. Characters in group 3 share the components on their right sides. The shared component of the characters in group 4 appears at different places within the characters.

Radicals are used in Chinese dictionaries to organize characters, so are useful for finding visually similar characters. The characters in group 1 and group 2 belong to the radicals "田" and "讠", respectively. Notice that, although the radical for group 2 is clear, the radical for group 1 is not obvious because "田" is not a standalone component.

However, the shared components might not be the radicals of characters. The shared components in groups 3 and 4 are not the radicals. In

Figure 1. Examples of visually similar characters

many cases, radicals are semantic components of Chinese characters. In groups 3 and 4, the shared components carry information about the pronunciations of the characters. Hence, those characters are listed under different radicals, though they do look similar in some ways.

Hence, a mechanism other than just relying on information about characters in typical lexicons is necessary, and we will use the extended Cangjie codes for finding visually similar characters.

3.1 Cangjie Codes for Simplified Chinese

Table 1 shows the Cangjie codes for the 13 characters listed in Figure 1 and five other characters. The "ID" column shows the identification number for the characters, and we will refer to the i^{th} character by c_i, where i is the ID. The "CC" column shows the Chinese characters, and the "Cangjie" column shows the Cangjie codes. Each symbol in the Cangjie codes corresponds to a key on the keyboard, e.g. "田" and "中" collocate with "W" and "L", respectively. Information about the complete correspondence is available on the Wikipedia[2].

Using the Cangjie codes saves us from using image processing methods to determine the degrees of similarity between characters. Take the Cangjie codes for the characters in group 2 (c_5, c_6, and c_7) for example. It is possible to find that the characters share a common component, based on the shared substrings of the Cangjie codes, i.e., "戈女". Using the common substring (shown in black bold) of the Cangjie codes, we may also find the shared component "勾" for characters in group 3 (c_{10}, c_{11}, and c_{12}), the shared component "员" in c_{13} and c_{14}, the shared component "力" in c_{15} and c_{16}, and the shared component "⻥" in c_{16} and c_{17}.

Despite the perceivable advantages, these original Cangjie codes are not good enough. In order to maintain efficiency in inputting Chinese characters, the Cangjie codes have been limited to no more than five keys. Thus, users of the Cangjie input method must familiarize themselves with the principles for simplifying the Cangjie codes. While the simplified codes help the input efficiency, they also introduce difficulties and ambiguities when we compare the Cangjie codes for computing similar characters. The prefix "弓一" in c_{16} and c_{17} can represent "⻥", "鱼" (e.g., c_8), and "马" (e.g., c_9). Characters whose Cangjie codes include "弓一" may contain any of these three components, but they do not really look alike.

ID	CC	Cangjie	ID	CC	Cangjie
1	田	田	10	购	月人**心戈**
2	由	中田	11	沟	水**心戈**
3	甲	田中	12	构	木**心戈**
4	申	中田中	13	员	**口月人**
5	许	**戈女**人十	14	圆	田**口月人**
6	讦	**戈女**一十	15	勋	**口人大尸**
7	计	**戈女**十	16	劲	**弓一大尸**
8	鲳	**弓一**日日	17	颈	**弓一**一月人
9	驹	**弓一**心口	18	经	女一**弓人一**

Table 1. Examples of Cangjie codes

Therefore, we augment the original Cangjie codes by using the complete Cangjie codes and annotate each Chinese character with a layout identification that encodes the overall contours of the characters. This is how Liu and his colleagues (2008) did for the Cangjie codes for traditional Chinese characters, and we employ a similar exploration for the simplified Chinese.

3.2 Augmenting the Cangjie Codes

Figure 2 shows the twelve possible layouts that are considered for the Cangjie codes for simplified Chinese characters. Some of the layouts contain smaller areas, and the rectangles show a subarea within a character. The smaller areas are assigned IDs between one and three. Notice that, to maintain read-ability of the figures, not all IDs for subareas are shown in Figure 2. An example character is provided below each layout. From left to right and from top to bottom, each layout is assigned an identification number from 1 to 12. For example, the layout ID of "国" is 8. "国" has two parts, i.e., "口" and "玉".

Researchers have come up with other ways to

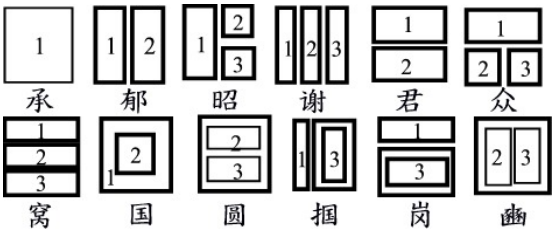


Figure 2. Layouts of Chinese characters

[2] en.wikipedia.org/wiki/Cangjie_input_method#Keyboard_layout ; last visited on 22 April 2010.

decompose individual Chinese characters. The Chinese Document Lab at the Academia Sinica proposed a system with 13 operators for describing the relationships among components in Chinese characters (CDL, 2010). Lee (2010b) propose more than 30 possible layouts.

The layout of a character affects how people perceive visual similarity between characters. For instance, c_{16} in Table 1 is more similar to c_{17} than to c_{18}, although they share "⿰". We rely on the expertise in Cangjie codes reported in (Lee, 2010a) to split the codes into parts.

Table 2 shows the extended codes for some characters listed in Table 1. The "ID" column provides links between the characters listed in both Table 1 and Table 2. The "CC" column shows the Chinese characters. The "LID" column shows the identifications for the layouts of the characters. The columns with headings "P1", "P2", and "P3" show the extended Cangjie codes, where "P*i*" shows the i^{th} part of the Cangjie codes, as indicated in Figure 2.

We decide the extended codes for the parts with the help of computer programs and subjective judgments. Starting from the original Cangjie codes, we can compute the most frequent substrings just like we can compute the frequencies of n-grams in corpora (cf. Jurafsky and Martin, 2009). Computing the most common substrings in the original codes is not a complex task because the longest original Cangjie codes contain just five symbols.

Often, the frequent substrings are simplified codes for popular components in Chinese characters, e.g., "⿰" and "⿰". The original codes for "⿰" and "⿰" are "戈弓女" and "弓人一", but they are often simplified to "戈女" and "弓一", respectively. When simplified, "⿰" have the same Cangjie code with "戌", and "⿰" have the same Cangjie code with "马" and "鱼".

After finding the frequent substrings, we verify whether these frequent substrings are simplified codes for meaningful components. For meaningful components, we replace the simplified codes with complete codes. For instance the Cangjie codes for "许" and "讦" are extended to include "弓" in Table 2, where we indicate the extended keys that did not belong to the original Cangjie codes in boldface and with a surrounding box. Most of the non-meaningful frequent substrings have two keys: one is the last key of a part, and the other is the first key of another part. They were by observed by coincidence.

ID	CC	LID	P1	P2	P3
5	许	2	戈**弓**女	人十	
6	讦	2	戈**弓**女	一十	
7	计	2	戈**弓**女	十	
10	购	10	月人	心	戈
11	沟	10	水	心	戈
12	构	10	木	心	戈
13	员	5	口	月人	
14	圆	9	田	口	月人
15	勋	2	口**月**人	大尸	
16	劲	2	弓**人**一	大尸	
17	颈	2	弓**人**一	一月人	
18	经	3	女**女**一	弓人	一
19	恸	4	心	一**一**戈	**大**尸

Table 2. Examples of extended Cangjie codes

Although most of the examples provided in Table 2 indicate that we expand only the first part of the Cangjie codes, it is absolutely possible that the other parts, i.e., P2 and P3, may need to be extended too. c_{19} shows such an example.

Replacing simplified codes with complete codes not only help us avoid incorrect matches but also help us find matches that would be missed due to simplification of Cangjie codes. Using just the original Cangjie codes in Table 1, it is not easy to determine that c_{18} ("经") in Table 1 shares a component ("⿰") with c_{16} and c_{17} ("劲" and "颈"). In contrast, there is a chance to find the similarity with the extended Cangjie codes in Table 2, given that all of the three Cangjie codes include "弓人一".

We can see an application of the LIDs, using "劲", "颈" and "经" as an example. Consider the case that we want to determine which of "颈" and "经" is more similar to "劲". Their extended Cangjie codes will indicate that "颈" is the answer to this question for two reasons. First, "劲" and "颈" belong to the same type of layout; and, second, the shared components reside at the same area in "劲" and "颈".

3.3 Similarity Measures

The main differences between the original and the extended Cangjie codes are the degrees of details about the structures of the Chinese characters. By recovering the details that were ignored in the original codes, our programs will be

better equipped to find the similarity between characters.

In the current study, we experiment with three different scoring methods to measure the visual similarity between two characters based on their extended Cangjie codes. Two of these methods had been tried by Liu and his colleagues' study for traditional Chinese characters (Liu et al., 2009b-c). The first method, denoted **SC1**, considers the total number of matched keys in the matched parts (without considering their part IDs). Let c_i denote the i^{th} character listed in Table 2. We have SC1(c_{15}, c_{16}) = 2 because of the matched "大尸". Analogously, we have SC1(c_{19}, c_{16}) = 2.

The second method, denoted **SC2**, includes the score of SC1 and considers the following conditions: (1) add one point if the matched parts locate at the same place in the characters and (2) if the first condition is met, an extra point will be added if the characters belong to the same layout. Hence, we have SC2(c_{15}, c_{16}) =SC1(c_{15}, c_{16})+1+1=4 because (1) the matched "大尸" locate at P2 in both characters and (2) c_{15} and c_{16} belong to the same layout. Assuming that c_{16} belongs to layout 5, than SC2(c_{15}, c_{16}) would become 3. In contrast, we have SC2(c_{19}, c_{16})=2. No extra weights for the matching "大尸" because it locates at different parts in the characters. The extra weight considers the spatial influences of the matched parts on the perception of similarity.

While splitting the extended Cangjie codes into parts allows us to tell that c_{15} is more similar to c_{16} than to c_{19}, it also creates a new barrier in computing similarity scores. An example of this problem is that SC2(c_{17}, c_{18})=0. This is because that "弓人一" at P1 in c_{17} can match neither "弓人" at P2 nor "一" at P3 in c_{18}.

To alleviate this problem, we consider **SC3** which computes the similarity in three steps. First, we concatenate the parts of a Cangjie code for a character. Then, we compute the longest common subsequence (**LCS**) (cf. Cormen et al., 2009) of the concatenated codes of the two characters being compared, and compute a Dice's coefficient (cf. Croft et al., 2010) as the similarity. Let X and Y denote the concatenated, extended Cangjie codes for two characters, and let Z be the LCS of X and Y. The similarity is defined by the following equation.

$$Dice_{LCS} = \frac{2 \times |Z|}{|X| + |Y|}, \text{ where } |S| \text{ is the length of string } S \quad (1)$$

We compute another Dice's coefficient between X and Y. The formula is the similar to (1), except that we set Z to the longest common *consecutive* subsequence. We call this score $Dice_{LCCS}$. Notice that $Dice_{LCCS} \le Dice_{LCS}$, $Dice_{LCCS} \le 1$, and $Dice_{LCS} \le 1$. Finally, SC3 of two characters is the sum of their SC2, $10 \times Dice_{LCCS}$, and $5 \times Dice_{LCS}$. We multiply the Dice's coefficients with constants to make them as influential as the SC2 component in SC3. The constants were not scientifically chosen, but were selected heuristically.

4 Error Analysis and Evaluation

We evaluate the effectiveness of using the phonologically and visually similar characters to captures errors in simplified Chinese words with two lists of reported errors that were collected from the Internet.

4.1 Data Sources

We need two types of data for the experiments. The information about the pronunciation and structures of the Chinese characters help us generate lists of similar characters. We also need reported errors so that we can evaluate whether the similar characters catch the reported errors.

A lexicon that provides the pronunciation information about Chinese characters and a database that contains the extended Cangjie codes are necessary for our programs to generate lists of characters that are phonologically and visually similar to a given character.

It is not difficult to acquire lexicons that show standard pronunciations for Chinese characters. As we stated in Section 2, the main problem is that it is not easy to predict how people in different areas in China actually pronounce the characters. Hence, we can only rely on the standards that are recorded in lexicons.

With the procedure reported in Section 3.2, we built a database of extended Cangjie codes for the simplified Chinese. The database was designed to contain 5401 common characters in the BIG5 encoding, which was originally designed for the traditional Chinese. After converting the traditional Chinese characters to the simplified counterparts, the database contained only 5170

different characters.

We searched the Internet for reported errors that were collected in real-world scenarios, and obtained two lists of errors. The first list[3] came from the entrance examinations for senior high schools in China, and the second list[4] contained errors observed at senior high schools in China. We used 160 and 524 errors from the first and the second lists, respectively, and we refer to the combined list as the **Ilist**. An item of reported error contained two parts: the correct word and the mistaken character, both of which will be used in our experiments.

4.2 Preliminary Data Analysis

Since our programs can compare the similarity only between characters that are included in our lexicon, we have to exclude some reported errors from the Ilist. As a result, we used only 621 errors in this section.

Two native speakers subjectively classified the causes of these errors into three categories based on whether the errors were related to phonological similarity, visual similarity, or neither. Since the annotators did not always agree on their classifications, the final results have five interesting categories: "P", "V", "N", "D", and "B" in Table 3. P and V indicate that the annotators agreed on the types of errors to be related to phonological and visual similarity, respectively. N indicates that the annotators believed that the errors were not due to phonological or visual similarity. D indicates that the annotators believed that the errors were due to phonological or visual similarity, but they did not have a consensus. B indicates the intersection of P and V.

Table 3 shows the percentages of errors in these categories. To get 100% from the table, we can add up P, V, N, and D, and subtract B from the total. In reality there are errors of type N, and Liu and his colleagues (2009b) reported this type of errors. Errors in this category happened to be missing in the Ilist. Based on our and Liu's observations, the percentages of phonological and visual similarities contribute to the errors in simplified and traditional Chinese words with similar percentages.

	P	V	N	D	B
Ilist	83.1	48.3	0	3.7	35.1

Table 3. Percentages of types of errors

[3] www.0668edu.com/soft/4/12/95/2008/20080913571140.htm ; last visited on 22 April 2010.

[4] gaozhong.kt5u.com/soft/2/38018.html; last visited on 22 April 2010.

4.3 Experimental Procedure

We design and employ the `ICCEval` procedure for the evaluation task.

At step 1, given the correct word and the correct character to be intentionally replaced with incorrect characters, we created a list of characters based on the selection criterion. We may choose to evaluate phonologically or visually similar characters. For a given character, `ICCEval` can generate characters that are in the SS, SD, MS, and MD categories for phonologically similar characters (cf. Section 2). For visually similar characters, `ICCEval` can select characters based on SC1, SC2, and SC3 (cf. Section 3.3). In addition, `ICCEval` can generate a list of characters that belong to the same radical and have the same number of strokes with the correct character. In the experimental results, we refer to this type of similar characters as **RS**.

At step 2, for a correct word that people originally wanted to write, we replaced the correct character with an incorrect character with the characters that were generated at step 1, submitted the incorrect word to Google AJAX Search

```
Procedure ICCEval
Input:
  ccr: the correct character; cwd:
  the correct word; crit: the selec-
  tion criterion; num: number of re-
  quested characters; rnk: the cri-
  terion to rank the incorrect
  words;
Output: a list of ranked candidates
    for ccr
Steps:
1.  Generate a list, L, of charac-
    ters for ccr with the specified
    criterion, crit. When using SC1,
    SC2, or SC3 to select visually
    similar characters, at most num
    characters will be selected.
2.  For each c in L, replace ccr in
    cwd with c, submit the resulting
    incorrect word to Google, and
    record the ENOP.
3.  Rank the list of incorrect words
    generated at step 2, using the
    criterion specified by rnk.
4.  Return the ranked list.
```

API, and extracted the estimated numbers of pages (ENOP) [5] that contained the incorrect words. In an ordinary interaction with Google, an ENOP can be retrieved from the search results, and it typically follows the string "`Results 1-10 of about`" on the upper part of the browser window. Using the AJAX API, we just have to parse the returned results with a simple method.

Larger ENOPs for incorrect words suggest that these words are incorrect words that people frequently used on their web pages. Hence, we ranked the similar characters based on their ENOPs at step 3, and return the list.

Since the reported errors contained information about the incorrect ways to write the correct words, we could check whether the real incorrect characters were among the similar characters that our programs generated at step 1 (inclusion tests). We could also check whether the actual incorrect characters were ranked higher in the ranked lists (ranking tests).

Take the word "和藹可亲" as an example. In the collected data, it is reported that people wrote this word as "和靄可亲", i.e., the second character was incorrect. Hoping to capture the error, `ICCEval` generated a list of possible substitutions for "藹" at step 1. Depending on the categories of sources of errors, `ICCEval` generated a list of characters. When aiming to test the effectiveness of visually similar characters, we could ask `ICCEval` to apply SC3 to generate a list of alternatives for "藹", possibly including "靄", "谒", "葛", and other candidates. At step 2, we created and submitted query strings "和靄可亲", "和谒可亲", and "和葛可亲" to obtain the ENOPs for the candidates. If the ENOPs were, respectively, 410000, 26100, and 7940, these candidates would be returned in the order of "靄", "谒", and "葛". As a result, the returned list contained the actual incorrect character "靄", and placed "靄" on top of the ranked list.

Notice that we considered the contexts in which the incorrect characters appeared to rank. We did not rank the incorrect characters with just the unigrams. In addition, although this running example shows that we ranked the characters directly with the ENOPs, we also ranked the list of alternatives with pointwise mutual information:

$$PMI(C,X) = \frac{\Pr(C \wedge X)}{\Pr(C) \times \Pr(X)}, \quad (2)$$

where X is the candidate character to replace the correct character and C is the correct word excluding the correct character to be replaced. To compute the score of replacing "藹" with "靄" in "和藹可亲", X = "靄", C="和□可亲", and ($C \wedge X$) is "和靄可亲". (□ denotes a character to be replaced.) PMI is a common tool for judging collocations in natural language processing. (cf. Jurafsky and Martin, 2009).

It would demand very much computation effort to find Pr(C). Fortunately, we do not have to consider Pr(C) because it is a common denominator for all incorrect characters. Let X_1 and X_2 be two competing candidates for the correct character. We can ignore Pr(C) because of the following relationship.

$$PMI(C,X_1) \geq PMI(C,X_2) \Leftrightarrow \frac{\Pr(C \wedge X_1)}{\Pr(X_1)} \geq \frac{\Pr(C \wedge X_2)}{\Pr(X_2)}$$

Hence, X_1 prevails if $score(C,X_1)$ is larger.

$$score(C,X) = \frac{\Pr(C \wedge X)}{\Pr(X)} \quad (3)$$

In our work, we approximate the probabilities used in (3) by the corresponding frequencies that we can collect through Google, similar to the methods that we used to collect the ENOPs.

4.4 Experimental Results: Inclusion Tests

We ran `ICCEval` with 621 errors in the Ilist. The experiments were conducted for all categories of phonological and visual similarity. When using SS, SD, MS, MD, and RS as the selection criterion, we did not limit the number of candidate characters. When using SC1, SC2, and SC3 as the criterion, we limited the number candidates to be no more than 30. We consider only words that the native speakers have consensus over the causes of errors. Hence, we dropped those 3.7% of words in Table 3, and had just 598 errors. The ENOPs were obtained during March and April 2010.

Table 4 shows the chances that the lists, gen-

	SS	SD	MS	MD	Phone
Ilist	82.6	29.3	1.7	1.6	97.3
	SC1	SC2	SC3	RS	Visual
Ilist	78.3	71.0	87.7	1.3	90.0

Table 4. Chances of the recommended list contains the incorrect character

[5] According to (Croft et al., 2010), the ENOPs may not reflect the actual number of pages on the Internet.

erated with different `crit` at step 1, contained the incorrect character in the reported errors. In the Ilist, there were 516 and 300[6] errors that were related to phonological and visual similarity, respectively. Using the characters generated with the SS criterion, we captured 426 out of 516 phone-related errors, so we showed 426/516 = 82.6% in the table.

Results in Table 4 show that we captured phone-related errors more effectively than visually-similar errors. With a simple method, we can compute the union of the characters that were generated with the SS, SD, MS, and MD criteria. This integrated list suggested how well we captured the errors that were related to phones, and we show its effectiveness under "Phone". Similarly, we integrated the lists generated by SC1, SC2, SC3, and RS to explore the effectiveness of finding errors that are related to visual similarity, and the result is shown under "Visual".

4.5 Experimental Results: Ranking Tests

To put the generated characters into work, we wish to put the actual incorrect character high in the ranked list. This will help the efficiency in supporting computer assisted test-item writing. Having short lists that contain relatively more confusing characters may facilitate the data preparation for psycholinguistic studies.

At step 3, we ranked the candidate characters by forming incorrect words with other characters in the correct words as the context and submitted the words to Google for ENOPs. The results of ranking, shown in Table 5, indicate that we may just offer the leading five candidates to cover the actual incorrect characters in almost all cases.

The "Total" column shows the total number of errors that were captured by the selection criterion. The column "R*i*" shows the percentage of all errors, due to phonological or visual similarity, that were re-created and ranked i^{th} at step 3 in `ICCEVAL`. The row headings show the selection criteria that were used in the experiments. For instance, using SS as the criterion, 70.3% of actual phone-related errors were rank first, 7.4% of the phone-related errors were ranked second, etc. If we recommended only 5 leading incorrect characters only with SS, we would have captured the actual incorrect characters that were phone related 81.6% (the sum of R1 to R5) of the time. For errors that were related to visual similarity, recommending the top five candidates with SC3 would capture the actual incorrect characters 87.1% of the time. Since we do not show the complete distributions, the sums over the rows are not 100%. In the current experiments, the worst rank was 21.

	Total	R1	R2	R3	R4	R5
SS	426	70.3	7.4	2.9	0.4	0.6
SD	151	25.6	2.7	0.6	0.0	0.4
MS	9	1.4	0.4	0.0	0.0	0.0
MD	8	1.6	0.0	0.0	0.0	0.0
SC1	235	61.3	10.3	4.3	2.0	0.3
SC2	213	53.7	11.0	3.7	2.3	0.3
SC3	263	66.7	12.7	5.7	1.7	0.3
RS	4	1.3	0.0	0.0	0.0	0.0

Table 5. Ranking the candidates

We also used PMI to rank the incorrect words. Due to page limits, we cannot show complete details about the results. The observed distributions in ranks were not very different from those shown in Table 5.

5 Discussion

Compared with Liu et al.'s analysis (2009b-c) for the traditional Chinese, the proportions of errors related to phonological factors are almost the same, both at about 80%. The proportion of errors related to visual factors varied, but the averages in both studies were about 48%. A larger scale of study is needed for how traditional and simplified characters affect the distributions of errors. Results shown in Table 4 suggest that it is relatively easy to capture errors related to visual factors in simplified Chinese. Although we cannot elaborate, we note that Cangjie codes are not good for comparing characters that have few strokes, e.g., c_1 to c_4 in Table 1. In these cases, the coding method for Wubihua input method (Wubihua, 2010) should be applied.

Acknowledgement

This research was supported in part by the research contract NSC-97-2221-E-004-007-MY2 from the National Science Council of Taiwan. We thank the anonymous reviewers for constructive comments. Although we are not able to respond to all the comments

[6]The sum of 516 and 300 is larger than 598 because some of the characters are similar both phonologically and visually.

in this paper, we have done so in an extended version of this paper.

References

Cangjie. 2010. Last visited on 22 April 2010: en.wikipedia.org/wiki/Cangjie_input_method.

CDL. 2010. Chinese document laboratory, Academia Sinica. Last visited on 22 April, 2010; cdp.sinica.edu.tw/cdphanzi/. (in Chinese)

Chen, Matthew. Y. 2000. *Tone Sandhi: Patterns across Chinese Dialects*, (Cambridge Studies in Linguistics 92). Cambridge University Press.

Chu, Bong-Foo. 2010. *Handbook of the Fifth Generation of the Cangjie Input Method*. last visited on 22 April 2010: www.cbflabs.com/book/5cjbook/. (in Chinese)

Cormen, Thomas H., Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. 2009. *Introduction to Algorithms*, third edition. MIT Press.

Croft, W. Bruce, Donald Metzler, and Trevor Strohman, 2010. *Search Engines: Information Retrieval in Practice*, Pearson.

Dict. 2010. Last visited on 22 April 2010, www.cns11643.gov.tw/AIDB/welcome.do

Fan, Kuo-Chin, Chang-Keng Lin, and Kuo-Sen Chou. 1995. Confusion set recognition of on-line Chinese characters by artificial intelligence technique. *Pattern Recognition*, **28**(3):303−313.

HanDict. 2010. Last visit on 22 April 2010, www.zdic.net/appendix/f19.htm.

Jurafsky, Daniel and James H. Martin. 2009. *Speech and Language Processing*, second edition, Pearson.

Kuo, Wen-Jui, Tzu-Chen Yeh, Jun-Ren Lee, Li-Fen Chen, Po-Lei Lee, Shyan-Shiou Chen, Low-Tone Ho, Daisy L. Hung, Ovid J.-L. Tzeng, and Jen-Chuen Hsieh. 2004. Orthographic and phonological processing of Chinese characters: An fMRI study. *NeuroImage*, **21**(4):1721−1731.

Lee, Chia-Ying, Jie-Li Tsai, Hsu-Wen Huang, Daisy L. Hung, Ovid J.-L. Tzeng. 2006. The temporal signatures of semantic and phonological activations for Chinese sublexical processing: An even-related potential study. *Brain Research*, **1121**(1):150-159.

Lee, Hsiang. 2010a. *Cangjie Input Methods in 30 Days 2*. Foruto. Last visited on 22 April 2010: input.foruto.com/cccls/cjzd.html.

Lee, Mu. 2010b. A quantitative study of the formation of Chinese characters. Last visited on 22 April 2010: chinese.exponode.com/0_1.htm. (in Chinese)

Liu, Chao-Lin, and Jen-Hsiang Lin. 2008. Using structural information for identifying similar Chinese characters. *Proc. of the 46th Annual Meeting of the Association for Computational Linguistics*, short papers, 93−96.

Liu, Chao-Lin, Kan-Wen Tien, Yi-Hsuan Chuang, Chih-Bin Huang, and Juei-Yu Weng. 2009a. Two applications of lexical information to computer-assisted item authoring for elementary Chinese. *Proc. of the 22nd Int'l Conf. on Industrial Engineering & Other Applications of Applied Intelligent Systems*, 470−480.

Liu, Chao-Lin, Kan-Wen Tien, Min-Hua Lai, Yi-Hsuan Chuang, and Shih-Hung Wu. 2009b. Capturing errors in written Chinese words. *Proc. of the 47th Annual Meeting of the Association for Computational Linguistics*, short papers, 25−28.

Liu, Chao-Lin, Kan-Wen Tien, Min-Hua Lai, Yi-Hsuan Chuang, and Shih-Hung Wu. 2009c. Phonological and logographic influences on errors in written Chinese words. *Proc. of the 7th Workshop on Asian Language Resources*, the 47th Annual Meeting of the ACL, 84−91.

Liu, Cheng-Lin, Stefan Jaeger, and Masaki Nakagawa. 2004. Online recognition of Chinese characters: The state-of-the-art. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, **26**(2):198−213.

Wubihua. 2010. Last visited on 22 April 2010: en.wikipedia.org/wiki/Wubihua_method.

Yeh, Su-Ling, and Jing-Ling Li. 2002. Role of structure and component in judgments of visual similarity of Chinese Characters. *Journal of Experimental Psychology: Human Perception and Performance*, **28**(4):933−947.

Head-modifier Relation based Non-lexical Reordering Model for Phrase-Based Translation

Shui Liu[1], Sheng Li[1], Tiejun Zhao[1], Min Zhang[2], Pengyuan Liu[3]
[1]School of Computer Science and Technology, Habin Institute of Technology
{liushui,lisheng,tjzhao}@mtlab.hit.edu.cn
[2]**Institute for Infocomm Research**
mzhang@i2r.a-star.edu.sg
[3]**Institute of Computational Linguistics, Peking University**
liupengyuan@pku.edu.cn

Abstract

Phrase-based statistical MT (SMT) is a milestone in MT. However, the translation model in the phrase based SMT is structure free which greatly limits its reordering capacity. To address this issue, we propose a non-lexical head-modifier based reordering model on word level by utilizing constituent based parse tree in source side. Our experimental results on the NIST Chinese-English benchmarking data show that, with a very small size model, our method significantly outperforms the baseline by 1.48% bleu score.

1 Introduction

Syntax has been successfully applied to SMT to improve translation performance. Research in applying syntax information to SMT has been carried out in two aspects. On the one hand, the syntax knowledge is employed by directly integrating the syntactic structure into the translation rules i.e. syntactic translation rules. On this perspective, the word order of the target translation is modeled by the syntax structure explicitly. Chiang (2005), Wu (1997) and Xiong (2006) learn the syntax rules using the formal grammars. While more research is conducted to learn syntax rules with the help of linguistic analysis (Yamada and Knight, 2001; Graehl and Knight, 2004). However, there are some challenges to these models. Firstly, the linguistic analysis is far from perfect. Most of these methods require an off-the-shelf parser to generate syntactic structure, which makes the translation results sensitive to the parsing errors to some extent. To tackle this problem, n-best parse trees and parsing forest (Mi and Huang, 2008; Zhang, 2009) are proposed to relieve the error propagation brought by linguistic analysis. Secondly, some phrases which violate the boundary of linguistic analysis are also useful in these models (DeNeefe et al., 2007; Cowan et al. 2006). Thus, a tradeoff needs to be found between linguistic sense and formal sense.

On the other hand, instead of using syntactic translation rules, some previous work attempts to learn the syntax knowledge separately and then integrated those knowledge to the original constraint. Marton and Resnik (2008) utilize the language linguistic analysis that is derived from parse tree to constrain the translation in a soft way. By doing so, this approach addresses the challenges brought by linguistic analysis through the log-linear model in a soft way.

Starting from the state-of-the-art phrase based model Moses (Koehn e.t. al, 2007), we propose a head-modifier relation based reordering model and use the proposed model as a soft syntax constraint in the phrase-based translation framework. Compared with most of previous soft constraint models, we study the way to utilize the constituent based parse tree structure by mapping the parse tree to sets of head-modifier for phrase reordering. In this way, we build a word level reordering model instead of phrasal/constituent level model. In our model, with the help of the alignment and the head-modifier dependency based relationship in the source side, the reordering type of each target word with alignment in source side is identified as one of pre-defined reordering types. With these reordering types, the reordering of phrase in translation is estimated on word level.

Coling 2008: Poster Volume, pages 748–756,
Beijing, August 2010

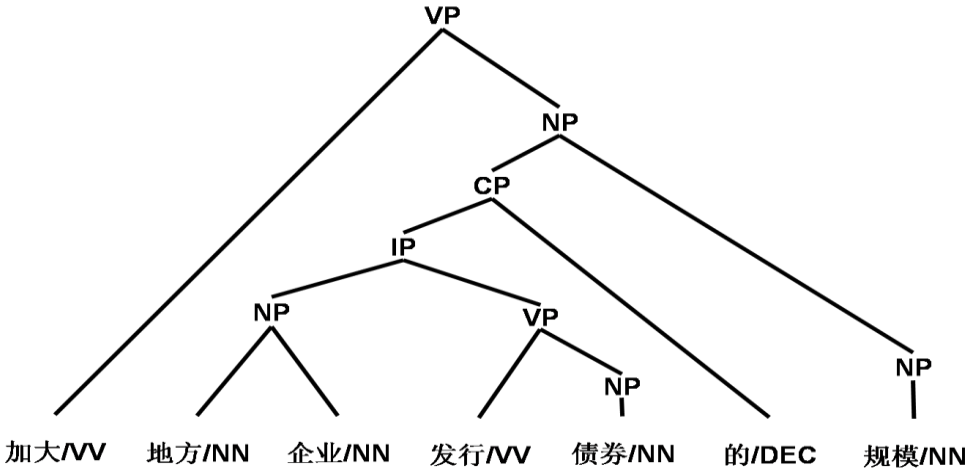


Fig 1. An Constituent based Parse Tree

2 Baseline

Moses, a state-of-the-art phrase based SMT system is used as our baseline system. In Moses, given the source language *f* and target language *e*, the decoder is to find:

$$e_{best} = \text{argmax}_e\, p(e \mid f)\, pLM(e)\, \omega^{length(e)} \quad (1)$$

where p(e|f) can be computed using phrase translation model, distortion model and lexical reordering model. pLM(e) can be computed using the language model. $\omega^{length(e)}$ is word penalty model.

Among the above models, there are three reordering-related components: language model, lexical reordering model and distortion model. The language model can reorder the local target words within a fixed window in an implied way. The lexical reordering model and distortion reordering model tackle the reordering problem between adjacent phrase on lexical level and alignment level. Besides these reordering model, the decoder induces distortion pruning constraints to encourage the decoder translate the leftmost uncovered word in the source side firstly and to limit the reordering within a certain range.

3 Model

In this paper, we utilize the constituent parse tree of source language to enhance the reordering capacity of the translation model. Instead of directly employing the parse tree fragments (Bod, 1992; Johnson, 1998) in reordering rules (Huang and Knight, 2006; Liu 2006; Zhang and Jiang 2008), we make a mapping from trees to sets of head-modifier dependency relations (Collins 1996) which can be obtained from the constituent based parse tree with the help of head rules (Bikel, 2004).

3.1 Head-modifier Relation

According to Klein and Manning (2003) and Collins (1999), there are two shortcomings in n-ary Treebank grammar. Firstly, the grammar is too coarse for parsing. The rules in different context always have different distributions. Secondly, the rules learned from training corpus cannot cover the rules in testing set.

Currently, the state-of-the-art parsing algorithms (Klein and Manning, 2003; Collins 1999) decompose the n-ary Treebank grammar into sets of head-modifier relationships. The parsing rules in these algorithms are constructed in the form of finer-grained binary head-modifier dependency relationships. Fig.2 presents an example of head-modifier based dependency tree mapped from the constituent parse tree in Fig.1.

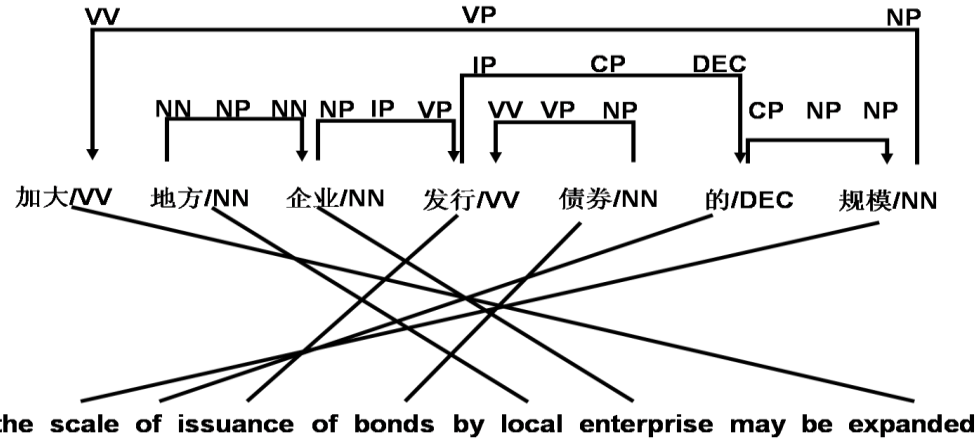


Fig. 2. Head-modifier Relationships with Aligned Translation

Moreover, there are several reasons for which we adopt the head-modifier structured tree as the main frame of our reordering model. Firstly, the dependency relationships can reflect some underlying binary long distance dependency relations in the source side. Thus, binary dependency structure will suffer less from the long distance reordering constraint. Secondly, in head-modifier relation, we not only can utilize the context of dependency relation in reordering model, but also can utilize some well-known and proved helpful context (Johnson, 1998) of constituent base parse tree in reordering model. Finally, head-modifier relationship is mature and widely adopted method in full parsing.

3.2 Head-modifier Relation Based Reordering Model

Before elaborating the model, we define some notions further easy understanding. $S=<f_1, f_2 \ldots f_n>$ is the source sentence; $T=<e_1,e_2,\ldots,e_m>$ is the target sentence; $A_S=\{a_s(i) \mid 1\leq a_s(i) \leq n\}$ where $a_s(i)$ represents that the ith word in source sentence aligned to the $a_s(i)th$ word in target sentence; $A_T=\{a_T(i) \mid 1\leq a_T(i) \leq n\}$ where $a_T(i)$ represents that the ith word in target sentence aligned to the $a_T(i)th$ word in source sentence; $D=\{(d(i), r(i)) \mid 0\leq d(i) \leq n\}$ is the head-modifier relation set of the words in S where $d(i)$ represents that the ith word in source sentence is the modifier of $d(i)th$ word in source sentence under relationship $r(i)$; $O=<o_1, o_2,\ldots, o_m>$ is the sequence of the reordering type of every word in target language. The reordering model probability is P(O| S, T, D, A).

Relationship: in this paper, we not only use the label of the constituent label as Collins (1996), but also use some well-known context in parsing to define the head-modifier relationship *r(.)*, including the POS of the modifier *m*, the POS of the head *h*, the dependency direction *d*, the parent label of the dependency label *l*, the grandfather label of the dependency relation *p*, the POS of adjacent siblings of the modifier *s*. Thus, the head-modifier relationship can be represented as a 6-tuple <*m*, *h*, *d*, *l*, *p*, *s*>.

r(.)	relationship
r(1)	<VV, - , -, -, -, - >
r(2)	<NN, NN, right, NP, IP, - >
r(3)	<NN,VV, right, IP, CP, - >
r(4)	<VV, DEC, right, CP, NP, - >
r(5)	<NN,VV, left, VP, CP, - >
r(6)	<DEC, NP, right, NP, VP, - >
r(7)	<NN, VV, left, VP, TOP, - >

Table 1. Relations Extracted from Fig 2.

In Table 1, there are 7 relationships extracted from the source head-modifier based dependency tree as shown in Fig.2. Please notice that, in this paper, each source word has a corresponding relation.

Reordering type: there are 4 reordering types for target words with linked word in the source side in our model: $R=\{rm_1, rm_2, rm_3, rm_4\}$. The reordering type of target word $a_s(i)$ is defined as follows:

- **rm_1**: if the position number of the ith word's head is less than i (d(i) < i) in source language, while the position number of the word aligned to i is less than

$a_s(d(i))$ ($a_s(i) < a_s(d(i))$) in target language;

- **rm_2**: if the position number of the ith word's head is less than i (d(i) < i) in source language, while the position number of the word aligned to i is larger than $a_s(d(i))$ ($a_s(i) > a_s(d(i))$) in target language.
- **rm_3**: if the position number of the ith word's head is larger than i (d(i) > i) in source language, while the position number of the word aligned to i is larger than $a_s(d(i))$ ($a_s(i) > a_s(d(i))$) in target language.
- **rm_4**: if the position number of the ith word's head is larger than i (d(i) > i) in source language, while the position number of the word aligned to i is less than $a_s(d(i))$ ($a_s(i) < a_s(d(i))$) in target language.

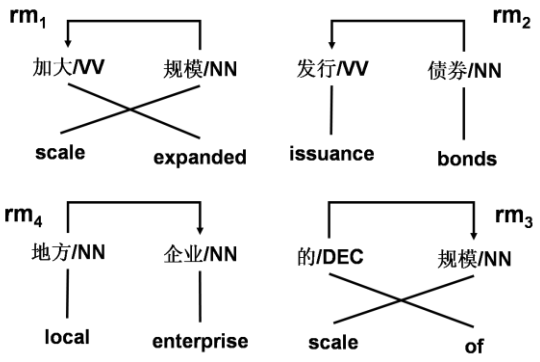


Fig. 3. An example of the reordering types in Fig. 2.

Fig. 3 shows examples of all the reordering types. In Fig. 3, the reordering type is labeled at the target word aligned to the modifier: for example, the reordering type of rm_1 belongs to the target word "*scale*". Please note that, in general, these four types of reordering can be divided into 2 categories: the target words order of rm_2 and rm_4 is identical with source word order, while rm_1 and rm_3 is the swapped order of source. In practice, there are some special cases that can't be classified into any of the defined reordering types: the head and modifier in source link to the same word in target. In such cases, rather than define new reordering types, we classify these special cases into these four defined reordering types: if the head is right to the modifier in source, we classify the reordering type into rm_2; otherwise, we classify the reordering type into rm_4.

Probability estimation: we adopt maximum likelihood (ML) based estimation in this paper. In ML estimation, in order to avoid the data sparse problem brought by lexicalization, we discard the lexical information in source and target language:

$$P(O \mid S, T, D, A) = \prod_{i=1}^{m} P(o_i \mid -,-, r(a_T(i))) \quad (2)$$

where $o_i \in \{rm1, rm2, rm3, rm4\}$ is the reordering type of ith word in target language.

To get a non-zero probability, additive smooth ing(Chen and Goodman, 1998) is used:

$$P(o_i \mid -,-, F(r(a_t(i)))) = \frac{F(o_i, r(a_T(i))) + \alpha}{\sum_{o_i \in R} F(r(a_t(i))) + |O| \times \alpha} = \frac{F(o_i, m_{a_T(i)}, h_{a_T(i)}, d_{a_T(i)}, l_{a_T(i)}, p_{a_T(i)}, s_{a_T(i)}) + \alpha}{\sum_{o_i \in R} F(m_{a_T(i)}, h_{a_T(i)}, d_{a_T(i)}, l_{a_T(i)}, p_{a_T(i)}, s_{a_T(i)}) + |O| \times \alpha} \quad (3)$$

where F(.) is the frequency of the statistic event in training corpus. For a given set of dependency relationships mapping from constituent tree, the reordering type of ith word is confined to two types: it is whether one of rm1 and rm2 or rm3 and rm4. Therefore, |O|=2 instead of |O|=4 in (2). The parameter α is an additive factor to prevent zero probability. It is computed as:

$$\alpha = \frac{1}{C \times \sum_{o_i \in R} F(m_{a_T(i)}, h_{a_T(i)}, d_{a_T(i)}, l_{a_T(i)}, p_{a_T(i)}, s_{a_T(i)})} \quad (4)$$

where c is a constant parameter(c=5 in this paper).

In above, the additive parameter α is an adaptive parameter decreasing with the size of the statistic space. By doing this, the data sparse problem can be relieved.

4 Apply the Model to Decoder

Our decoding algorithm is exactly the same as (Kohn, 2004). In the translation procedure, the decoder keeps on extending new phrases without overlapping, until all source words are translated. In the procedure, the order of the target

words in decoding procedure is fixed. That is, once a hypothesis is generated, the order of target words cannot be changed in the future. Taking advantage of this feature, instead of computing a totally new reordering score for a newly generated hypothesis, we merely calculate the reordering score of newly extended part of the hypothesis in decoding. Thus, in decoding, to compute the reordering score, the reordering types of each target word in the newly extended phrase need to be identified.

The method to identify the reordering types in decoding is proposed in Fig.4. According to the definition of reordering, the reordering type of the target word is identified by the direction of head-modifier dependency on the source side, the alignment between the source side and target side, and the relative translated order of word pair under the head-modifier relationship. The direction of dependency and the alignment can be obtained in input sentence and phrase table. While the relative translation order needs to record during decoding. A word index is employed to record the order. The index is constructed in the form of true/false array: the index of the source word is set with true when the word has been translated. With the help of this index, reordering type of every word in the phrase can be identified.

1: **Input**: alignment array A_T; the *Start* is the start position of the phrase in the source side; head-modifier relation *d(.)*; source word index *C*, where *C[i]=true* indicates that the *ith* word in source has been translated.
2: **Output:** reordering type array *O* which reserves the reordering types of each word in the target phrase
3: **for** i = 1, $|A_T|$ **do**
4: P ← a_T(i) + Start
5: **if** (d (P)<P) **then**
6: **if** C [d(p)] = false **then**
7: O[i] ← rm_1
8: **else**
9: O[i] ← rm_2
10: **end if**
11: **else**
12: **if** C[d(p)] = true **then**
13: O[i] ← rm_3
14: **else**
15: O[i] ← rm_4
16: **end if**
17: **end if**
18: C[p] ←true //update word index
19: **end for**

Fig. 4. Identify the Reordering Types of Newly Extended Phrase

After all the reordering types in the newly extended phrase are identified, the reordering scores of the phrase can be computed by using equation (3).

5 Preprocess the Alignment

In Fig. 4, the word index is to identify the reordering type of the target translated words. Actually, in order to use the word index without ambiguity, the alignment in the proposed algorithm needs to satisfy some constraints.

Firstly, every word in the source must have alignment word in the target side. Because, in the decoding procedure, if the head word is not covered by the word index, the algorithm cannot distinguish between the head word will not be translated in the future and the head word is not translated yet. Furthermore, in decoding, as shown in Fig.4, the index of source would be set with true only when there is word in target linked to it. Thus, the index of the source word without alignment in target is never set with true.

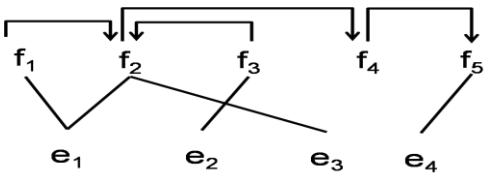


Fig. 5. A complicated Example of Alignment in Head-modifier based Reordering Model

Secondly, if the head word has more than one alignment words in target, different alignment possibly result in different reordering type. For example, in Fig. 5, the reordering type of e_2 is different when f_2 select to link word e_1 and e3 in the source side.

To solve this problem, we modify the alignment to satisfy following conditions: a) each word in source just has only one alignment word in target, and b) each word in target has at most one word aligned in source as its anchor word which decides the reordering type of the target word.

To make the alignment satisfy above constraints, we modify the alignment in corpus. In

order to explain the alignment preprocessing, the following notions are defined: if there is a link between the source word f_j and target word e_i, let $l(e_i, f_j) = 1$, otherwise $l(e_i, f_j) = 0$; the source word $f_j \in F_{1\text{-to-N}}$, iff $\sum_i l(e_i, f_j) > 1$, such as the source word f_2 in Fig. 5; the source word $f_j \in F_{NULL}$, iff $\sum_i l(e_i, f_j) = 0$, such as the source word f_4 in Fig. 5; the target word $e_i \in E_{1\text{-to-N}}$, iff $\sum_j l(e_i, f_j) > 1$, such as the target word e_1 in Fig. 5.

In preprocessing, there are 3 types of operation, including *DiscardLink(f_j)* , *BorrowLink(f_j)* and *FindAnchor*(e_i) :

DiscardLink(f_j) : if the word f_j in source with more than one words aligned in target, i.e. $f_j \in F_{1\text{-to-N}}$; We set the target word e_n with $l(e_n, f_j) = 1$, where $e_n = \text{argmax}_i\ p(e_i \mid f_j)$ and $p(e_i \mid f_j)$ is estimated by (Koehn e.t. al, 2003), while set rest of words linked to f_j with $l(e_n, f_j) = 0$.

BorrowLink(f_j): if the word f_j in source without a alignment word in target, i.e. $f_j \in F_{NULL}$; let $l(e_i, f_j)=1$ where e_i aligned to the word f_j , which is the nearest word to f_j in the source side; when there are two words nearest to f_j with alignment words in the target side at the same time, we select the alignment of the left word firstly .

FindAnchor(e_i): for the word e_i in target with more than one words aligned in source , i.e. $e_i \in E_{1\text{-to-N}}$; we select the word f_m aligned to e_i as its anchor word to decide the reordering type of e_i , where $f_m = \text{argmax}_j\ p(e_i \mid f_j)$ and $p(f_j \mid e_i)$ is estimated by (Koehn et al, 2003); For the rest of words aligned to e_i , we would set their word indexes with true in the update procedure of decoding in the 18th line of Fig.4.

With these operations, the required alignment can be obtained by preprocessing the origin alignment as shown in Fig. 6.

1: **Input**: set of alignment A between target language *e* and source language *f*
2: **Output:** the 1-to-1 alignment required by the model
3: **foreach** $f_i \in F_{1\text{-to-N}}$ **do**
4: DiscardLink(f_i)
5: **end for**
6: **foreach** $f_i \in F_{NULL}$ **do**
7: BorrowLink(f_i)
8: **end for**
9: **foreach** $e_i \in E_{1\text{-to-N}}$ **do**
10: FindAnchor(e_i)
11: **endfor**

Fig. 6. Alignment Pre-Processing algorithm

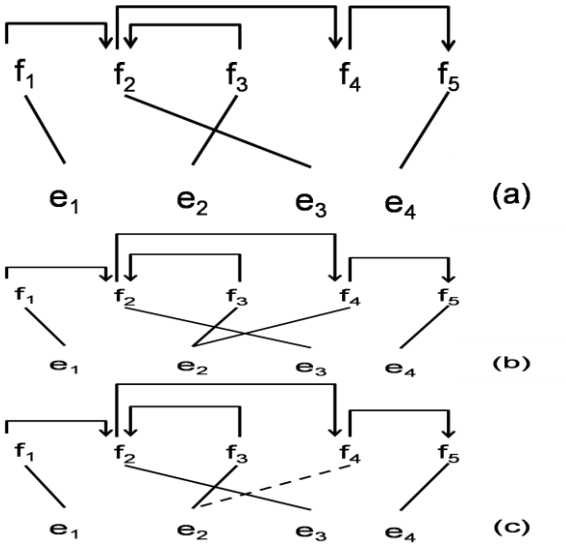


Fig. 7. An Example of Alignment Preprocessing.

An example of the preprocess the alignment in Fig. 5 is shown in Fig. 7 : firstly, *DiscardLink(f_2)* operation discards the link between f_2 and e_1 in (a); then the link between f_4 and e_3 is established by operation *BorrowLink(f_4)* in (b); at last, *FindAnchor(e_3)* select f_2 as the anchor word of e_3 in source in (c). After the preprocessing, the reordering type of e_3 can be identified. Furthermore, in decoding, when the decoder scans over e_2, the word index sets the word index of f_3 and f_4 with true. In this way, the never-true word indexes in decoding are avoided.

6 Training the Reordering Model

Before training, we get the required alignment by alignment preprocessing as indicated above. Then we train the reordering model with this alignment: from the first word to the last word in the target side, the reordering type of each word is identified. In this procedure, we skip the words without alignment in source. Finally, all the statistic events required in equation (3) are added to the model.

In our model, there are 20,338 kinds of relations with reordering probabilities which are much smaller than most phrase level reordering models on the training corpus FBIS.

Table 1 is the distribution of different reordering types in training model.

Type of Reordering	Percentage %
rm_1	3.69
rm_2	27.61
rm_3	20.94
rm_4	47.75

Table 1: Percentage of different reordering types in model

From Table 1, we can conclude that the reordering type rm_2 and rm_4 are preferable in reordering which take over nearly 3/4 of total number of reordering type and are identical with word order of the source. The statistic data indicate that most of the words order doesn't change in our head-modifier reordering view. This maybe can explain why the models (Wu, 1997; Xiong, 2006; Koehn, et., 2003) with limited capacity of reordering can reach certain performance.

7 Experiment and Discussion

7.1 Experiment Settings

We perform Chinese-to-English translation task on NIST MT-05 test set, and use NIST MT-02 as our tuning set. FBIS corpus is selected as our training corpus, which contains 7.06M Chinese words and 9.15M English words. We use GIZA++(Och and Ney, 2000) to make the corpus aligned. A 4-gram language model is trained using Xinhua portion of the English Gigaword corpus (181M words). All models are tuned on BLEU, and evaluated on both BLEU and NIST score.

To map from the constituent trees to sets of head-modifier relationships, firstly we use the Stanford parser (Klein, 2003) to parse the source of corpus FBIS, then we use the head-finding rules in (Bikel, 2004) to get the head-modifier dependency sets.

In our system, there are 7 groups of features. They are:

1. Language model score (1 feature)
2. word penalty score (1 feature)
3. phrase model scores (5 features)
4. distortion score (1 feature)
5. lexical RM scores (6 features)
6. Number of each reordering type (4 features)
7. Scores of each reordering type (4 features, computed by equation (3))

In these feature groups, the top 5 groups of features are the baseline model, the left two group scores are related with our model.

In decoding, we drop all the OOV words and use default setting in Moses: set the distortion limitation with 6, beam-width with 1/100000, stack size with 200 and max number of phrases for each span with 50.

7.2 Results and Discussion

We take the replicated Moses system as our baseline. Table 2 shows the results of our model. In the table, Baseline model is the model including feature group 1, 2, 3 and 4. $Baseline_{rm}$ model is the Baseline model with feature group 5. H-M model is the Baseline model with feature group 6 and 7. $H\text{-}M_{rm}$ model is the $Baseline_{rm}$ model with feature group 6 and 7.

Model	BLEU%	NIST
Baseline	27.06	7.7898
$Baseline_{rm}$	27.58	7.8477
H-M	28.47	8.1491
$H\text{-}M_{rm}$	29.06	8.0875

Table 2: Performance of the Systems on NIST-05(bleu4 case-insensitive).

From table 2, we can conclude that our reordering model is very effective. After adding feature group 6 and 7, the performance is improved by 1.41% and 1.48% in bleu score separately. Our reordering model is more effective than the lexical reordering model in Moses: 1.41% in bleu score is improved by adding our reordering model to Baseline model, while 0.48 is improved by adding the lexical reordering to Baseline model.

threshold	KOR	BLEU	NIST
$\geqslant 1$	20,338	29.06	8.0875
$\geqslant 2$	13,447	28.83	8.3658
$\geqslant 3$	10,885	28.64	8.0350
$\geqslant 4$	9,518	28.94	8.1002
$\geqslant 5$	8,577	29.18	8.1213

Table 3: Performance on NIST-05 with Different Relation Frequency Threshold (bleu4 case-insensitive).

Although our model is lexical free, the data sparse problem affects the performance of the model. In the reordering model, nearly half numbers of the relations in our model occur less than three times. To investigate this, we statistic

the frequency of the relationships in our model, and expertise our H-M_{full} model with different frequency threshold.

In Table 3, when the frequency of relation is not less than the *threshold*, the relation is added into the reordering model; KOR is the number of relation type in the reordering model.

Table 3 shows that, in our model, many relations occur only once. However, these low-frequency relations can improve the performance of the model according to the experimental results. Although low frequency statistic events always do harm to the parameter estimation in ML, the model can estimate more events in the test corpus with the help of low frequency event. These two factors affect the experiment results on opposite directions: we consider that is the reason the result don't increase or decrease with the increasing of frequency threshold in the model. According to the results, the model without frequency threshold achieves the highest bleu score. Then, the performance drops quickly, when the frequency threshold is set with 2. It is because there are many events can't be estimated by the smaller model. Although, in the model without frequency threshold, there are some probabilities overestimated by these events which occur only once, the size of the model affects the performance to a larger extent. When the frequency threshold increases above 3, the size of model reduces slowly which makes the overestimating problem become the important factor affecting performance. From these results, we can see the potential ability of our model: if our model suffer less from data spars problem, the performance should be further improved, which is to be verified in the future.

8 Related Work and Motivation

There are several researches on adding linguistic analysis to MT in a "soft constraint" way. Most of them are based on constituents in parse tree. Chiang(2005), Marton and Resnik(2008) explored the constituent match/violation in hiero; Xiong (2009 a) added constituent parse tree based linguistic analysis into BTG model; Xiong (2009 b) added source dependency structure to BTG; Zhang(2009) added tree-kernel to BTG model. All these studies show promising results. Making soft constrain is an easy and efficient way in adding linguistic analysis into formal sense SMT model.

In modeling the reordering, most of previous studies are on phrase level. In Moses, the lexical reordering is modeled on adjacent phrases. In (Wu, 1996; Xiong, 2006), the reordering is also modeled on adjacent translated phrases. In hiero, the reordering is modeled on the segments of the unmotivated translation rules. The tree-to-string models (Yamada et al. 2001; Liu et al.2006) are model on phrases with syntax representations. All these studies show excellent performance, while there are few studies on word level model in recent years. It is because, we consider, the alignment in word level model is complex which limits the reordering capacity of word level models.

However, our work exploits a new direction in reordering that, by utilizing the decomposed dependency relations mapped from parse tree as a soft constraint, we proposed a novel head-modifier relation based word level reordering model. The word level reordering model is based on a phrase based SMT framework. Thus, the task to find the proper position of translated words converts to score the reordering of the translated words, which relax the tension between complex alignment and word level reordering in MT.

9 Conclusion and Future Work

Experimental results show our head-modifier relationship base model is effective to the baseline (enhance by 1.48% bleu score), even with limited size of model and simple parameter estimation. In the future, we will try more complicated smooth methods or use maximum entropy based reordering model. We will study the performance with larger distortion constraint, such as the performances of the distortion constraint over 15, or even the performance without distortion model.

10 Acknowledgement

The work of this paper is funded by National Natural Science Foundation of China (grant no. 60736014), National High Technology Research and Development Program of China (863 Program) (grant no. 2006AA010108), and Microsoft Research Asia IFP (grant no. FY09-RES-THEME-158).

References

Dekai Wu. 1997. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational Lingustics*,23(3):377-403.

David Chiang. 2005. A hierarchical phrase-based model for SMT. ACL-05.263-270.

David Chiang. 2007. Hierarchical phrase-based translation. *Computational Linguistics*, 33(2):201-228.

Kenji Yamada and K. Knight. 2001. A syntax-based statistical translation model. *ACL-01*.523-530.

Yuval Marton and Philip Resnik. 2008. Soft syntactic Constraints for Hierarchical Phrased-based Translation. *ACL-08*. 1003-1011.

Libin shen, Jinxi Xu and Ralph Weischedel. 2008. A New String-to-Dependency Machine Translation Algorithm with a Target Dependency Language Model. *ACL-08*. 577-585.

J. Graehl and K. Knight.2004.Training Tree transducers. In proceedings of *the 2004 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*.

Dekai Wu. 1996. A Polynomial-Time Algorithm for Statistical Machine Translation. In proceedings of *ACL-1996*

Deyi Xiong, Qun Liu, and Shouxun Lin. 2006. Maximum Entropy Based Phrase Reordering Model for Statistical Machine Translation. In proceedings of *COLING-ACL 2006*

Deyi Xiong, Min Zhang, Aiti AW and Haizhou Li. 2009a. A Syntax-Driven Bracket Model for Phrase-Based Translation. *ACL-09*.315-323.

Deyi Xiong, Min Zhang, Aiti AW and Haizhou Li. 2009b. A Source Dependency Model for Statistic Machine translation. *MT-Summit 2009*.

Och, F.J. and Ney, H. 2000. Improved statistical alignment models. In Proceedings of *ACL 38*.

Philipp Koehn, et al. Moses: Open Source Toolkit for Statistical Machine Translation, ACL 2007.

Philipp Koehn, Franz Joseph Och, and Daniel Marcu.2003. Statistical Phrase-based Translation. In Proceedings of *HLT-NAACL*.

Philipp Koehn. 2004. A Beam Search Decoder for Phrase-Based Translation model. In: Proceeding of *AMTA-2004*,Washington

Rens Bod. 1992. Data oriented Parsing(DOP). In Proceedings of *COLING-92*.

Mark Johnson. 1998. PCFG models of linguistic tree representations. Computational Linguistics, 24:613-632.

Liang Huang, Kevin Knight, and Aravind Joshi. Statistical Syntax-Directed Translation with Extended Domain of Locality. 2006. In Proceedings of *the 7th AMTA*.

Yang Liu, Qun Liu, and Shouxun Lin. Tree-to-String Alignment Template for Statistical Machine Translation. 2006.In Proceedings of t*he ACL 2006*.

Min Zhang, Hongfei Jiang, Ai Ti Aw, Haizhou Li, Chew Lim Tan and Sheng Li. 2008. A Tree Sequence Alignment-based Tree-to-Tree Translation Model. *ACL-HLT-08*. 559-567.

Dan Klein, Christopher D. Manning. Accurate Unlexicalized Parsing. 2003. In Proceedings of *ACL-03*. 423-430.

M. Collins. 1996. A new statistical parser based on bigram lexical dependencies. In Proceedings of *ACL-96*. 184-191.

M. Collins. 1999. Head-Driven Statistical Models for Natural Language Parsing. Ph.D. thesis, Univ. of Pennsylvania.

Andreas Zollmann. 2005. A Consistent and Efficient Estimator for the Data-Oriented Parsing Model. Journal of Automata, Languages and Combinatorics. 2005(10):367-388

Mark Johnson. 2002. The DOP estimation method is biased and inconsistent. *Computational Linguistics* 28, 71-76.

Daniel M. Bikel. 2004. On the Parameter Space of Generative Lexicalized Statistical Parsing Models. Ph.D. thesis. Univ. of Pennsylvania.

S. F. Chen, J. Goodman. An Empirical Study of Smoothing Techniques for Language Modeling. In Proceedings of *the 34th annual meeting on Association for Computational Linguistics*, 1996.310-318.

Haitao Mi and Liang Huang. 2008. Forest-based translation Rule Extraction. *ENMLP*-08. 2006-214.

Hui Zhang, Min Zhang , Haizhou Li, Aiti Aw and Chew Lim Tan. Forest-based Tree Sequence to String Translation Model. *ACL*-09: 172-180

S DeNeefe, K. Knight, W. Wang, and D. Marcu. 2007. What can syntax-based MT learn from phrase-based MT ? In Proc. *EMNLP-CoNULL*.

Brooke Cowan, Ivona Kucerova, and Michael Collins.2006. A discriminative model for tree-to-tree translation. In Proc. *EMNLP*.

Dependency-Driven Feature-based Learning for Extracting Protein-Protein Interactions from Biomedical Text

Bing Liu Longhua Qian* Hongling Wang Guodong Zhou
Jiangsu Provincial Key Lab for Computer Information Processing Technology
School of Computer Science and Technology
Soochow University
Email: liubingnlp@gmail.com
{qianlonghua,redleaf,gdzhou}@suda.edu.cn

Abstract

Recent kernel-based PPI extraction systems achieve promising performance because of their capability to capture structural syntactic information, but at the expense of computational complexity. This paper incorporates dependency information as well as other lexical and syntactic knowledge in a feature-based framework. Our motivation is that, considering the large amount of biomedical literature being archived daily, feature-based methods with comparable performance are more suitable for practical applications. Additionally, we explore the difference of lexical characteristics between biomedical and newswire domains. Experimental evaluation on the AIMed corpus shows that our system achieves comparable performance of 54.7 in F1-Score with other state-of-the-art PPI extraction systems, yet the best performance among all the feature-based ones.

1 Introduction

In recent years, automatically extracting biomedical information has been the subject of significant research efforts due to the rapid growth in biomedical development and discovery. A wide concern is how to characterize protein interaction partners since it is crucial to understand not only the functional role of individual proteins but also the organization of the entire biological process. However, manual collection of relevant Protein-Protein Interaction (PPI) information from thousands of research papers published every day is so time-consuming that automatic extraction approaches with the help of Natural Language Processing (NLP) techniques become necessary.

Various machine learning approaches for relation extraction have been applied to the biomedical domain, which can be classified into two categories: feature-based methods (Mitsumori et al., 2006; Giuliano et al., 2006; Sætre et al., 2007) and kernel-based methods (Bunescu et al., 2005; Erkan et al., 2007; Airola et al., 2008; Kim et al., 2010).

Provided a large-scale manually annotated corpus, the task of PPI extraction can be formulated as a classification problem. Typically, for featured-based learning each protein pair is represented as a vector whose features are extracted from the sentence involving two protein names. Early studies identify the existence of protein interactions by using "bag-of-words" features (usually uni-gram or bi-gram) around the protein names as well as various kinds of shallow linguistic information, such as POS tag, lemma and orthographical features. However, these systems do not achieve promising results since they disregard any syntactic or semantic information altogether, which are very useful for the task of relation extraction in the newswire domain (Zhao and Grishman, 2005; Zhou et al., 2005). Furthermore, feature-based methods fail to effectively capture the structural information, which is essential to

* Corresponding author

Coling 2008: Poster Volume, pages 757–765,
Beijing, August 2010

identify the relationship between two proteins in a syntactic representation.

With the wide application of kernel-based methods to many NLP tasks, various kernels such as subsequence kernels (Bunescu and Mooney, 2005) and tree kernels (Li et al., 2008), are also applied to PPI detection.. Particularly, dependency-based kernels such as edit distance kernels (Erkan et al., 2007) and graph kernels (Airola et al., 2008; Kim et al., 2010) show some promising results for PPI extraction. This suggests that dependency information play a critical role in PPI extraction as well as in relation extraction from newswire stories (Culotta and Sorensen, 2004). In order to appreciate the advantages of both feature-based methods and kernel-based methods, composite kernels (Miyao et al., 2008; Miwa et al., 2009a; Miwa et al., 2009b) are further employed to combine structural syntactic information with flat word features and significantly improve the performance of PPI extraction. However, one critical challenge for kernel-based methods is their computation complexity, which prevents them from being widely deployed in real-world applications regarding the large amount of biomedical literature being archived everyday.

Considering the potential of dependency information for PPI extraction and the challenge of computation complexity of kernel-based methods, one may naturally ask the question: "Can the essential dependency information be maximally exploited in featured-based PPI extraction so as to enhance the performance without loss of efficiency?" "If the answer is Yes, then How?"

This paper addresses these problems, focusing on the application of dependency information to feature-based PPI extraction. Starting from a baseline system in which common lexical and syntactic features are incorporated using Support Vector Machines (SVM), we further augment the baseline with various features related to dependency information, including predicates in the dependency tree. Moreover, in order to reveal the linguistic difference between distinct domains we also compare the effects of various features on PPI extraction from biomedical texts with those on relation extraction from newswire narratives. Evaluation on the AIMed and other PPI corpora shows that our method achieves the best performance among all feature-based systems.

The rest of the paper is organized as follows. A feature-based PPI extraction baseline system is given in Section 2 while Section 3 describes our dependency-driven method. We report our experiments in Section 4, and compare our work with the related ones in Section 5. Section 6 concludes this paper and gives some future directions.

2 Feature-based PPI extraction: Baseline

For feature-based methods, PPI extraction task is re-cast as a classification problem by first transforming PPI instances into multi-dimensional vectors with various features, and then applying machine learning approaches to detect whether the potential relationship exists for a particular protein pair. In training, a feature-based classifier learning algorithm, such as SVM or MaxEnt, uses the annotated PPI instances to learn a classifier while, in testing, the learnt classifier is in turn applied to new instances to determine their PPI binary classes and thus candidate PPI instances are extracted.

As a baseline, various linguistic features, such as words, overlap, chunks, parse tree features as well as their combined ones are extracted from a sentence and formed as a vector into the feature-based learner.

1) Words

Four sets of word features are used in our system: 1) the words of both the proteins; 2) the words between the two proteins; 3) the words before M1 (the 1st protein); and 4) the words after M2 (the 2nd protein). Both the words before M1 and after M2 are classified into two bins: the first word next to the proteins and the second word next to the proteins. This means that we only consider the two words before M1 and after M2. Words features include:

- MW1: bag-of-words in M1
- MW2: bag-of-words in M2
- BWNULL: when no word in between
- BWO: other words in between except first and last words when at least three words in between
- BWM1FL: the only word before M1

- BWM1F: first word before M1
- BWM1L: second word before M1
- BWM1: first and second word before M1
- BWM2FL: the only word after M2
- BWM2F: first word after M2
- BWM2L: second word after M2
- BWM2: first and second word after M2

2) Overlap

The numbers of other protein names as well as the words that appear between two protein names are included in the overlap features. This category of features includes:

- #MB: number of other proteins in between
- #WB: number of words in between
- E-Flag: flag indicating whether the two proteins are embedded or not

3) Chunks

It is well known that chunking plays an important role in the task of relation extraction in the ACE program (Zhou et al., 2005). However, its significance in PPI extraction has not fully investigated. Here, the Stanford Parser[1] is first employed for full parsing, and then base phrase chunks are derived from full parse trees using the Perl script[2]. The chunking features usually concern about the head words of the phrases between the two proteins, which are further classified into three bins: the first phrase head in between, the last phrase head in between and other phrase heads in between. In addition, the path of phrasal labels connecting two proteins is also a common syntactic indicator of the polarity of the PPI instance, just as the path NP_VP_PP_NP in the sentence "*The ability of PROT1 to interact with the PROT2 was investigated.*" is likely to suggest the positive interaction between two proteins. These base phrase chunking features contain:

- CPHBNULL: when no phrase in between.
- CPHBFL: the only phrase head when only one phrase in between
- CPHBF: the first phrase head in between when at least two phrases in between.
- CPHBL: the last phrase head in between when at least two phrase heads in between.
- CPHBO: other phrase heads in between except first and last phrase heads when at least three phrases in between.
- CPP: path of phrase labels connecting the two entities in the chunking

Furthermore, we also generate a set of bi-gram features which combine the above chunk features except CPP with their corresponding chunk types.

4) Parse Tree

It is obvious that full pares trees encompass rich structural information of a sentence. Nevertheless, it is much harder to explore such information in featured-based methods than in kernel-based ones. Thus so far only the path connecting two protein names in the full-parse tree is considered as a parse tree feature.

- PTP: the path connecting two protein names in the full-parse tree.

Again, take the sentence "The ability of PROT1 to interact with the PROT2 was investigated." as an example, the parse path between PROT1 and PROT2 is NP_S_VP_PP_NP, which is slightly different from the CPP feature in the chunking feature set.

3 Dependency-Driven PPI Extraction

The potential of dependency information for PPI extraction lies in the fact that the dependency tree may well reveal non-local or long-range dependencies between the words within a sentence. In order to capture the necessary information inherent in the depedency tree for identifying their relationship, various kernels, such as edit distance kernel based on dependency path (Erkan et al., 2007), all-dependency-path graph kernel (Airola et al., 2008), and walk-weighted subsequence kernels (Kim et al., 2010) as well as other composite kernels (Miyao et al., 2008; Miwa et al., 2009a; Miwa et al., 2009b), have been proposed to address this problem. It's true that these methods achieve encouraging results, neverthless, they suffer from prohibitive computation burden.

1 http://nlp.stanford.edu/software/lex-parser.shtml
2 http://ilk.kub.nl/~sabine/chunklink/

Thus, our solution is to fold the structural dependency information back into flat features in a feature-based framework so as to speed up the learning process while retaining comparable performance. This is what we refer to as dependency-driven PPI extraction.

First, we construct dependency trees from grammatical relations generated by the Stanford Parser. Every grammatical relation has the form of *dependent-type* (*word1*, *word2*), Where *word1* is the head word, *word2* is dependent on *word1*, and *dependent-type* denotes the pre-defined type of dependency. Then, from these grammatical relations the following features called DependenecySet1 are taken into consideration as illustrated in Figure 1:

- DP1TR: a list of words connecting PROT1 and the dependency tree root.
- DP2TR: a list of words connecting PROT2 and the dependency tree root.
- DP12DT: a list of dependency types connecting the two proteins in the dependency tree.
- DP12: a list of dependent words combined with their dependency types connecting the two proteins in the dependency tree.
- DP12S: the tuple of every word combined with its dependent type in DP12.
- DPFLAG: a boolean value indicating whether the two proteins are directly dependent on each other.

The typed dependencies produced by the Stanford Parser for the sentence "PROT1 contains a sequence motif binds to PROT2." are listed as follows:

```
nsubj(contains-2,PROT1-1)
det(motif-5, a-3)
nn(motif-5, sequence-4)
nsubj(binds-6, motif-5)
ccomp(contains-2, binds-6)
prep_to(binds-6, PROT2-8)
```

Each word in a dependency tuple is followed by its index in the original sentence, ensuring accurate positioning of the head word and dependent word. Figure 1 shows the dependency tree we construct from the above grammatical relations.

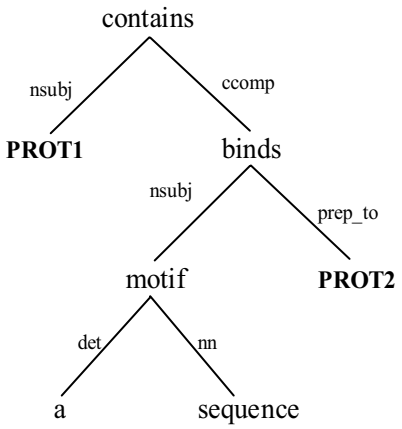


Figure 1: Dependency tree for the sentence "PROT1 contains a sequence motif binds to PROT2."

Erkan et al. (2007) extract the path information between PROT1 and PROT2 in the dependency tree for kernel-based PPI extraction and report promising results, neverthless, such path is so specific for feature-based methods that it may incure higher precision but lower recall. Thus we alleviate this problem by collapsing the feature into multiple ones with finer granularity, leading to the features such as DP12S.

It is widely acknowledged that predicates play an important role in PPI extraction. For example, the change of a pivot predicate between two proteins may easily lead to the polarity reversal of a PPI instance. Therefore, we extract the predicates and their positions in the dependency tree as predicate features called DependencySet2:

- FVW: the predicates in the DP12 feature occurring prior to the first protein.
- LVW: the predicates in the DP12 feature occurring next to the second entity.
- MVW: other predicates in the DP12 features.
- #FVW: the number of FVW
- #LVW: the number of LVW
- #MVW: the number of MVW

4 Experimentation

This section systematically evaluates our feature-based method on the AIMed corpus as well as other commonly used corpus and reports our experimental results.

4.1 Data Sets

We use five corpora[3] with the AIMed corpus as the main experimental data, which contains 177 Medline abstracts with interactions between two interactions, and 48 abstracts without any PPI within single sentences. There are 4,084 protein references and around 1,000 annotated interactions in this data set.

For corpus pre-procession, we first rename two proteins of a pair as PROT1 and PROT2 respectively in order to blind the learner for fair comparison with other work. Then, all the instances are generated from the sentences which contain at least two proteins, that is, if a sentence contains *n* different proteins, there are $\binom{n}{2}$ different pairs of proteins and these pairs are considered untyped and undirected. For the purpose of comparison with previous work, all the self-interactions (59 instances) are removed, while all the PPI instances with nested protein names are retained (154 instances). Finally, 1002 positive instances and 4794 negative instances are generated and their corresponding features are extracted.

We select Support Vector Machines (SVM) as the classifier since SVM represents the state-of-the-art in the machine learning research community. In particular, we use the binary-class SVMLigh [4] developed by Joachims (1998) since it satisfies our requirement of detecting potential PPI instances.

Evaluation is done using 10-fold document-level cross-validation. Particularly, we apply the extract same 10-fold split that was used by Bunescu et al. (2005) and Giuliano et al. (2006). Furthermore, OAOD (One Answer per Occurrence in the Document) strategy is adopted, which means that the correct interaction must be extracted for each occurrence. This guarantees the maximal use of the available data, and more important, allows fair comparison with earlier relevant work.

The evaluation metrics are commonly used Precision (P), Recall (R) and harmonic F1-score (F1). As an alternative to F1-score, the AUC (*area under the receiver operating characteristics curve*) measure is proved to be invariant to the class distribution of the training dataset. Thus we also provide AUC scores for our system as Airola et al. (2008) and Miwa et al. (2009a).

[3] http://mars.cs.utu.fi/PPICorpora/GraphKernel.html
[4] http://svmlight.joachims.org/

4.2 Results and Discussion

Features	P(%)	R(%)	F1
Baseline features			
Words	59.4	40.6	47.6
+Overlap	60.4	39.9	47.4
+Chunk	59.2	44.5	50.6
+Parse	60.9	44.8	51.4
Dependency-driven features			
+Dependency Set1	62.9	48.0	53.9
+Dependency Set2	**63.4**	**48.8**	**54.7**

Table 1: Performance of PPI extraction with various features in the AIMed corpus

We present in Table 1 the performance of our system using document-wise evaluation strategies and 10-fold cross-validation with different features in the AIMed corpus, where the plus sign before a feature means it is incrementally added to the feature set. Table 1 reports that our system achieves the best performance of 63.4/48.8/54.7 in P/R/F scores. It also shows that:

- Words features alone achieve a relatively low performance of 59.4/40.9/47.6 in P/R/F, particularly with fairly low recall score. This suggests the difficulty of PPI extraction and words features alone can't effectively capture the nature of protein interactions.
- Overlap features slightly decrease the performance. Statistics show that both the distributions of #MB and #WB between positives and negatives are so similar that they are by no means the discriminators for PPI extraction. Hence, we exclude the overlap features in the succeeding experiments.
- Chunk features significantly improves the F-measure by 3 units largely due to the increase of recall by 3.9%, though at the slight expense of precision. This suggests the effectiveness of shallow parsing information in the form of headwords captured by chunking on PPI extraction.
- The usefulness of the parse tree features is quite limited. It only improves the F-measure by 0.8 units. The main reason may be that these paths are usually long

and specific, thus they suffer from the problem of data sparsity. Furthermore, some of the parse tree features are already involved in the chunk features.

- The DependencySet1 features are very effective in that it can increase the precision and recall by 2.0 and 3.2 units respectively, leading to the increase of F1 score by 2.5 units. This means that the dependency-related features can effectively retrieve more PPI instances without introducing noise that will severely harm the precision. According to our statistics, there are over 60% sentences with more than 5 words between their protein entities in the AIMed corpus. Therefore, dependency information exhibit great potential to PPI extraction since they can capture long-range dependencies within sentences. Take the aforementioned sentence "PROT1 contains a sequence motif binds to PROT2." as an example, although the two proteins step over a relatively long distance, the dependency path between them is concise and accurate, reflecting the essence of the interaction.
- The predicate features also contribute to the F1-score gain of 0.8 units. It is not surprising since some predicates, such as "interact", "activate" and "inhibit" etc, are strongly suggestive of the interaction polarity between two proteins.

We compare in Table 2 the performance of our system with other systems in the AIMed corpus using the same 10-fold cross validation strategy. These systems are grouped into three distinct classes: feature-based, kernel-based and composite kernels. Except for Airola et al. (2008) Miwa et al. (2009a) and Kim et al. (2010), which adopt graph kernels, our system performs comparably with other systems. In particular, our dependency-driven system achieves the best F1-score of 54.7 among all feature-based systems.

In order to measure the generalization ability of our dependency-driven PPI extraction system across different corpora, we further apply our method to other four publicly available PPI corpora: BioInfer, HPRD50, IEPA and LLL.

Systems	P(%)	R(%)	F1
Feature-based methods			
Our system	63.4	48.8	54.7
Giuliano et al., 2006[5]	60.9	57.2	59.0
Sætre et al., 2007	64.3	44.1	52.0
Mitsumori et al., 2006	54.2	42.6	47.7
Yakushiji et al., 2005	33.7	33.1	33.4
Kernel-based methods			
Kim et al., 2010	61.4	53.3	56.7
Airola et al., 2008	52.9	61.8	56.4
Bunescu et al., 2006	65.0	46.4	54.2
Composite kernels			
Miwa et al., 2009a	-	-	62.0
Miyao et al., 2008[6]	51.8	58.1	54.5

Table 2: Comparison with other PPI extraction systems in the AIMed corpus

The corresponding performance of F1-score and AUC metrics as well as their standard deviations is present in Table 3. Comparative available results from Airola et al. (2008) and Miwa et al. (2009a) are also included in Table 3 for comparison. This table shows that our system performs almost consistently with the other two systems, that is, the LLL corpus gets the best performance yet with the greatest variation, while the AIMed corpus achieves the lowest performance with reasonable variation.

It is well known that biomedical texts exhibit distinct linguistic characteristics from newswire narratives, leading to dramatic performance gap between PPI extraction and relation detection in the ACE corpora. However, no previous work has ever addressed this problem and empirically characterized this difference. In this paper, we devise a series of experiments over the ACE RDC corpora using our dependency-driven feature-based method as a touchstone task. In order to do that, a sub-

[5] Airola et al. (2008) repeat the method published by Giuliano et al. (2006) with a correctly preprocessed AIMed and reported an F1-score of 52.4%.

[6] The results from Table 1 (Miyao et al., 2009) with the most similar settings to ours (Stanford Parser with SD representation) are reported.

set of 5796 relation instances is randomly sampled from the ACE 2003 and 2004 corpora respectively. The same cross-validation and evaluation metrics are applied to these two sets as PPI extraction in the AIMed corpus.

	Our system				Airola et al. (2008) [7]				Miwa et al. (2009a)			
Corpus	F1	σ_{F1}	AUC	σ_{AUC}	F1	σ_{F1}	AUC	σ_{AUC}	F1	σ_{F1}	AUC	σ_{AUC}
AIMed	54.7	4.5	82.4	3.5	56.4	5.0	84.8	2.3	60.8	6.6	86.8	3.3
BioInfer	59.8	3.5	80.9	3.3	61.3	5.3	81.9	6.5	68.1	3.2	85.9	4.4
HPRD50	64.9	13.4	79.8	8.5	63.4	11.4	79.7	6.3	70.9	10.3	82.2	6.3
IEPA	62.1	6.2	74.8	6.6	75.1	7.0	85.1	5.1	71.7	7.8	84.4	4.2
LLL	78.1	15.8	85.1	8.3	76.8	17.8	83.4	12.2	80.1	14.1	86.3	10.8

Table 3: Comparison of performance across the five PPI corpora

Features	AIMed			ACE2003			ACE2004		
	P(%)	R(%)	F1	P(%)	R(%)	F1	P(%)	R(%)	F1
Words	59.4	40.6	47.6	66.5	51.6	57.9	68.1	59.6	63.4
+Overlap	+1.0	-0.7	-0.2	+5.4	+1.8	+3.2	+4.6	+1.2	+2.7
+Chunk	-1.7	+4.6	+3.2	+2.3	+5.1	+4.0	+1.5	+1.9	+1.7
+Parse	+1.7	+0.3	+0.8	+0.3	+0.6	+0.5	+0.6	+0.4	+0.5
+Dependency Set1	+2.0	+3.2	+2.5	+0.8	+0.7	+0.7	+0.5	+0.9	+0.7
+Dependency Set2	+0.5	+0.8	+0.8	+0.3	+0.2	+0.3	+0.2	+0.4	+0.3

Table 4: Comparison of contributions of different features to relation detection across multiple domains

Table 4 compares the performance of our method over different domains. The table reports that the words features alone achieve the best F1-score of 63.4 in ACE2004 but the lowest F1-score of 47.6 in AIMed. This suggests the wide difference of lexical distribution between these domains. We extract the words appearing before the 1st mention, between the two mentions and after the 2nd mention from the training sets of these corpora respectively, and summarize the statistics (the number of tokens, the number of occurrences) in Table 5, where the KL divergence between positives and negatives is summed over the distribution of the 500 most frequently occurring words.

Statistics	AIMed	ACE2003	ACE2004
# of tokens	2,340	2,064	2,099
# of occurrences	69,976	53,744	49,570
KL divergence	0.22	0.28	0.33

Table 5: Lexical statistics on three corpora

The table shows that AIMed uses the most kinds of words and the most words around the two mentions than the other two. More important, AIMed has the least distribution difference between the words appearing in positives and negatives, as indicated by its least KL divergence. Therefore, the lexical words in AIMed are less discriminative for relation detection than they do in the other two. This naturally explains the reason why the performance by words feature alone is AIMed<ACE2003<ACE2004. In addition, Table 4 also shows that:

- The overlap features significantly improve the performance in ACE while slightly deteriorating that in AIMed. The reason is that, as indicated in Zhou et al. (2005), most of the positive relation instances in ACE exist in local contexts, while the positive interactions in AIMed occur in relative long-range just as the negatives, therefore these features are not discriminative for AIMed.
- The chunk features consistently greatly boost the performance across multiple corpora. This implies that the headwords in chunk phrases can well capture the partial nature of relation instances regardless of their genre.
- It's not surprising that the parse feature attain moderate performance gain in all domains since these parse paths are usually

[7] The performance results of F1 and AUC on the BioInfer corpus are slightly adjusted according to Table 3 in Miwa et al. (2009b)

long and specificity, leading to data sparseness problem.

- It is interesting to note that the dependency-related features exhibit more significant improvement in AIMed than that in ACE. The reason may be that, these dependency features can effectively capture long-range relationships prevailing in AIMed, while in ACE a large number of local relationships dominate the corpora.

5 Related Work

Among feature-based methods, the PreBIND system (Donaldson et al., 2003) uses words and word bi-grams features to identify the existence of protein interactions in abstracts and such information is used to enhance manual expert reviewing for the BIND database. Mitsumori et al. (2006) use SVM to extract protein-protein interactions, where bag-of-words features, specifically the words around the protein names, are employed. Sugiyama et al. (2003) extract various features from the sentences based on the verbs and nouns in the sentences such as the verbal forms, and the part-of-speech tags of the 20 words surrounding the verb. In addition to word features, Giuliano et al. (2006) extract shallow linguistic information such as POS tag, lemma, and orthographic features of tokens for PPI extraction. Unlike our dependency-driven method, these systems do not consider any syntactic information.

For kernel-based methods, there are several systems which utilize dependency information. Erkan et al. (2007) defines similarity functions based on cosine similarity and edit distance between dependency paths, and then incorporate them in SVM and KNN learning for PPI extraction. Airola et al. (2008) introduce all-dependency-paths graph kernel to capture the complex dependency relationships between lexical words and attain significant performance boost at the expense of computational complexity. Kim et al. (2010) adopt walk-weighted subsequence kernel based on dependency paths to explore various substructures such as e-walks, partial match, and non-contiguous paths. Essentially, their kernel is also a graph-based one.

For composite kernel methods, Sætre et al. (2007) combine a "bag-of-words" kernel with dependency and PAS (Predicate Argument Structure) tree kernels to exploit both the words features and the structural syntactic information. Hereafter, Miyao et al. (2008) investigate the contribution of various syntactic features using different representations from dependency parsing, phrase structure parsing and deep parsing by different parsers. Miwa et al. (2009a) integrate "bag-of-words" kernel, PAS tree kernel and all-dependency-paths graph kernel to achieve the higher performance. They (Miwa et al., 2009b) also use similar composite kernels for corpus weighting learning across multiple PPI corpora.

6 Conclusion and Future Work

In this paper, we have combined various lexical and syntactic features, particularly dependency information, into a feature-based PPI extraction system. We find that the dependency information as well as the chunk features contributes most to the performance improvement. The predicate features involved in the dependency tree can also moderately enhance the performance. Furthermore, comparative study between biomedical domain and the ACE newswire domain shows that these domains exhibit different lexical characteristics, rendering the task of PPI extraction much more difficult than that of relation detection from the ACE corpora.

In future work, we will explore more syntactic features such as PAS information for feature-based PPI extraction to further boost the performance.

Acknowledgment

This research is supported by Projects 60873150 and 60970056 under the National Natural Science Foundation of China and Project BK2008160 under the Natural Science Foundation of Jiangsu, China. We are also very grateful to Dr. Antti Airola from Truku University for providing partial experimental materials.

References

A. Airola, S. Pyysalo, J. Björne, T. Pahikkala, F. Ginter, and T. Salakoski. 2008. All-paths graph kernel for protein-protein interaction extraction

with evaluation of cross corpus learning. *BMC Bioinformatics*.

R. Bunescu, R. Ge, R. Kate, E. Marcotte, R. Mooney, A. Ramani, and Y. Wong. 2005. Comparative Experiments on learning information extractors for Proteins and their interactions. *Journal of Artificial Intelligence In Medicine,* 33(2).

R. Bunescu and R. Mooney. 2005. Subsequence kernels for relation extraction. In *Proceedings of NIPS'05*, pages 171–178.

A. Culotta and J. Sorensen. 2004. Dependency Tree Kernels for Relation Extraction. In *Proceedings of ACL'04.*

I. Donaldson, J. Martin, B. de Bruijn, C. Wolting, V. Lay, B. Tuekam, S. Zhang, B. Baskin, G. D. Bader, K. Michalockova, T. Pawson, and C. W. V. Hogue. 2003. Prebind and textomy - mining the biomedical literature for protein-protein interactions using a support vector machine. *Journal of BMC Bioinformatics*, 4(11).

G. Erkan, A. Özgür, and D.R. Radev. 2007. Semi-Supervised Classification for Extracting Protein Interaction Sentences using Dependency Parsing, In *Proceedings of EMNLP-CoNLL'07*, pages 228–237.

C. Giuliano, A. Lavelli, and L. Romano. 2006. Exploiting Shallow Linguistic Information for Relation Extraction from Biomedical Literature. In *Proceedings of EACL'06*, pages 401–408.

S. Kim, J. Yoon, J. Yang, and S. Park. 2010. Walk-weighted subsequence kernels for protein-protein interaction extraction. *Journal of BMC Bioinformatics*, 11(107).

J. Li, Z. Zhang, X. Li, and H. Chen. 2008. Kernel-Based Learning for Biomedical Relation extraction. *Journal of the American Society for Information Science and Technology*, 59(5).

T. Mitsumori, M. Murata, Y. Fukuda, K. Doi, and H. Doi. 2006. Extracting protein-protein interaction information from biomedical text with SVM. *IEICE Transactions on Information and System,* E89-D (8).

M. Miwa, R. Sætre, Y. Miyao, and J. Tsujii. 2009a. Protein-Protein Interaction Extraction by Leveraging Multiple Kernels and Parsers. *Journal of Medical Informatics,* 78(2009).

M. Miwa, R. Sætre, Y. Miyao, and J. Tsujii. 2009b. A Rich Feature Vector for Protein-Protein Interaction Extraction from Multiple Corpora. In *Proceedings of EMNLP'09*, pages 121–130.

Y. Miyao, R. Sætre, K. Sagae, T. Matsuzaki, and J.Tsujii. 2008. Task-oriented evaluation of syntactic parsers and their representations. In *Proceedings of ACL'08*, pages 46–54.

T. Ono, H. Hishigaki, A. Tanigami, and T. Takagi. 2001. Automated extraction of information on protein-protein interactions from the biological literature. *Journal of Bioinformatics*, 17(2).

K. Sugiyama, K. Hatano, M. Yoshikawa, and S. Uemura. 2003. Extracting information on protein-protein interactions from biological literature based on machine learning approaches. *Journal of Genome Informatics*, (14): 699–700.

R. Sætre, K. Sagae, and J. Tsujii. 2007. Syntactic features for protein-protein interaction extraction. In *Proceedings of LBM'07*, pages 6.1–6.14.

A. Yakushiji, M. Yusuke, T. Ohta, Y. Tateishi, J. Tsujii. 2006. Automatic construction of predicate-argument structure patterns for biomedical information extraction. In *Proceedings of EMNLP'06*, pages 284–292.

S.B. Zhao and R. Grishman. 2005. Extracting Relations with Integrated Information Using Kernel Methods. In *Proceedings of ACL'05*, pages 419-426.

G.D. Zhou, J. Su, J. Zhang, and M. Zhang. 2005. Exploring various knowledge in relation extraction. In *Proceedings of ACL'05*, pages 427-434.

A Review Selection Approach for Accurate Feature Rating Estimation

Chong Long[†] **Jie Zhang**[‡] **Xiaoyan Zhu**[†§]
[†] State Key Laboratory on Intelligent Technology and Systems,
Tsinghua National Laboratory for Information Science and Technology,
Department of Computer Science, Tsinghua University
[‡]School of Computer Engineering, Nanyang Technological University
[§]{Corresponding Author: zxy-dcs@tsinghua.edu.cn}

Abstract

In this paper, we propose a review selection approach towards accurate estimation of feature ratings for services on participatory websites where users write textual reviews for these services. Our approach selects reviews that comprehensively talk about a feature of a service by using information distance of the reviews on the feature. The rating estimation of the feature for these selected reviews using machine learning techniques provides more accurate results than that for other reviews. The average of these estimated feature ratings also better represents an accurate overall rating for the feature of the service, which provides useful feedback for other users to choose their satisfactory services.

1 Introduction

Most of participatory websites such as Amazon (amazon.com) do not collect from users feature[1] ratings for services, simply because it may cost users too much effort to provide detailed feature ratings. Even for a very few websites that do collect feature ratings such as a popular travel website TripAdvisor (tripadvisor.com), a big portion (approximately 43%) of users may still not provide them. However, feature ratings are useful for users to make informed consumption decisions especially in the case where users may be interested more in some particular features of the services. Machine learning techniques have been proposed for sentiment classification (Pang et al., 2002; Mullen and Collier, 2004) based on annotated samples from experts, but they have limited performance especially when estimating ratings of a multi-point scale (Pang and Lee, 2005).

In this paper, we propose a novel review selection approach for accurate feature rating estimation. More specifically, our approach selects reviews written by the users who comprehensively talk about a certain feature of a service - that are comprehensive on this feature, using information distance of reviews on the feature based on Kolmogorov complexity (Li and Vitányi, 1997). This feature is obviously important to the users. People tend to be more knowledgable in the aspects they consider important. These users therefore represent a subset of experts. Statistical analysis reveals that these expert users are more likely to agree on a common rating for the feature of the service. The rating estimation of the feature for these selected reviews based on annotated samples from experts using machine learning techniques is thus able to provide more accurate results than that for other reviews. This statistical evidence also allows us to use the average of the estimated feature ratings to better represent an overall opinion of experts for the feature of the service, which will be particularly useful for assisting other users to correctly make their consumption decisions.

We verify our approach and arguments based on real data collected from the TripAdvisor website. First, our approach is shown to be able to effectively select reviews that comprehensively talk about features of a service. We then adopt the machine learning method proposed in (Pang and Lee, 2005) and the Bayesian Network classifier (Russell and Norvig, 2002) for feature rating estimation. Our experimental results show that the accuracy of estimating feature ratings for these selected reviews is higher than that for other reviews, for both the machine learning

[1]A feature broadly means an attribute or a function of a service.

Coling 2008: Poster Volume, pages 766–774,
Beijing, August 2010

methods. And, the average of these estimated ratings is testified to closely represent the overall feature rating of the service. Our approach is therefore verified to be a successful step towards accurate feature rating estimation.

2 Related Work

Our work aims at estimating feature ratings of a service based on its textual reviews. It is related to sentiment classification. The task of sentiment classification is to determine the semantic orientations of words, sentences or documents. (Pang et al., 2002) is the earliest work of automatic sentiment classification at document level, using several machine learning approaches with common textual features to classify movie reviews. Mullen and Collier (Mullen and Collier, 2004) integrated PMI values, Osgood semantic factors and some syntactic relations into the features of SVM. Pang and Lee (Pang and Lee, 2004) proposed another machine learning method based on subjectivity detection and minimum-cut in graph. However, these approaches focus only on binary classification of reviews.

In 2005, Pang and Lee extended their earlier work in (Pang and Lee, 2004) to determine a reviewer's evaluation with respect to multi-scales (Pang and Lee, 2005). The rating estimation is viewed as multi-class sentiment categorization on documents. They used SVM regression as the multi-class classifier, and also applied a meta-algorithm based on a metric labeling formulation of the problem, which alters a given n-ary classifier's output in an explicit attempt to ensure that similar items receive similar labels. They collected movie reviews from a website named IMDB and tested the performance of their classifier under both four-class and five-class categorization. The five-class sentiment classification is adopted in the evaluation of our method (see Section 5). The performance of their approach is limited. One important reason is that their method considers every review when estimating a feature rating of a movie. However, some reviews do not contain much of the users' opinions about a certain feature simply because the users do not care much or are not knowledgable about the feature. In our work, we study the characteristics of reviews' feature ratings. We investigate which reviews are more useful for us to estimate feature ratings. From some observations stated in the next section, we will see that reviews written by different users reflect their own preferred features of a service.

3 Accurate Feature Rating Estimation

Participatory websites allow users to write textual reviews to discuss features of services that they have consumed. These reviews usually contain words that strongly express the users' opinions about the corresponding features. These words contain important information for estimating a numerical rating for the feature. The estimated ratings can be used for assisting other users when they need to choose which services to consume. Machine learning techniques are often used for training a learner based on annotated samples from experts and estimating a rating for a feature discussed in a review. However, for a review that does not mention a feature or discusses it only in a limited sense, the estimation accuracy is expected to be very low. Besides, the opinion expressed by the user who writes this kind of review is not representative because this user obviously does not care much about the feature. We believe that if we carefully select reviews for estimating feature ratings, the accuracy will be increased and the estimated ratings will be more representative.

We then statistically analyze real data collected from the TripAdvisor website. The results reveal that users who comprehensively discuss a feature of a service in their reviews are more likely to agree on a common rating for this feature of the service. This phenomenon can also be intuitively explained as follows. For the users who comprehensively discuss about a feature, the feature is obviously more important to them. People tend to be more knowledgable in the aspects they consider important. These users therefore represent a subset of experts. Experts likely provide more objective and representative feedback about the feature, and therefore the ratings from them for the feature contain less noise and

are more similar.

Based on the above discussion that experts tend to have similar opinions on a feature of a service, a learner trained by a machine learning technique based on annotated samples from experts should then be able to more accurately estimate the feature ratings from reviews written by other experts. Since the opinions of experts converge, the average of the estimated feature ratings also better represents an overall rating for the feature of the service.

We propose a review selection approach using information distance of reviews on the feature based on Kolmogorov complexity, to select reviews that comprehensively discuss a feature of a service. We rank the reviews based on the comprehensiveness on the feature. The top reviews will be selected for the estimation of feature ratings. Also, the average of these estimated feature ratings will be used for representing the overall rating for the feature. Next, we will first describe in detail how our approach selects comprehensive reviews on a given feature.

4 Our Review Selection Approach

Our review selection approach selects reviews that comprehensively talk about a feature. According to this definition, a review's comprehensiveness depends on the amount of information discussed on a feature. We use Kolmogorov complexity and information distance to measure the amount of information. Kolmogorov complexity was introduced almost half a century ago by R. Solomonoff, A.N. Kolmogorov and G. Chaitin, see (Li and Vitányi, 1997). It is now widely accepted as an information theory for individual objects parallel to that of Shannon's information theory which is defined on an ensemble of objects.

4.1 Theory

Fix a universal Turing machine U. The Kolmogorov complexity (Li and Vitányi, 1997) of a binary string x condition to another binary string y, $K_U(x|y)$, is the length of the shortest (prefix-free) program for U that outputs x with input y. It can be shown that for different universal Turing machine U', for all x, y

$$K_U(x|y) = K_{U'}(x|y) + C,$$

where the constant C depends only on U'. Thus $K_U(x|y)$ can be simply written as $K(x|y)$. They write $K(x|\epsilon)$, where ϵ is the empty string, as $K(x)$. It has also been defined in (Bennett et al., 1998) that the energy to convert between x and y to be the smallest number of bits needed to convert from x to y and vice versa. That is, with respect to a universal Turing machine U, the cost of conversion between x and y is:

$$E(x, y)=\min\{|p|: U(x, p)=y,\ U(y, p)=x\} \quad (1)$$

It is clear that $E(x, y) \leq K(x|y) + K(y|x)$. From this observation, the following theorem has been proved in (Bennett et al., 1998):

Theorem 1 $E(x, y) = \max\{K(x|y), K(y|x)\}$.

Thus, the max distance was defined in (Bennett et al., 1998):

$$D_{\max}(x, y) = \max\{K(x|y), K(y|x)\}. \quad (2)$$

This distance is shown to satisfy the basic distance requirements such as positivity, symmetricity, triangle inequality and is admissible.

Here for an object x, we can measure its information by Kolmogorov complexity $K(x)$; for two objects x and y, their shared information can be measured by information distance $D(x, y)$. In (Long et al., 2008), the authors generalize the theory of information distance to more than two objects. Similar to Equation 1, given strings $x_1, \ldots, x_n$, they define the minimum amount of thermodynamic energy needed to convert from any x_i to any x_j as:

$$E_m(x_1, .., x_n)=\min\{|p|:U(x_i, p, j)=x_j \text{ for all } i,j\}$$

Then it is proved in (Long et al., 2008) that:

Theorem 2 *Modulo to an $O(\log n)$ additive factor,*

$$\min_i K(x_1 \ldots x_n|x_i) \leq E_m(x_1, \ldots, x_n)$$

Given n objects, the left-hand side of Equation 3 may be interpreted as the most comprehensive object that contains the most information about all of the others.

4.2 Review Selection Method

Our review selection method is based on the information distance discussed in the previous section. However, our problem is that neither the Kolmogorov complexity $K(\cdot,\cdot)$ nor $D_{max}(\cdot,\cdot)$ is computable. Therefore, we find a way to "approximate" these two measures. The most useful information in a review article is the English words that are related to the features. If we can extract all of these related words from the review articles, the size of the word set can be regarded as a rough estimation of information content (or Kolmogorov complexity) of the review articles. In Section 5 we will see that this gives very good practical results.

4.2.1 Outline

Our method is outlined in the following. First, for each type of product or service (such as a hotel), a small set of core feature words (such as price and room) is generated through statistics. Then, these feature words are used to generate the expanded words. Third, a parser is used to find the dependent words associated with the occurrence of the core feature words and expanded words in a review. For each review-feature pair, the union of the core feature words, expanded words and dependent words in the review defines the related word set of the review on the feature. Lastly, information distance is used to select the most comprehensive reviews on a feature.

4.2.2 Word Extraction

Feature words are the most direct and frequent words describing a feature, for example, price, room or service of a hotel. Given a feature, the core feature words are the very few most common English words that are used to refer to that feature. For example, both "value" and "price" are used to refer to the same feature of a hotel. In (Hu and Liu, 2004), the authors indicate that when customers comment on product features, the words they use converge. If we remove the feature words with frequency lower than 1% of the total frequency of all feature words, the remaining words, which are just core feature words, can still cover more than 90% occurrences. So firstly we extract those words through statistics; then some of those with the same meaning (such as "value" and "price") are grouped into one feature. They are just "core feature words".

Apart from core feature words, many other less-frequently used words that are connected to the feature also contribute to the information content of the feature. For example, "price" is an important feature of a hotel, but the word "price" is usually dropped from a sentence. Instead, words such as "$", "dollars", "USD", and "CAD" are used. We use information distance $d(.,.)$ based on Google to expand words (Cilibrasi and Vitányi, 2007). Let α be a feature and $\mathcal{A}$ be the set of its core feature words. The distance between a word w and the feature α is then defined to be

$$d(w,\alpha) = \min_{v\in\mathcal{A}} d(w,v)$$

A distance threshold is then used to determine which words should be in the set of expanded words for a given feature.

If a core feature word or an expanded word is found in a sentence, the words which have grammatical dependent relationship with it are called the dependent words (de Marneffe et al., 2006). For example, in sentence "It has a small, but beautiful room", the words "small" and "beautiful" are both dependent words of the core feature word "room". All these words also contribute to the reviews and are important to determine the reviewer's attitude towards a feature.

The Stanford Parser (de Marneffe et al., 2006) is used to parse each review. For review i and feature j, the core feature words and expanded words in the review are first computed. Then the parsing result is examined to find all the dependent words for the core feature words and expanded words, all of which are called "related words".

4.2.3 Computing Information Distance

If there are m reviews $x_1, x_2, \ldots, x_m$, n features $u_1, u_2, \ldots, u_n$, and the related word set S_i is defined to be the union of all the related words that occur in the review x_i. From the left-hand side of Equation 3, the most comprehensive x_i

on feature u_k is such that

$$i = \arg\min_i K(S_1 \dots S_n | S_i, u_k). \quad (3)$$

Let S_i and S_j be two sets of words,

$$K(S_i S_j | u_k) = K(S_i \cup S_j | u_k),$$

$$K(S_i | S_j, u_k) = K(S_i \setminus S_j | u_k),$$

and

$$K(S_i|u_k) = \sum_w K(w|u_k) \approx \sum_w (K(w, u_k) - K(u_k))$$

where $w \in S_i$ and w is in x_i's related word set on feature u_k. For each word w in a set S, the Kolmogorov complexity can be estimated through coding theorem (Li and Vitányi, 1997):

$$K(w, u_k) = -\log P(w, u_k), \;\; K(u_k) = -\log P(u_k)$$

where $P(w, u_k)$ can be estimated by $df(w, u_k)$, which is the document frequency of word w and feature u_k co-exist on the whole corpus. Similarly, $P(u_k)$ can be estimated by feature u_k's document frequency on the corpus. In the next section, Equation 3 will be used to select reviews that comprehensively talk about a feature.

5 Experimental Verification

In this section, we present a set of experimental results to support our work. Our experiments are carried out using real data collected from the travel website TripAdvisor. This website indexes hotels from cities across the world. It collects feedback from travelers. Feedback of each traveler consists of a textual review written by the traveler and numerical ratings (from 1, lowest, to 5, highest) for different features of hotels (e.g., value, service, rooms).

Table 1: Summary of the Data Set

Location	# Hotels	# Feedback	# Feedback with feature rating
Boston	57	3949	2096
Sydney	47	1370	879
Vegas	40	5588	3144

We crawled this website to collect travelers' feedback for hotels in three cities: Boston, Sydney and Las Vegas. Note that during this crawling process, we carefully removed information about travelers and hotels to protect their privacy. For users' feedback, we recorded only the textual reviews and the numerical ratings on four features: Value(V), Rooms(R), Service(S) and Cleanliness(C). These features are rated by a significant number of users. Table 1 summarizes our data set. For each one of the cities, this table contains information about the number of hotels, the total amount of feedback and the amount of feedback with feature ratings. In general, each hotel has sufficient amount of feedback with feature ratings for us to evaluate our work.

Table 2: Comprehensive Reviews on Each Feature (Boston)

Top #	V	R	S	C
1	Y	Y	Y	Y
2	Y	Y	Y	Y
3	N	Y	Y	N
4	Y	Y	Y	N
5	Y	Y	Y	Y
6	Y	Y	N	Y
⋮	⋮	⋮	⋮	⋮

5.1 Evaluation of Review Selection

We first evaluate the performance of our review selection approach using manually annotated data. More specifically, in our data set, for one city, 40 reviews (120 reviews in total) are selected for manual annotation. The annotator looks over each review and decides whether the review is comprehensive on a given feature. Comprehensive reviews on the feature are annotated as "Y", and the reviews that are not comprehensive on this feature are annotated as "N". For the review set of each city, the number of reviews annotated as comprehensive is equal to or less than 20% of the total number of the selected reviews for this city (eight in this experiment). Note that it is possible that one review can be comprehensive on more than one features.

We then use our review selection approach

discussed in Section 4 to rank the reviews for hotels in each city, according to their comprehensiveness on each feature. For example, the most comprehensive review on the feature "Value", which has the minimal information distance to this feature (see Equation 3), is ranked No.1. Table 2 shows the annotated reviews for Boston hotels that are ranked on top six on each feature. It can be obviously seen from the table that most of these top reviews are labeled as comprehensive reviews on respective features. Our comprehensive review selection approach generally performs well.

Table 3: Performance of Comprehensive Review Selection

City	Feature	Precision	Recall	F-Score
Boston	V	0.833	0.714	0.769
	R	1.000	0.875	0.933
	S	0.857	1.000	0.923
	C	0.833	1.000	0.909
Sydney	V	0.667	1.000	0.800
	R	0.600	0.857	0.706
	S	0.667	0.857	0.750
	C	0.750	1.000	0.857
Vegas	V	0.778	1.000	0.875
	R	0.727	1.000	0.842
	S	0.714	0.714	0.714
	C	0.667	0.800	0.727

To clearly present the performance of our comprehensive review selection approach, we use the measures of precision, recall and f-score. The measure f-score is a single value that can represent the result of our evaluation. It is the harmonic mean of precision and recall. Suppose there are n reviews in total. Let p_{jk} $(1 \leq k \leq n)$ be the review ranked the kth comprehensive on feature j. Define

$$z_{jk} = \begin{cases} 1 & \text{if } p_{jk} \text{ is labelled comprehensive on } j; \\ 0 & \text{otherwise.} \end{cases}$$

The precision P, recall R, and f-score F of top k comprehensive reviews on feature j are formalized as follows

$$P_{jk} = \frac{\sum_{l=1}^{k} z_{jl}}{k}, R_{jk} = \frac{\sum_{l=1}^{k} z_{jl}}{\sum_{l=1}^{N} z_{jl}},$$

$$F_{jk} = \frac{2P_{jk}R_{jk}}{P_{jk} + R_{jk}}$$

For each ranked review set on feature j, the maximum F_{jk} and its associated P_{jk} and R_{jk} are listed in Table 3. From this table, it can be seen that for the best f-scores, the precision and recall values are mostly larger than 70%, that is, a great part of reviews that are labeled as comprehensive receive top rankings from our comprehensive review selection approach. Our approach is thus carefully verified to be able to accurately select comprehensive reviews on any given feature.

5.2 Statistical Analysis

A group of users who comprehensively discuss a certain feature are more likely to agree on a common rating for that feature. In this experiment, we use our review selection approach to verify this argument.

Table 4: Deviation of Feature Ratings

City	Feature	20%	50%	All
Boston	V	0.884 (0.0003)	1.030	1.136
	R	0.940 (0.2248)	1.037	1.013
	S	1.026 (0.0443)	1.130	1.144
	C	0.798 (0.0093)	0.892	0.949
Sydney	V	0.862 (0.0266)	1.009	1.054
	R	0.788 (0.0497)	0.932	0.945
	S	0.941 (0.0766)	1.162	1.116
	C	0.651 (0.0037)	0.905	0.907
Vegas	V	0.845 (0.0002)	1.236	1.291
	R	1.105 (0.2111)	1.148	1.175
	S	1.112 (0.0574)	1.286	1.269
	C	0.936 (0.0264)	1.096	1.158

More specifically, for each city, hotels that receive no less than 10 reviews with feature ratings are selected. We use our comprehensive review selection approach to select top 20% and 50% comprehensive reviews on each feature for hotels in each city. We calculate the standard deviation of their feature ratings, as well as that of all feature ratings, for each hotel in a city. We then average these standard deviations over the hotels in the same city. The average values are listed in Table 4. The feature ratings of comprehensive reviews on the feature have smaller average stan-

dard deviations. Standard T-test is used to measure the significance of the results between top 20% comprehensive reviews and all reviews, city by city and feature by feature. Their p-values are shown in the braces, and they are significant at the standard 0.05 significance threshold. It can be seen from the table that although for some items there does not seem to be a significant difference, the results are significant for the entire data set.

Therefore, when these travelers write reviews that are comprehensive on one feature, their ratings for this feature tend to converge. This evidence indicates that the estimation of ratings for the feature from these comprehensive reviews can provide better results, which will be confirmed in Section 5.3. These estimated feature ratings can also be averaged to represent a specific opinion of these travelers on the feature, which will be verified in Section 5.4.

5.3 Feature Rating Estimation

In this section, we carry out experiments to testify that the estimation of feature ratings for comprehensive reviews using our review selection approach provides better performance than that for all reviews. We adopt the approach of Pang and Lee (Pang and Lee, 2005) described in Section 2 for feature rating estimation. In short, they applied a meta-algorithm, based on a metric labeling formulation of the problem to alter a given n-ary SVM's output in an explicit attempt. We also adopt a Baysian Network classifier for feature rating estimation.

Similar to the method of Pang and Lee, we build up a feature rating classification system to estimate reviews' feature ratings. However, the method of Pang and Lee focuses only on single rating classification for a review and assumes that every word of the review can contribute to this single rating. While it comes to feature rating classification, the system has to decide which terms or phrases in the review are talking about this feature. We train a Naive Bayes classifier to retrieve all the sentences related to a feature. Then all the core feature words, expanded words and dependent words are extracted to train a SVM classifier and the Bayesian Network classifier for five-class classification (1 to 5). The eight-fold cross-validation is used to train and test the performance of feature rating estimation on all the reviews and the top 20% comprehensive reviews, respectively.

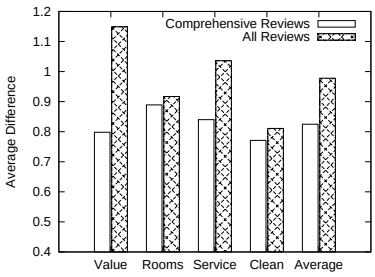


Figure 1: Average Error of Feature Rating Estimation for the Adopted Method of Pang and Lee

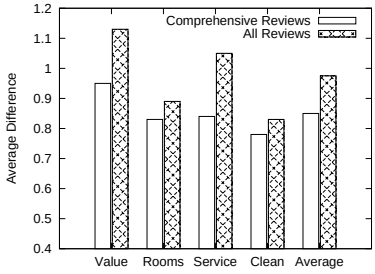


Figure 2: Average Error of Feature Rating Estimation for the Bayesian Network classifier

We formalize a performance measure as follows. Suppose there are n reviews in total. For a test review i ($1 \leq i \leq n$), its real feature rating (given by the review writer) is f_i, and its predicated feature rating (predicted by our classification system) is g_i. Both f_i and g_i are integers between 1 and 5. The performance of the classification on all n reviews can be measured by the average of the absolute difference (d) between each f_i and g_i pair,

$$d = \frac{\sum_{i=1}^{n} |f_i - g_i|}{n}. \quad (4)$$

The lower d is, the better performance the classifier can provide.

Figures 1 and 2 show the results for the performance of feature rating estimation on all reviews versus that on selected comprehensive reviews,

for the adopted approach of Pang and Lee and the Baysian Network classifier respectively. It can be seen that the average difference between real feature ratings and estimated feature ratings on each feature when using selected comprehensive reviews is significantly lower than that when using all reviews, for both the approaches. On average, the performance of feature rating estimation is improved by more than 12.5% using our review selection approach. And, our review selection approach is generally applicable to different classifiers.

5.4 Estimating Overall Feature Rating

Supported by the statistical evidence verified in Section 5.2 that the users who write comprehensive reviews on one feature will more likely agree on a common rating for this feature, we can then use an average of the feature ratings for top 20% comprehensive reviews to reflect a general opinion of knowledgable/expert users. In this section, we show directly the performance of estimating an overall feature rating for a hotel using ratings for the selected comprehensive reviews, and compare it with that for all reviews.

Table 5: Performance of Estimating Overall Feature Rating for Comprehensive Reviews

City	V	R	S	C	AVG
Boston	0.637	0.426	0.570	0.660	0.573
Sydney	0.273	0.729	0.567	0.680	0.562
Vegas	0.485	0.502	0.277	0.613	0.469
Average	0.465	0.552	0.471	0.651	0.535

Table 6: Performance of Estimating Overall Feature Rating for All reviews

City	V	R	S	C	AVG
Boston	0.809	0.791	0.681	0.642	0.731
Sydney	0.433	0.886	0.588	0.593	0.625
Vegas	0.652	0.733	0.502	0.942	0.707
Average	0.631	0.803	0.590	0.726	0.688

Suppose there are m hotels. For each hotel j, we first select the top 20% comprehensive reviews on each feature using our review selection approach. We average the real ratings of one feature provide by travelers for these reviews, denoted as $\bar{f}_j$. We then estimate the feature ratings for these comprehensive reviews using the adopted machine learning method of Pang and Lee. The average of these estimated ratings is denoted as $\bar{g}_j$. Similar to Equation 4, the average difference between all $\bar{f}_j$ and $\bar{g}_j$ pairs on each feature for hotels in each city are calculated and listed in Table 5. From this table, we can see that the average difference between the estimated average feature rating and real average feature rating is only about 0.53. Our review selection approach produces fairly good performance for estimating an overall feature rating for a hotel. We then also calculate the average difference for all reviews. The results are listed in Table 6. We can see that the average difference is larger (about 0.69) in this case. The performance of estimating an overall feature rating is increased by nearly 23.2% through our review selection approach.

6 Conclusion

In this paper, we presented a novel review selection approach to improve the accuracy of feature rating estimation. We select reviews that comprehensively talk about a feature of one service, using information distance of reviews on the feature based on Kolmogorov complexity. As evaluated using real data, the rating estimation for the feature from these reviews provides more accurate results than that for other reviews, independent of which classifiers are used. The average of these estimated feature ratings also better represents an accurate overall rating for the feature of the service.

In future work, we will further improve the accuracy of estimating a general rating for a feature of a service based on the selected comprehensive reviews on this feature using our review selection approach. Comprehensive reviews may contribute differently to the estimation of an overall feature rating. In our next step, a more sophisticated model will be developed to assign different weights to these different reviews.

References

Bennett, C.H., P Gacs, M Li, P.M.B. Vitányi, and W.H. Zurek. 1998. Information distance. *IEEE Transactions on Information Theory*, 44(4):1407–1423, July.

Cilibrasi, Rudi L. and Paul M.B. Vitányi. 2007. The google similarity distance. *IEEE Transactions on Knowledge and Data Engineering*, 19(3):370–383, March.

de Marneffe, Marie Catherine, Bill MacCartney, and Christopher D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *The fifth international conference on Language Resources and Evaluation (LREC)*, May.

Hu, Minqing and Bing Liu. 2004. Mining and summarizing customer reviews. In *10th ACM International Conference on Knowledge Discovery and Data Mining*, pages 168–177.

Li, M. and P. Vitányi. 1997. *An Introduction to Kolmogorov Complexity and its Applications*. Springer-Verlag.

Long, Chong, Xiaoyan Zhu, Ming Li, and Bin Ma. 2008. Information shared by many objects. In *ACM 17th Conference on Information and Knowledge Management*.

Mullen, Tony and Nigel Collier. 2004. Sentiment analysis using support vector machines with diverse information sources. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 412–418, July.

Pang, Bo and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Annual Meeting of the Association of Computational Linguistics (ACL)*, pages 271–278, July.

Pang, Bo and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Annual Meeting of the Association of Computational Linguistics (ACL)*, pages 115–124, June.

Pang, Bo, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up? sentiment classification using machine learning techniques. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 79–86, July.

Russell, S. and P. Norvig. 2002. *Artificial Intelligence: A Modern Approach*. Second Edition, Prentice Hall, Englewood Cliffs, New Jersey.

Connective-Based Measuring of the Inter-Annotator Agreement in the Annotation of Discourse in PDT

Jiří Mírovský, Lucie Mladová, Šárka Zikánová
Charles University in Prague
Institute of Formal and applied Linguistics
{mirovsky,mladova,zikanova}@ufal.mff.cuni.cz

Abstract

We present several ways of measuring the inter-annotator agreement in the ongoing annotation of semantic inter-sentential discourse relations in the Prague Dependency Treebank (PDT). Two ways have been employed to overcome limitations of measuring the agreement on the exact location of the start/end points of the relations. Both methods – skipping one tree level in the start/end nodes, and the connective-based measure – are focused on a recognition of the existence and of the type of the relations, rather than on fixing the exact positions of the start/end points of the connecting arrows.

1 Introduction

1.1 Prague Dependency Treebank 2.0

The Prague Dependency Treebank 2.0 (PDT 2.0; Hajič et al., 2006) is a manually annotated corpus of Czech. It belongs to the most complex and elaborate linguistically annotated treebanks in the world. The texts are annotated on three layers of language description: morphological, analytical (which expresses the surface syntactic structure), and tectogrammatical (which expresses the deep syntactic structure). On the tectogrammatical layer, the data consist of almost 50 thousand sentences.

For the upcoming release of PDT, many additional features are planned, coming as results of several projects. Annotation of semantic inter-sentential discourse relations is one of the planned additions.

To ensure the highest possible quality of the annotated data, it would be best if several annotators annotated the whole data in parallel. After solving discrepancies in the annotations of the individual annotators, we would get a high-quality annotation. This approach is sometimes employed, but most of the times, the available resources prohibit it (which is also the case of the discourse annotation project). Manual annotation of data is a very expensive and time consuming task. To overcome the restriction of limited resources, each part of the data is annotated by one annotator only, with the exception of a small overlap for studying and measuring the inter-annotator (dis-)agreement.

1.2 Inter-Annotator Agreement in Computational Linguistics

Measuring the inter-annotator agreement has long been studied (not only) in computational linguistics. It is a complex field of research and different domains require different approaches.

Classical measures *recall*, *precision* and *F-measure* offer the most straightforward and intuitively interpretable results. Since they do take into account neither the contribution of chance in agreement, nor different importance of different types of disagreement, etc., other more or less elaborate coefficients for measuring the inter-annotator agreement have been developed. Cohen's κ (Cohen, 1960) is suitable for classification tasks and tries to measure the agreement "above chance". Krippendorff's α (Krippendorff, 1980) can be used if we need to distinguish various levels of disagreement. Rebecca Passonneau (2004) offered a solution for measuring agreement between sets of elements (like words in coreferential chains). Variants of these coefficients can be used for measuring agreement among more than two annotators. A comprehensive overview of methods for measuring the inter-annotator agreement in various areas of

Coling 2008: Poster Volume, pages 775–781,
Beijing, August 2010

computational linguistics was given in Artstein and Poesio (2008).

For measuring the inter-annotator agreement in the annotation of semantic inter-sentential discourse relations in PDT, we have chosen two measures. The relations do not form natural chains (unlike e.g. textual and grammatical coreference) and a simple *F_1-measure* is well suited for the agreement on existence of the relations. For the agreement on types of the relations, which is a typical classification task, we use Cohen's κ.

Our research has then been focused not on "how to measure" the agreement (which coefficient to use), but rather on "what to measure" (which phenomena), which is the topic of this paper.

2 Annotated Phenomena

Since the Prague Dependency Treebank 2.0 already contains three layers of linguistic annotation, two of which (the analytical layer – surface syntax, and the tectogrammatical layer – underlying syntax and semantics) are tree representations, we took advantage of these existing analyses and carry out the annotation of discourse phenomena directly on the trees (the tectogrammatical layer). It means that we capture the discourse relation between any two (sub)trees in the document by drawing a link (an arrow) between the highest nodes in the (sub)trees, see Figure 1.

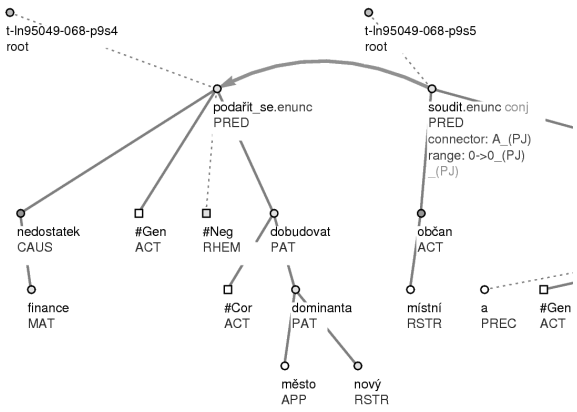


Figure 1. A discourse arrow between two nodes represents a discourse relation between two trees – subtrees of the nodes.

Discourse relations we annotate are in principle semantic relations that apply between two abstract objects (Asher, 1993) (i.e. discourse units or text spans) and help make the text a coherent whole. These relations are often signaled by the presence of a discourse connective, i.e. expressions as *"ale"*, *ačkoliv"*, *"tedy"*, *"ovšem"* (in English *"but"*, *"although"*, *"then"*, *"however"* etc. In the first phase of the project, we only annotate relations (link the (sub)trees) where such a connective is present.

Every relation gets assigned two important attributes: first, the discourse connective that anchors the relation, and, second, the semantic type of the relation. For assigning semantic relations in the discourse, we developed a set of 22 discourse-semantic tags (Mladová et al., 2009). It is inspired partly by the set of semantic labels used for the annotation of the tectogrammatical layer in PDT 2.0, relations within the sentence (the tectogrammatical syntactico-semantic labels called functors, Mikulová et al., 2005) – since some of the semantic relations apply also intra-sententially, like causal or contrastive relations; and partly by the set of semantic tags in the Penn Discourse Treebank 2.0 (Prasad et al., 2008), a discourse annotation project for English with similar aims.

Hence, there are three important issues for the inter-annotator measurement on the discourse level of annotation in PDT: the agreement on the start and target nodes of the discourse relation (and so the extent of the discourse arguments), the agreement on the discourse connective assigned to the relation, and, last but not least, the agreement on the semantic type of the relation.

3 Measuring the Inter-Annotator Agreement in the Annotation of Discourse in PDT 2.0

3.1 Simple (Strict) Approach

The basic method we use for measuring the inter-annotator agreement requires a perfect match in the start and end points of the relations. We calculate *recall* and *precision* between the two annotators. Since these measures are not symmetric in respect to the annotators, we use their combination – *F_1-measure* – which is symmetric. At each node, we compare target

nodes of the discourse relations created by the two annotators. We consider two relations to be in agreement strictly only if they share both the start node and the target node.

A second number we measure is an agreement on the relation and the type. For considering two relations to be in agreement, we require that they share their start and target nodes, and also have attached the same type.

Similarly, we measure an agreement on the relation and the connective, and an agreement on the relation, the type and the connective.

Attaching a type to a relation can be understood as a classification task. We calculate two numbers – simple ratio agreement and Cohen's κ – on the types attached to those relations where the annotators agreed on the start and the target nodes. Cohen's κ shows the level of agreement on the types above chance.

For completeness, we also calculate simple ratio agreement on the connectives attached to those relations the annotators agreed on.

Table 1 shows results of these measurements on two hundred sentences annotated in parallel by two annotators.[1]

measure	value
F_1-measure on relations	0.43
F_1-measure on relations + types	0.34
F_1-measure on relations + connectives	0.41
F_1-measure on rel. + types + connect.	0.32
agreement on types	0.8
agreement on connectives	0.95
Cohen's κ on types	0.74

Table 1. The inter-annotator agreement for a strict match.

3.2 Skipping a Tree Level

Requiring a perfect agreement on the start node and the target node of the discourse relations turns out to be too strict for a fair evaluation of the inter-annotator agreement. It often happens that the annotators recognize the same discourse relation in the data but they disagree either in the start node or the target node of the relation.

In Zikánová et al. (2010), we elaborate on typical cases of this type of disagreement and show that in many times, the difference in the start node or the target node is only one level in the tree. We have also shown that these disagreements usually depend on a subtle and not crucial difference in the interpretation of the text.

Figure 2 shows an example of a disagreement caused by a one-level difference in the target node of a relation. The two trees (a cut of them) represent these two sentences:

"Vím, že se nás Rusů bojíte, že nás nemáte rádi, že námi trochu pohrdáte. Ale Rusko není jenom Žirinovskij, Rusko není jenom vraždění v Čečensku."

(In English: *"I know that you are afraid of us Russians, that you dislike us, that you despise us a little. But Russia is not only Zhirinovsky, Russia is not only murdering in Chechnya."*)

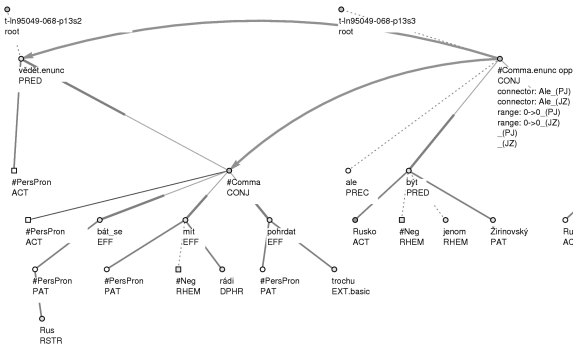


Figure 2. Disagreement in the target node.

Both annotators recognized the discourse relation between the two sentences, both selected the same type (opposition), and both marked the same connective (*"Ale"*, in English *"But"*). The disagreement in the target node is caused by the fact that one annotator has connected the second sentence with *"knowing that something is going on"*, while the other has connected it directly with the expression *"something is going on"*.

We have shown in Zikánová et al. (2010) that allowing for skipping one tree level either at the start node or the target node of the relations leads to an improvement in the inter-annotator

[1] The annotators did not know which part of the data will be used for the measurement. The agreement was measured on 200 sentences (6 documents). PDT 2.0 contains data from three sources. The proportion of the sentences selected for the measurement reflected the total proportion of these data sources in the whole treebank.

agreement (F_1-*measure* on the relations) of about 10%. To be exact, by allowing to skip one tree level we mean: if node A is a parent of node B, then we consider arrows A→C and B→C to be in agreement, as well as arrows D→A and D→B. Table 2 shows present results of this type of measurement, performed on the same data as Table 1.

measure	value
F_1-measure on relations	0.54
F_1-measure on relations + types	0.43
F_1-measure on relations + connectives	0.49
F_1-measure on rel. + types + connect.	0.39
agreement on types	0.8
agreement on connectives	0.92
Cohen's κ on types	0.73

Table 2. The inter-annotator agreement with one-level skipping.

The results seem to be consistent, since the improvement here is similar to the previously published test. The F_1-*measure* on the relations improved from 0.43 to 0.54. On the other hand (and also consistently with the previous test), simple ratio agreement on types (or connectives) and Cohen's κ on types, all measured on those arrows the annotators agreed on, do not change (more or less) after skipping one level is allowed. For these three measures, skipping one level only adds more data to evaluate and does not change conditions of the evaluation.

3.3 Connective-Based Approach

Further studies of discrepancies in parallel annotations show that skipping one level does not cover all "less severe" cases of disagreement.

Figure 3 presents an example of a disagreement in the start node of a relation with a two-level distance between the nodes. The two trees (a cut of them) represent these two sentences:

"Racionální kalkulace vlastníků nájemních bytů je proto povede k jedinému závěru: jakékoliv investice do oprav a modernizace nájemního bytového fondu jsou a budou ztrátové. Proto je další chátrání nájemních domů neodvratné."

(In English: *A rational calculation of the owners of the apartments will lead them to the only conclusion: any investment in repairs and renovation of the rental housing resources is and will be loss-making. Therefore, further dilapidation of the apartment buildings is inevitable."*)

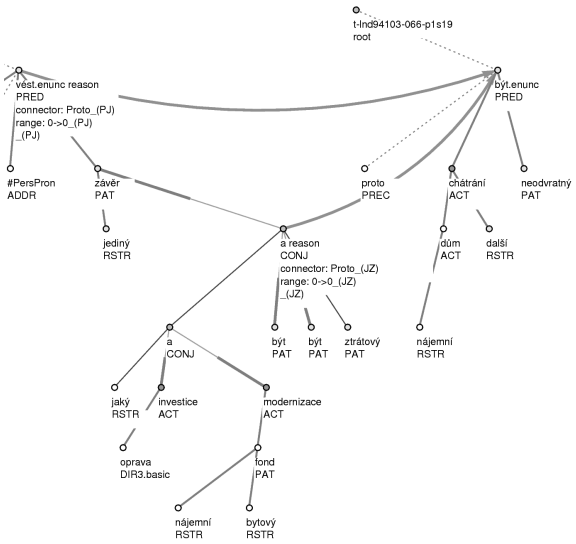


Figure 3. Two-level disagreement in the start nodes

The difference between the annotators is that one of them started the relation at the phrase *"will lead to the only conclusion: any investment ... is and will be ..."*, while the other started the relation directly at the phrase *"any investment ... is and will be ..."*.

However, both the annotators admittedly recognized the existence of the discourse relation, they also selected the same type (reason), and marked the same connective (*"Proto"*, in English *"Therefore"*).

Figure 4 shows an example of a disagreement caused by a different selection of nodes and by the opposite direction of the arrows. The trees represent these sentences: *"To je jasné, že bych byl radši, kdyby tady dosud stál zámek a ne tohle monstrum. Ale proč o tom stále uvažovat?"*

(In English: *It is clear that I would prefer if there still was a castle here and not this monster. But why keep thinking about it forever?"*)

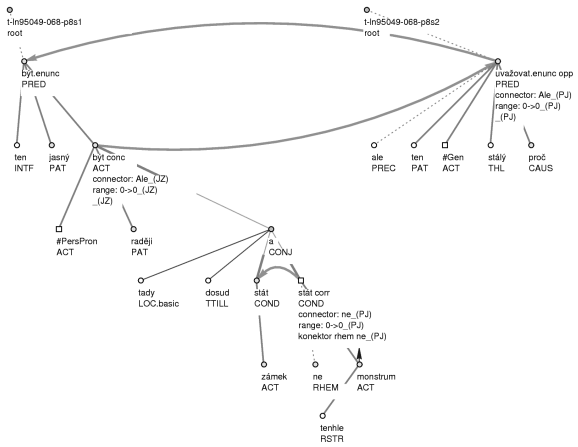


Figure 4. Disagreement in the nodes and in the direction of the arrows.

This time, both annotators recognized a presence of a discourse relation and marked the same connective (*"Ale"*, in English *"But"*). They did not agree on the start/end nodes and on the type of the relation (opposition vs. concession).

Figure 5 shows another type of "slight" disagreement. This time, the annotators agreed on everything but the range of the relation. They agreed both on the type (reason) and the connective (*"tak"*, in English *"Thus"*). The three trees (again a cut of them) represent these three sentences:

"Podle šéfa kanceláře představenstva a. s. Škoda Zdeňka Lavičky jsou však v říjnu schopny fungovat prakticky všechny závody bez vážnějšího omezení. To je v rozporu s tvrzením vedení koncernu z minulého týdne, ve kterém škodovácký management tvrdil, že se odstávka dotkne většiny provozů a závodů Škody Plzeň, která má v současnosti 28000 zaměstnanců. Vzniká tak podezření, že se vedení koncernu snažilo vyvinout tlak na vládu a donutit ji k zaplacení dluhů."

(In English: *"According to Zdeněk Lavička, the chief of the board of directors of Škoda corp., virtually all factories are able to operate in October without serious limitations. It contradicts the statement of the syndicate administration from the last week, in which the management of Škoda claimed that the downtime would affect most of the plants and factories of Škoda Plzeň, which presently has 28,000 employees. Thus a suspicion arises that the syndicate administration tried to exert pressure on the government and force it to pay the debts."*)

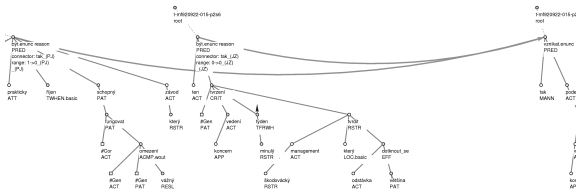


Figure 5. Disagreement in the range of the discourse relation.

The difference between the annotators is in the range of the start part of the arrows. One of the annotators marked the two first sentences as a start point of the relation, while the other marked the second sentence as the start point only. They agreed on the target point of the relation being the third sentence.

Inspired by these examples, we designed another – a connective-based – measure for evaluating the inter-annotator agreement of the discourse relations. It seems that although the annotators sometimes fail to mark the same start/target nodes, or to select the same type or the same range of the relations, they usually agree on the connective. This idea is also supported by high levels of the simple ratio agreement on connectives measured on relations the annotators agreed on from Tables 1 and 2 (0.95 and 0.91). These numbers show that once the annotators agree on a relation, they almost always agree also on the connective.[2]

The connective-based measure considers the annotators to be in agreement on recognizing a discourse relation if they agree on recognizing the same connective (please note that we only annotate discourse relations with explicitly expressed connectives).

Table 3 shows results of the evaluation of the inter-annotator agreement, performed using the connective-based measure, on the same data as Tables 1 and 2.

[2] This is only an interpretation of the numbers, not a description of the annotation process; in fact, the annotators usually first find a connective and then search for the arguments of the discourse relation.

measure	value
F_1-measure on relations	0.86
F_1-measure on relations + types	0.56
F_1-measure on rel. + start/end nodes	0.43
F_1-measure on rel. + types + nodes	0.34
agreement on types	0.65
agreement on start/end nodes	0.50
Cohen's κ on types	0.56

Table 3. The inter-annotator agreement evaluated with the connective-based measure.

This time (compared with Tables 1 and 2, i.e. the simple strict measure and the one-level skipping measure), the agreement (*F_1-measure*) on relations is much higher – 0.86 (vs. 0.43 and 0.54). On the other hand, simple ratio agreement (and Cohen's κ) measured on relations recognized by both annotators are lower than in Tables 1 and 2. Although the annotators might have recognized the same discourse relation, a (possibly small) difference in the interpretation of the text caused sometimes not only a disagreement in the positions of the start/end nodes, but also in the type of the relation.

The simple ratio agreement on types from Table 3 (0.65) is probably the closest measure to the way of measuring the inter-annotator agreement on subtypes in the annotation of discourse relations in the Penn Discourse Treebank 2.0, reported in Prasad et al. (2008). Their agreement was 0.8.

4 Conclusion

We have presented several ways of measuring the inter-annotator agreement in the project of annotating the semantic inter-sentential discourse relations with explicitly expressed connectives in the Prague Dependency Treebank. We have shown examples from parallel annotations that substantiate the importance of the alternative approaches to the evaluation of the agreement.

Skipping a tree level in the start node or the end node of the relations helps to recognize factual agreement in some cases where the strict approach detects disagreement. We have shown that it is still too strict and that there are cases which we would like to classify as agreement but the measure does not recognize them.

The connective-based measure seems to be the closest one to what we would like to consider a criterion of agreement. It disregards the actual nodes that are connected with a discourse relation, and even disregards the direction of the relation. In this sense, it is the most benevolent of the three measures.

It does not mean that the simple strict measure or skipping a tree level are inferior or obsolete ways of measuring the agreement. All the measures focus on different aspects of the agreement and they are all important in the process of annotating the corpus, studying the parallel annotations and improving the annotation instructions. We may agree on the fact that on this level of language description, it is very hard to achieve perfect agreement (Lee at al., 2006), yet we should never cease the effort to further specify and clarify the ways of annotation, in order to catch the same linguistic phenomena in the same way, and thus provide systematic and coherent linguistic data.

Acknowledgments

We gratefully acknowledge support from the Czech Ministry of Education (grant MSM-0021620838), the Grant Agency of the Czech Republic (grants 405/09/0729 and P406/2010/0875), the Czech Science Foundation (grant 201/09/H057), and the Grant Agency of Charles University in Prague (GAUK 103609).

References

Artstein R. and M. Poesio. 2008. *Inter-coder agreement for computational linguistics.* Computational Linguistics 34/4, pp. 555–596.

Asher, N. 1993. *Reference to Abstract Objects in Discourse.* Kluwer Academic Publishers

Cohen, J. 1960. *A coefficient of agreement for nominal scales.* Educational and Psychological Measurement, 20(1), pp. 37–46.

Hajič, J., Panevová, J., Hajičová, E., Sgall, P., Pajas, P., Štěpánek, J., Havelka, J., Mikulová, M., Žabokrtský, Z., and M. Ševčíková-Razímová. 2006. *Prague Dependency Treebank 2.0.* CD-ROM, LDC2006T01, Linguistic Data Consortium, Philadelphia, USA.

Krippendorff, K. 1980. *Content Analysis: An Introduction to Its Methodology*. Chapter 12. Sage, Beverly Hills, CA, USA.

Lee, A., Prasad, R., Joshi, A., Dinesh, N., and B. Weber. 2006. *Complexity of dependencies in discourse: Are dependencies in discourse more complex than in syntax?* J. Hajič and J. Nivre, (eds.). Proceedings of the 5th Workshop on Treebanks and Linguistic Theories (TLT 2006). Prague, Czech Republic, pp. 79–90.

Mikulová, M. et al. 2005: *Annotation on the tectogrammatical layer in the Prague Dependency Treebank. Annotation manual*. Avaiable from http://ufal.mff.cuni.cz/pdt2.0/doc/manuals/en/t-layer/pdf/t-man-en.pdf

Mladová L., Zikánová, Š., Bedřichová, Z., and E. Hajičová. 2009. *Towards a Discourse Corpus of Czech.* Proceedings of the Corpus Linguistics Conference, Liverpool, Great Britain, in press (online proceedings: http://ucrel.lancs.ac.uk/publications/cl2009/).

Passonneau, R. 2004. *Computing Reliability for Coreference.* Proceedings of LREC, vol. 4, Lisbon, Portugal, pp. 1503–1506.

Prasad, R., Dinesh N., Lee A., Miltsakaki, E., Robaldo, L., Joshi, A., Webber, B. 2008. *The Penn Discourse Treebank 2.0.* Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC 2008), Morocco.

Zikánová Š., Mladová L., Mírovský J., and P. Jínová. 2010. *Typical Cases of Annotators' Disagreement in Discourse Annotations in Prague Dependency Treebank.* LREC 2010, Malta, in press.

Opinion Target Extraction in Chinese News Comments

Tengfei Ma **Xiaojun Wan***

Institute of Compute Science and Technology
The MOE Key Laboratory of Computational Linguistics
Peking University
{matengfei, wanxiaojun}@icst.pku.edu.cn

Abstract

News Comments on the web express readers' attitudes or opinions about an event or object in the corresponding news article. And opinion target extraction from news comments is very important for many useful Web applications. However, many sentences in the comments are irregular and informal, and sometimes the opinion targets are implicit. Thus the task is very challenging and it has not been investigated yet. In this paper, we propose a new approach to uniformly extracting explicit and implicit opinion targets from news comments by using Centering Theory. The approach uses global information in news articles as well as contextual information in adjacent sentences of comments. Our experimental results verify the effectiveness of the proposed approach.

1 Introduction

With the dramatic development of web 2.0, there are more and more news web sites allowing users to comment on news events. These comments have become valuable resources for researchers to make advanced opinion analysis, such as tracking the attitudes to a focused event, person or corporation. In these advanced opinion analysis tasks, opinion target extraction is a necessary step. Unfortunately, former works did not focus on the domain of news comments. Though some researchers and workshops have investigated the task of opinion target extraction in product reviews and news articles, the methods cannot perform well on news comments. Actually, target extraction in news comments significantly differs from that in product reviews and news articles in the following ways.

1) Products usually have a set of definite attributes (e.g. size) and related opinion words (e.g. large), and thus researchers can use a small fixed set of keywords to recognize frequent feature words (Zhuang et al., 2006), or leverage the associated rules between feature words and opinion words to improve the performance (Hu and Liu, 2004; Su et al., 2008; Jin and Ho, 2009; Du and Tan, 2009). But news comments are more complicated. There are much more potential opinion targets in news comments. In other words, the candidate targets are in a much more open domain. On the other hand, the opinion targets in news comments are not strongly associated with the opinion words. We cannot judge a target by a special opinion word as easily as in product reviews.

2) The opinionated sentences in news articles mostly contain opinion operators (e.g. believe, realize), which can be used to find the positions of opinion expressions. However, news comments have already been considered to be declared by readers and they do not have many operators to indicate the positions of opinion targets.

3) Furthermore, many comment sentences are of free style. In many cases, there are even no manifest targets in the comment sentences. For example, a news article and its relational comment are as follows:

News: "迪拜将建超千米全世界最高摩天大楼"
(Dubai will build the highest skyscraper in the world)
Comment:
"真的很高，起到什么作用呢？"
(Really high, but what (is it) used for?)

* Contact author

Coling 2008: Poster Volume, pages 782–790,
Beijing, August 2010

The comment sentence obviously comments on "skyscraper" by human understanding, but in the sentence we cannot find the word or an alternative. Instead, the real target is included in the news article. Now we give two definitions of the phenomenon.

Implicit targets: The implicit targets are those opinion targets which do not occur in the current sentence. The sentence is called implicit sentence.

Explicit targets: The explicit targets are those opinion targets which occur in the current right sentence, and the sentence is called explicit sentence.

In Chinese comments, the phenomena of implicit targets are fairly common. In our dataset, the sentences with implicit targets make up nearly 30 percents of the total.

In this paper, we focus on opinion target extraction from news comments and propose a novel framework uniformly extracting explicit and implicit opinion targets. The method uses both information in news articles and information in comment contexts to improve the result. We extract focused concepts in news articles as candidate implicit targets, and exploit a new approach based on Centering Theory to taking advantage of comment contexts.

We evaluate our system on a test corpus containing different topics. The results show that it improves the baseline by 8.8%, and the accuracy is also 8.1% higher over the popular SVM-based method.

The rest of this paper is organized as follows: The next section gives an overview of the related work in opinion analysis. Section 3 introduces the background of Centering Theory and Section 4 describes our framework based on Centering Theory. In Section 5 we test the results and give a discussion on the errors. Finally Section 6 draws a conclusion.

2 Related Work

The early research of opinion mining only focused on the sentiment classification (Turney et al., 2002; Pang et al., 2002). However, for many applications only judging the sentiment orientation is not sufficient (eg. Hu and Liu, 2004). Fine-grained opinion analysis has attracted more and more attention these years. It mainly includes these types: opinion holder extraction (Kim and Hovy, 2005; Choi et al., 2005), opinion target extraction (Kim and Hovy, 2006; Ruppenhofer et al., 2008), and the identification of opinion proposals (Bethard et al., 2004) and some special opinion expressions (Bloom et al., 2007). Also, there are some other related tasks, such as detecting users' needs and wants (Kanayama and Nasukawa, 2008). However, these general systems are different from ours because they do not have or use any contextual information, and implicit opinion targets are not recognized and handled there.

A more special domain of feature extraction is product and movie reviews. Hu and Liu (2004) design a system to mine product features and generate opinion summaries of customer reviews. Frequent features are extracted by a statistical approach, and infrequent features are generated by the associated opinion words. The product features are limited in amount and they are strongly associated with specific opinion words, so researchers can use a fixed set of keywords or templates to extract frequent features (Zhuang et al., 2006; Popescu and Etzioni, 2005) or try various methods to augment the database of product features and improve the extraction accuracy by using the relations between attributes and opinions (Ghani et al., 2006; Su et al., 2008; Jin and Ho, 2009; Du and Tan, 2009). However, in news comments, the opinion targets are not strongly associated with specific opinion words and these techniques cannot be used.

There are also some works focusing on the target extraction in news articles, such as NTCIR7-MOAT (Seki et al., 2008). Different from the news comments, there are opinion indicators in the subjective sentences. However, in our task of this paper, the opinion holders are pre-assigned as the reviewers, so few opinion indicators and holders can be found.

To our best knowledge, this paper is the first work of extracting opinion targets in news comments. We analyze the complex phenomena in news comments and propose a framework to solve the problems of implicit targets. Our method synthesizes the information from related articles and contexts of comments, and it can effectively improve the extracting results.

3 Background of Centering Theory

Centering Theory (Grosz, Joshi and Weinstein, 1995) was developed for an original purpose of indicating the coherence of a discourse and choosing a referring expression. In the theory, the term "***centers***" of an utterance is used to refer to the entities serving to link this utterance to another utterance in a discourse. But this is not the only function of centers, and there are some other useful characteristics of centers to be recognized. Our observation shows that a center always represents the focus of attention, and the salience of a center indicates the significance of the component as a commented target. In news comments, we consider a comment as a discourse and a sentence as an utterance. If an utterance has a "center", then the center can be regarded as the target of the sentence.

Before introducing the common process of choosing the centers in utterances, several definitions are elaborated as follows:

Forward-looking center: Given an utterance U, there is a set of forward-looking centers $C_f(U)$ assigned. The set is a collection of all potential centers that may be realized by the next utterance.

Backward-looking center: Each utterance is assigned exactly one (in fact at most one) backward-looking center C_b. The backward-looking center of utterance U_{n+1} connects with one of the forward-looking centers of U_n. The C_b is the real focus of the utterance.

Rank: The rank is the salience of an element of C_f. Ranking of elements in $C_f(U_n)$ guides determination of $C_b(U_{n+1})$. The more highly ranked an element of $C_f(U_n)$, the more likely it is to be $C_b(U_{n+1})$. The most highly ranked element of $C_f(U_n)$ that is realized in U_{n+1} is the $C_b(U_{n+1})$. The rank is affected by several factors, the most important of which depends on the grammatical role, with SUBJECT > OBJECT(S) > OTHER.

Preferred center: In the set of $C_f(U_n)$, the element with the highest rank is a preferred center $C_p(U_n)$. This means that it has the highest probability to be $C_b(U_{n+1})$.

Table 1 is an example of the centers. In the example, the target of the first sentence is "Jack", which is exactly the preferred center; while in the second sentence, it is easy to see that "him" gets more attention than "the company" in this environment and thus the backward-looking center is more likely to be the target. So we assume that if $C_b(U_n)$ exists, it can be regarded as the opinion target of U_n, otherwise the $C_p(U_n)$ is the target.

Utterance	Center
U_1:杰克是把公司看作他的生命来做的。 (Jack regards the company as his life.)	C_f: 杰克(Jack)/ 公司(the company)/ 生命(life) C_b: null C_p: 杰克(Jack)
U_2: 公司能有今天的成果都是因为他。 (It attributes to him that the company can obtain today's achievement.)	C_f: 公司(the company)/ 成果(achievement)/ 他(杰克) (him(Jack)) C_b:他(杰克) (him(Jack)) C_p:公司(the company)

Table 1 Example of different centers.

4 Proposed Approach

Due to the problems we introduced in Section 1, the techniques of target extraction in other domains are not appropriate in news comments, and general approaches encounter the problems of free style sentences and implicit targets. Fortunately, news comments have their own characteristics, which can be used to improve the target extraction performance.

One important characteristic is that though potential opinion targets may be in large quantities, most comments focus on several central concepts in the corresponding news article, especially in the title. So we can extract the focused concepts in the news and use them as potential implicit targets for the comments.

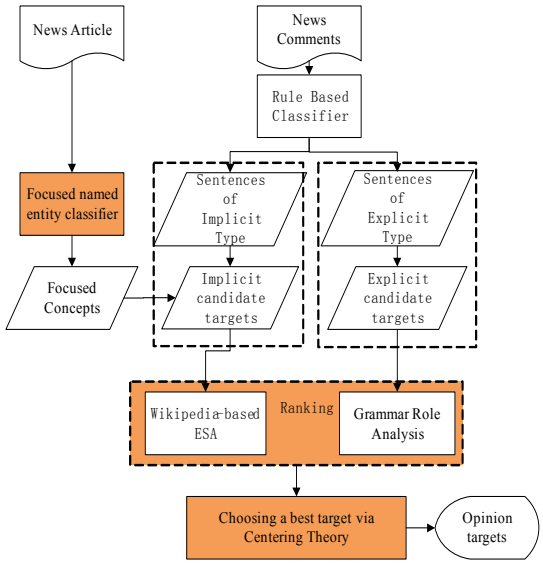


Figure 1: Framework of opinion target extraction in news comments

The other useful information comes from the fact that the sentences in one comment are usually coherent. As the comments may be long and each comment contains several sentences, the sentences within one comment are relevant and coherent. So the opinion targets in previous sentences have some influence on that in subsequent sentences. Using this kind of contextual information, we can eliminate noisy candidates and relax the dependence on an unreliable syntactic parser.

Considering the above characteristics, we propose a framework of target extraction based on focused concepts recognition and Centering Theory, as shown in Figure 1.

Given a news article and its relevant comments, we first adopt some syntactic rules to classify the comment sentences into implicit or explicit type. Whether a sentence includes an explicit target is mainly decided by whether it owns a subject. A few heuristic rules, such as the appearance of the subject, the combination of the POS, and the position of the predicate, are used based on the parse result by using a Chinese NLP toolkit[1], and the rule-based classification can attain an accuracy of 77.33%.

Then we exploit two different approaches for dealing with the two types of sentences, respectively. For the implicit type, we extract the focused concepts in the news article as candidate implicit targets, and rank them by calculating the semantic relatedness between the targets and the sentence. For the explicit type, all nouns and pronouns in the sentence are extracted as candidate targets and ranked mainly by their grammatical roles. At last, Centering Theory is used to choose the best candidate using the ranks and contextual information.

The details of the main parts are explained in the following sections.

4.1 Focused Concepts (FC) Recognition

As the comments usually point to the news article, it is highly probable that the implicit targets appear in the news article. Generally, the focused concepts of the news article are more likely to be the commented targets. Thus, if we extract the focused concepts of the news article, we will get the candidate implicit targets.

In general, the focused concepts are named entities (Zhang et al. 2004) or specific noun phrases. Taking the news

> "迪拜将建超千米全世界最高摩天大楼(*Dubai will build the highest skyscraper in the world*)" ----NEWS1

as an example, "迪拜(*Dubai*)" and "摩天大楼(*skyscraper*)" are the potential opinion targets. "*Dubai*" is a named entity, and "*skyscraper*" is a specific noun phrase. In addition, the focused concepts may also appear in the content of the news article, if they attract enough attention or have strong relations with the focused named entities in the title.

As the number of noun phrases is usually large, if we extract the two types of concepts together, there must be much noise to impact the final result. To be simple and accurate, we first extract focused named entities (FNE), and then expand them with other focused noun phrases, for the reason that the focused noun phrases usually have a strong relation with the focused named entities.

Entity Type	Person, Location, Organization, Time
Title	In title or not
Frequency	The number of occurrence
Relative Frequency	Frequency/the number of total words

[1] LTP, http://ir.hit.edu.cn/demo/ltp/Sharing_Plan.htm
LTP is an integrated NLP toolkit which contains segmentation, parsing, semantic role labeling, and etc.

Distribution Entropy (Here we take N=5 according to the length of articles)	$Entropy = \sum_{i=1}^{N} p_i \log p_i$, where $p_i = \frac{Occurrence\ in\ the\ i^{th}\ Section}{Occurrence\ in\ Total}$

Table 2 Features of FNE classification

Extracting FNEs can be seen as a classification problem. In this work, we choose the features in Table 2.

Given a news document, we first recognize all named entities with our own named entity recognizer (NER).Then all named entities are classified based on the above mentioned features. The noun phrases in the title are also extracted and filtered by their frequency in the news article and co-occurrences with FNEs. The filtering threshold is set to a relatively high value to guarantee that not much noise is brought in. Thus we can get a small set of focused concepts in the news article.

4.2 Ranking Implicit Targets

We use the semantic relatedness to decide which potential target is most likely to be the right implicit target. There are many methods to calculate the semantic relatedness. We choose the Wikipedia-base explicit semantic analysis (ESA) (Gabrilovich and Markovitch, 2007), for its adaptability and effectiveness for Chinese language. The method converts a word or a sentence to a series of wiki concepts, and then calculates the similarity between words or sentences.

Input: a Focused Concept t_0 in the news

Output: a vector C with a length of N. C= $<(c_j,w_j)>$, where c_j is a Wikipedia concept, and w_j is the weight of c_j

1. Find all nouns, adjectives and verbs co-occurring with t_0 in the same sentence, and put them into the set S= $\{t_i\}$.
2. Compute MI (Mutual information) of each t_i with t_0.
3. Choose 10 words in S with the highest MI (according to the total number of words, 10 is a proper value). Combine them with t_0 into a word vector and assign each word t_i a weight of its frequency v_i in the news article. The vector V= $<(t_i,v_i)>$, $|V|\leq 11$.
4. Let $<k_{ij}>$ be an inverted index entry for t_i, where k_{ij} quantifies the strength of association of t_i with Wikipedia concept c_j. Then the vector V can be interpreted as a vector constructed by All Wikipedia concepts. Each concept c_j has a weight $w_j = \sum_{t_i \in V} v_i k_{ij}$.
5. Select N concepts with the highest weights.

Table 3: Algorithm that converts a focused concept to a vector of Wikipedia concepts

Chinese Wikipedia is not as large as English Wikipedia. When some words are not included in the database, the original ESA algorithm will fail. To solve the problem, we first expand the input FC with a few words extracted from the news article. The words represent the semantic information related to the article, so they are more informative than a single concept while easily recognized by the Wikipedia database. The details of the algorithm are shown in Table 3.

On the other hand, when given a comment sentence, we segment it to words and remove the stop words (e.g. "的 (of)"). Then the serial of words are also converted by ESA into a vector of Wikipedia concepts.

After getting the vectors of wiki concepts for focused concepts and the comment sentence, we use the cosine metric to obtain their relatedness scores. In this way, the focused concepts are ranked by their relatedness scores with the sentence.

4.3 Ranking Explicit Targets

A comment sentence with explicit targets usually has a complete syntactic structure. According to Centering Theory, the ranks of explicit targets are decided mainly by their grammatical roles. Generally, a subject is most likely to be the opinion target, and the rank can be heuristically assigned by SUBJECT > OBJECT(S) > OTHER.

4.4 Choosing Best Candidate target via Centering Theory (CT)

After getting the candidate targets and their ranks, we start the matching step to make use of contextual information. The algorithm originates from the process of choosing preferred centers and backward-looking centers. A subtle adaption is that we add some global information in the news article as the context when dealing with the first sentence in a comment. The details of the algorithm are represented in Table 4.

Now we give an example to show the whole process of the framework. The following comment is associated with NEWS1 in Section 4.1.

U1:迪拜现在大力发展旅游和自由贸易。

(Dubai is developing travel and trades.)

U2:是一个很有活力的城市。

((It) is an active city.)

U3:在迪拜你可以感受到很多惊奇。

(In Dubai you can encounter many miracles.)

First, U1, U2 and U3 are classified as explicit, implicit and explicit, respectively. Then for U1 and U3 we choose noun phrases and pronouns in the sentence as candidate targets and rank them according to their grammatical roles. U2 chooses FC as candidates, and "Dubai" is more related than "skyscraper". At last, the final target is chosen by the algorithm in Table 4 and the whole process is illustrated in Table5.

Input: A comment with M sentences $S=\{s_i\}$, each sentence has a candidate target set $C_f(s_i)=\{c_i\}$;
The Focused Concepts set FC in the news article.

Output: A target set $\{t_i\}$, where each t_i is the opinion target of sentence s_i.

1. **For** each s_i in S
2. **If** $i=1$ (s_i is the first sentence)
3. **For** each c_i in $C_f(s_i)$
4. **If** c_i is contained in FC
5. Add c_i into the set $C_b(s_i)$
6. **If** $C_b(s_i)$ is not void
7. Choose the highest ranked element in $C_b(s_i)$ as t_i
8. **Else**
9. Choose the highest ranked element in $C_f(s_i)$ as t_i
10. **Else**
11. **For** each c_i in $C_f(s_i)$
12. **If** c_i realizes (equals or refers to) an element c'_i in $C_f(s_{i-1})$
13. Add c'_i into the set $C_b(s_i)$
14. **If** $C_b(s_i)$ is not void
15. Choose the highest ranked element in $C_b(s_i)$ as t_i
16. **Else**
17. Choose the highest ranked element in $C_f(s_i)$ as t_i

Table 4 Algorithm of choosing the best candidate target via CT

	type	ranks of candidates	target
U1	Explicit	迪拜>旅游>自由贸易 (Dubai >travel >trade)	迪拜 (Dubai)
U2	Implicit	迪拜>摩天大楼 (Dubai>skyscraper)	迪拜 (Dubai)
U3	Explicit	你>惊奇>迪拜 (you>miracles>Dubai)	迪拜 (Dubai)

Table 5 Example of the extraction process

5 Experiments

5.1 Evaluation Setup

To evaluate the whole system, we evaluate not only the result of the final target extraction but also some key steps. This makes the analysis of the bottleneck possible.

We first build a FNE dataset to evaluate the FNE classification result. As our target extraction task focuses on news comments, we collect 1000 news articles and the associated user comments from *http://comment.news.sohu.com,* which is a famous website offering a platform for users to comment on the news. Every news articles are annotated with its focused named entities, which are also the most possible commented targets.

Then we build the target dataset to evaluate the final target extraction. 9 articles and associated comments are randomly chosen from the FNE dataset, and each of their comment sentences is annotated with the opinion target. The target dataset focuses on 3 different topics: economics, technology and sports. Each document contains a news article and about 100 relevant comments, and there are 1597 comment sentences in total.

We assume that each comment sentence has one opinion target, but 108 sentences have more than one focused objects. In that case, we annotate all targets for evaluation and the result is regarded as true if we extract only one of the annotated targets.

In the target dataset, there are 444 sentences with implicit targets. This demonstrates that the implicit target extraction problem is prevalent and worth solving.

For the final target extraction, we use the accuracy metric to evaluate the result. It is defined as follows:

$$\text{Accuracy}=\frac{\text{Number of sentences with right extraction}}{\text{Number of total sentences}}$$

We do not use the precision and recall metric because every comment sentence in our dataset must have a target after extracting. The precision and the recall are both equal to the accuracy.

5.2 Evaluation Results

5.2.1 FNE Results

We perform a 4:1 cross-validation on the FNE dataset using a commonly used classifier SVM-light[2] and gain a mean f-measure of 80.43%.

Then, to assess the improvement by the FNE step and the classification of implicit and explicit sentences, we estimate the theoretic upper limit of the following three target extractions on the target dataset. **Test 1** assumes every noun phrases or nouns in the sentence can be possible to be extracted as the target. So if there is one candidate matching the target, we can recognize the sentence as extractable. **Test 2** adopts the annotation results of the classification of explicit and implicit sentences. For the manually annotated implicit targets, we adapt the candidate to be FC. Then, as same as Test 1, all candidates are determined whether to be the target. In **Test 3**, we follow the ruled-based classification of implicit and explicit sentences in our system and then judge the sentences whether extractable or not.

	Proportion of extractable sentences
Test 1	55.0%
Test 2	69.6%
Test 3	61.7%

Table 6 Improvement of the proportion of extractable sentences by FNE classification and explicit/implicit sentence classification

Table 6 shows the proportions of extractable sentences in the three tests. It is easy to see that the proportion of extractable sentences means the theoretic optimization of target extraction. So, by Test 2 we can see the extracted FC set is an effective complement of the candidate targets, while Test 3 demonstrates that the system still has much potential to improve the baseline after the rule-based classification of explicit and implicit sentences.

5.2.2 Target Extraction Results

To demonstrate the effectiveness of our approach, we design two baselines.

Baseline 1 treats all sentences as explicit type. In the method, we extract all noun phrases and pronouns in a sentence as candidates and obtain their ranks according to their grammatical roles.

Baseline 2, a SVM-based approach, is offered to compare with the popular target extraction methods. In this method we regard the target extraction as a classification problem. We extract the candidate noun phrases in a sentence first, and then use the semantic features to classify them as targets or not. The features mainly include: POS, whether or not a Named Entity, the positions in the sentence, the syntactic relations with the verb, and etc. As it is a supervised approach, the result is tested by a 2:1 cross validation.

Then we use a method called **FC-only (using only Focused Concepts)** to improve Baseline 1 by using the global information in news articles. For sentences of explicit type, we use the method in Baseline1. For sentences of implicit type, we take focused concepts in news articles as potential targets, and choose the highest ranked element as the final target.

Finally, our proposed approach **CT (using Centering Theory)** uses both Focused Concepts and Centering Theory. When the size of Wikipedia concept vector is set to be 800, the comparison results of the four approaches are shown in Table 7:

	Accuracy
Baseline1	34.38%
Baseline2(SVM-based)	35.13%
FC-only	37.25%
CT	**43.20%**

Table 7 Comparison results

FC-only is better than Baseline1, which demonstrates that the focused concepts are useful to provide information to implicit targets extraction. 444 implicit sentences are a large proportion of the total corpus. And the focused concepts do represent the global information and have influence on the target extraction.

Centering Theory is naturally another improvement. It mainly takes advantage of the information of contexts within a comment, using a rule of coherence to decide the center of attention. And the result indicates that it is very helpful.

Compared with the SVM-based approach, our approach is also much better. The SVM-based approach is only a little higher than Baseline 1. It seems that the manually annotated information is not very useful in target extraction in news comments. The reason may be that the target rules are complicated and exist not only in the current sentence. Using global and contextual

[2] http://svmlight.joachims.org/

information is a more economic and effective way to improve the result.

In the Wikipedia-based ESA algorithm,there is a parameter of *N*, which is the vector size of the expanded vector. It is important to choose a proper parameter value to achieve a high accuracy and meanwhile keep a low computational complexity. The accuracy curves for FC and CT with different values of *N* are represented in Figure 2. Apparently, when *N* exceeds 600, the extraction performance almost does not change any more. So we finally take 800 as the value of *N* .

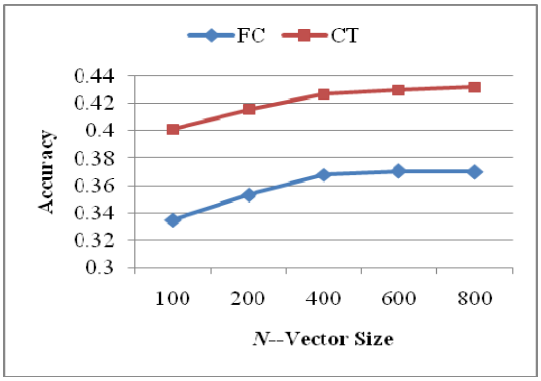


Figure 2: Accuracy vs. vector size *N*

5.3 Error Analysis

Generally there are two major types of errors in the extraction results. One common error is that the target is not in our extracted candidate nouns or noun phrases. For example:

> "可口买汇源，可真是中国饮料的灾难了." (It is a disaster of Chinese beverage that Coca Cola buys HuiYuan.)

The sentence comments on the event of "Coca Cola buys HuiYuan" but not a single concept "Coca Cola" or "HuiYuan". But our system cannot recognize this type of targets properly. Also there are some cases that the noun phrases missed to be extracted by the LTP toolkit. It causes that the target is not matched by the candidates.

Another error originates from the wrong classification of explicit and implicit sentences. For example,

> "还利于民才能化解中小企业生存危机." (Returning profits to civilians can get through the crisis of little companies.)

In this sentence, "还利于民(Returning profits to civilians)" is the opinion target and the sentence has a explicit target. But the rules based on the Chinese parser failed to recognize the phrase as a subject and thus the sentence is considered as implicit type by our approach. And lastly the target is extracted incorrectly.

In 5.2.1, we test the theoretic upper limit of the target extraction and prove the potential effectiveness of two steps. The tests also can be used to estimate the proportion of the types of errors and analyze the bottleneck. In Test 2, there are 298 un-extractable sentences among the annotated explicit sentences. It shows that there is at least 18.6% loss in accuracy caused by the candidate recognition, which accounts for the first error type. As for the second error type, its proportion can be computed by the reduction from Test 2 to Test 3, which is 7.9%.

6 Conclusion and Future Work

In this paper, we propose a novel approach to extracting opinion targets in Chinese news comments. In order to solve the problem of implicit target extraction, we extract focused concepts and rank their importance by computing the semantic relatedness with sentences via Wikipedia. In addition, we apply Centering Theory to the target extraction system, for utilizing contextual information. The experiment results demonstrate that our approach is effective.

Currently, the result does not reach an absolutely high accuracy. One bottleneck is that Chinese parsing results are far from satisfactory. Actually this bottleneck has impacted the general target extraction long, such as the low performances of all participants in the target extraction task of NTCIR7-MOAT-CS. We hope to improve our results by avoid this disadvantage. Moreover, the phenomenon of implicit opinion targets exists not only in Chinese but also in English and other languages, while sometimes it is similar to zero anaphora. So the approach in this paper can be extended to news comments in other languages.

Acknowledgement

This work was supported by NSFC (60873155), Beijing Nova Program (2008B03), NCET (NCET-08-0006) and National High-tech R&D Program (2008AA01Z421). We thank the anonymous reviewers for their useful comments.

References

Bethard, Steven, Hong Yu, Ashley Thornton, Vasileios Hatzivassiloglou, and Dan Jurafsky. 2004. *Automatic Extraction of Opinion Propositions and their Holders*. In Proceedings of AAAI Spring Symposium on Exploring Attitude and Affect in Text: Theories and Applications.

Choi, Yejin, Claire Cardie, Ellen Riloff, and Siddharth Patwardhan. 2005. *Identifying Sources of Opinions with Conditional Random Fields and Extraction Patterns*. In Proceeding of HLT/EMNLP' 05.

Ding Xiaowen, Bing Liu, Philip S. Yu. 2008. *A Holistic Lexicon based Approach to Opinion Mining*. Proceeding of the international conference on Web Search and Web Data Mining (WSDM'08), 231-239.

Du, Weifu. and Songbo Tan. 2009. *An Iterative Reinforcement Approach for Fine-Grained Opinion Mining*. The 2009 Annual Conference of the North American Chapter of the ACL

Gabrilovich, Evgeniy. and Shaul Markovitch. 2007. *Computing Semantic Relatedness using Wikipedia-based Explicit Semantic Analysis*. In Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI).

Ghani, Rayid, Katharina Probst, Yan Liu, Marko Krema, and Andrew Fano. 2006. *Text Mining for Product Attribute Extraction*. The Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.

Grosz, Barbara J., Scott Winstein, and Aravind K. Joshi. (1995). *Centering: A Framework for Modeling the Local Coherence of Discourse*. In Computational Linguistics, 21(2).

Hu, Minqing and Bing Liu. 2004. *Mining Opinion Features in Customer Reviews*. In Proceedings of Nineteenth National Conference on Artificial Intelligence (AAAI-2004)

Jin, Wei and Hung Hay Ho. 2009. *A Novel Lexicalized HMM-based Learning Framework for Web Opinion Mining*. In Proceedings of the 26th International Conference on Machine Learning (ICML 2009).

Jin, Wei and Hung Hay Ho, Rohini K. Srihari. 2009. *OpinionMiner: A Novel Machine Learning System for Web Opinion Mining and Extraction*. In The 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.

Kim, Soo-Min. and Eduard Hovy. 2006. *Extracting Opinions, Opinion Holders, and Topics Expressed in Online News Media Text*. In ACL Workshop on Sentiment and Subjectivity in Text.

Kim, Soo-Min. and Eduard Hovy. 2005. *Identifying Opinion Holders for Question Answering in Opinion Texts*. In Proceedings of AAAI-05 Workshop on Question Answering in Restricted Domains.

Pang, Bo and Lillian Lee, and Vaithyanathan, S. 2002. *Thumbs up?* Sentiment *classification using machine learning techniques.* In EMNLP 2002.

Popescu, Ana-Maria. and Oren Etzioni. 2005. *Extracting Product Features and Opinions from Reviews*. In Proceeding of 2005 Conference on Empirical Methods in Natural Language Processing (EMNLP'05), 339-346.

Riloff, Ellen and Janyce Wiebe. 2003. *Learning Extraction Patterns for Subjective Expressions*. Proceedings of the 2003 Conference on EMNLP.

Ruppenhofer, Josef, Swapna Somasundaran, and Janyce Wiebe. 2008. *Finding the Sources and Targets of Subjective Expressions*. In LREC08.

Seki, Yohei, David K. Evans, Lun-Wei Ku, Le Sun, Hsin-Hsi Chen, and Noriko Kando. 2008. *Overview of Multilingual Opinion Analysis Task at NTCIR-7*. The 7th NTCIR workshop (2007/2008).

Su Qi, Xinying Xu, Honglei Guo, Zhili Guo, XianWu, Xiaoxun Zhang, Bin Swen and Zhong Su. 2008. *Hidden Sentiment Association in Chinese WebOpinion Mining*. In The 17th International World Wide Web Conference (WWW).

Turney, Peter D. 2002. *Thumbs up or thumbs down? Semantic* orientation *applied to unsupervised classification of reviews.* In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL).

Zhang, Li, Yue Pan, and Tong Zhang. 2004. *Focused Named Entity Recognition using Machine Learning*. The 27th Annual International ACM SIGIR Conference.

Zhuang, Li, Feng Jing. and Xiao-yan Zhu. 2006. *Movie Review Mining and Summarization*. In Proceedings of the 15th ACM International Conference on Information and Knowledge Management (CIKM'06), 43-50.

Finite-state Scriptural Translation

M. G. Abbas Malik **Christian Boitet** **Pushpak Bhattacharyya**
GETALP – LIG (Grenoble Informatics Lab.)
University of Grenoble IIT Bombay
Abbas.Malik Christian.Boitet@imag.fr pb@iitb.ac.in

Abstract

We use robust and fast Finite-State Machines (FSMs) to solve scriptural translation problems. We describe a *phonetico-morphotactic pivot* UIT (universal intermediate transcription), based on the common phonetic repository of Indo-Pak languages. It is also extendable to other language groups. We describe a *finite-state scriptural translation model* based on finite-state transducers and UIT. We report its performance on Hindi, Urdu, Punjabi and Seraiki corpora. For evaluation, we design two classification scales based on the word and sentence accuracies for translation system classifications. We also show that subjective evaluations are vital for real life usage of a translation system in addition to objective evaluations.

1 Introduction

Transliteration refers to phonetic translation across two languages with different writing systems, such as Arabic to English (Arbabi *et al.*, 1994; Stall and Knight, 1998; Al-Onaizan and Knight, 2002; AbdulJaleel and Larkey, 2003). Most prior work on transliteration has been done for MT of English, Arabic, Japanese, Chinese, Korean, etc., for CLIR (Lee and Choi., 1998; Jeong *et al.*, 1999; Fujii and Ishikawa, 2001; Sakai *et al.*, 2002; Pirkola *et al.*, 2003; Virga and Khudanpur, 2003; Yan *et al.*, 2003), and for the development of multilingual resources (Kang and Choi, 2000; Yan, Gregory et al., 2003).

The terms transliteration and transcription are often used as generic terms for various processes like transliteration, transcription, romanization, transcribing and technography (Halpern, 2002). In general, the speech processing community uses the term transcription to denote a process of conversion from the script or writing system to the sound (phonetic representation). For example, the transcription of the word "love" in the International Phonetic Alphabet (IPA) is [ləv]. While the text processing community uses the term transliteration and defines it as a process of converting a word written in one writing system into another writing system while preserving the sound of the original word (Al-Onaizan and Knight, 2002; AbdulJaleel and Larkey, 2003). More precisely, the text processing community defines the term transliteration as two transcription processes "source script to sound transcription" and "sound to target script transcription" and sometimes as one transcription process "source script to target script transcription".

We propose a new term *Scriptural Translation* for this combined process. Scriptural translation is a process of transcribing a word written in the source language script into the target language script by preserving its articulation in the original language in such a way that the native speaker of the target language can produce the original pronunciation.

FSMs have been successfully used in various domains of Computational Linguistics and Natural Language Processing (NLP). The successful use of FSMs have already been shown in various fields of computational linguistics (Mohri, 1997; Roche and Schabes, 1997; Knight and Al-Onaizan, 1998). Their practical and advantageous features make them very strong candidates to be used for solving *scriptural translation problems*.

First, we describe scriptural translation and identify its problems that fall under weak translation problems. Then, we analyze various challenges for solving weak scriptural translation problems. We describe our finite-state scriptural translation model and report our results on Indo-Pak languages.

Coling 2008: Poster Volume, pages 791–800,
Beijing, August 2010

2 Scriptural Translation – a weak translation problem

A weak translation problem is a translation problem in which the number of possible valid translations, say *N*, is either very small, less than 5, or almost always 1.

Scriptural Translation is a sub-problem of general translation and almost always a *weak translation problem*. For example, French-IPA and Hindi-Urdu scriptural translation problems are weak translation problems due to their small number of valid translations. On the other hand, Japanese-English and French-Chinese scriptural translation problems are not weak.

Scriptural translation is not only vital for translation between different languages, but also becomes inevitable when the same language is written in two or more mutually incomprehensible scripts. For example, Punjabi is written in three different scripts: Shahmukhi (a derivation of the Perso-Arabic script), Gurmukhi and Devanagari. Kazakh and Kurdish are also written in three different scripts, Arabic, Latin and Cyrillic. Malay has two writing systems, Latin and Jawi (a derivation of the Arabic script), *etc.* Figure 1 shows an example of scriptural divide between Hindi and Urdu.

دنیا کو امن کی ضرورت ہے۔

दुनिया को अमन की ज़रूरत है।

[d̪ʊnɪjɑ ko əmən ki zərurət̪ hæ.]

The world needs peace.

Figure 1: Example of scriptural divide

Thus, solving the scriptural translation problem is vital to bridge the scriptural divide between the speakers of different languages as well as of the same language.

Punjabi, Sindhi, Seraiki and Kashmiri exist on both sides of the common border between India and Pakistan and all of them are written in two or more mutually incomprehensible scripts. The Hindi–Urdu pair exists both in India and Pakistan. We call all these languages the *Indo-Pak* languages.

3 Challenges of Scriptural Translation

In this section, we describe the main challenges of scriptural translation.

3.1 Scriptural divide

There exists a written communication gap between people who can understand each other verbally but cannot read each other. They are virtually divided and become *scriptural aliens*. Examples are the Hindi & Urdu communities, the Punjabi/Shahmukhi & Punjabi/Gurmukhi communities, *etc.* An example of scriptural divide is shown in Figure 1. Such a gap also appears when people want to read some foreign language or access a bilingual dictionary and are not familiar with the writing system. For example, Japanese–French or French–Urdu dictionaries are useless for French learners because of the scriptural divide. Table 1 gives some figures on how this scriptural divide affects a large population of the world.

Sr.	Language	Number of Speakers
1	Hindi	853,000,000
2	Urdu	164,290,000
3	Punjabi	120,000,000
4	Sindhi	21,382,120
5	Seraiki	13,820,000
6	Kashmir	5,640,940
Total		**1178,133,060**

Table 1: Number of Speakers of Indo-Pak languages

3.2 Under-resourced languages

Under-resourced and under-written features of the source or target language are the second big challenge for *scriptural translation*. The lack of standard writing practices or even the absence of a standard code page for a language makes transliteration or transcription very hard. The existence of various writing styles and systems for a language leads towards a large number of variants and it becomes difficult and complex to handle them.

In the case of Indo-Pak languages, Punjabi is the largest language of Pakistan (more than 70 million) and is more a spoken language than a written one. There existed only two magazines (one weekly and one monthly) in 1992 (Rahman, 1997). In the words of (Rahman, 2004), "… *there is little development in Punjabi, Pashto, Balochi and other languages*…". (Malik, 2005) reports the first effort towards establishing a standard code page for Punjabi-Shahmukhi and till date, a standard code page for Shahmukhi does not exist. Similar problems also exist for the Kashmiri and Seraiki languages.

3.3 Absence of necessary information

There are cases where the necessary and indispensable information for scriptural translation are missing in the source text. For example, the first word دنیا [d̪ʊnɪjɑ] (world) of the example sentence of Figure 1 misses crucial diacritical information, mandatory to perform Urdu to Hindi scriptural translation. Like in Arabic, diacritical marks are part of the Urdu writing system but are sparingly used in writings (Zia, 1999; Malik *et al.*, 2008; Malik *et al.*, 2009).

Figure 2(a) shows the example word without diacritical marks and its wrong Hindi conversion according to conversion rules (explained later). The Urdu community can understand the word in its context or without the context because people are tuned to understand the Urdu text or word without diacritical marks, but the Hindi conversion of Figure 2(a) is not at all acceptable or readable in the Hindi community.

Figure 2(b) shows the example word with diacritical marks and its correct Hindi conversion according to conversion rules. Similar problems also arise for the other Indo-Pak languages. Therefore, missing information in the source text makes the scriptural translation problem computationally complex and difficult.

[ɑ] ا [j] ی [ɪ] ِ [n] ن [ʊ] ُ [d̪] د = دُ نِیا
दुनिया = द [d̪] ु [ʊ] न [n] ि [ɪ] य [j] ा[ɑ]
(b) with necessary information
[ɑ] ا [j] ی [n] ن [d̪] د = دنیا
दनया = द [d̪] न [n] य [j] ा[ɑ]
(a) without necessary information

Figure 2: Example of missing information

3.4 Different spelling conventions

Different spelling conventions exist across different scripts used for the same language or for different languages because users of a script are tuned to write certain words in a traditional way. For example, the words یہ [je] (this) = ی [j] + ہ [h] and وہ [vo] (that) = و [v] + ہ [h] are used in Urdu and Punjabi/Shahmukhi. The character ہ [h] produces the vowel sounds [e] and [o] in the example words respectively. On the other hand, the example words are written as ये [je] & वो [vo] and ਯੇ [je] & ਵੋ [vo] in Devanagari and Gurmukhi, respectively. There exist a large number of such conventions between Punjabi/Shahmukhi–Punjabi Gurmukhi, Hindi–Urdu, *etc.*

Different spelling conventions are also driven by different religious influences on different communities. In the Indian sub-continent, Hindi is a part of the Hindu identity, while Urdu is a part of the Muslim identity[1] (Rahman, 1997; Rai, 2000). Hindi derives its vocabulary from Sanskrit, while Urdu borrows its literary and scientific vocabulary from Persian and Arabic. Hindi and Urdu not only borrow from Sanskrit and Persian/Arabic, but also adopt the original spellings of the borrowed word due the sacredness of the original language. These differences make scriptural translation across scripts, dialects or languages more challenging and complex.

3.5 Transcriptional ambiguities

Character level scriptural translation across different scripts is ambiguous. For example, the Sindhi word اِنسان [ɪnsɑn] (human being) can be converted into Devanagari either as इंसान [ɪnsɑn] or इसान* [ɪsɑn] (* means wrong spellings). The transliteration process of the example word from Sindhi to Devanagari is shown in Figure 3(a). The transliteration of the third character from the left, Noon (ن) [n], is ambiguous because in the middle of a word, Noon may represent a consonant [n] or the nasalization [ɲ] of a vowel.

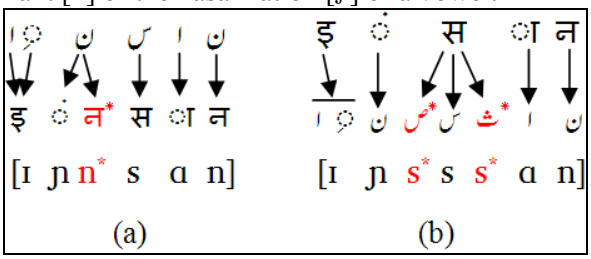


Figure 3: Sindhi transliteration example

In the reverse direction, the Sindhi Devanagari word इंसान [ɪnsɑn] can be converted into a set of possible transliterations [اِنسان, اِنصان*, اِنثان*]. All these possible transliterations have the same pronunciation [ɪnsɑn] but have different spellings in

[1] The Hindi movement of the late 19th century played a central role in the ideologization of Hindi. The movement started in reaction to the British Act 29 of 1837 by which Persian was replaced by Hindustani/Urdu, written in Persian script, as the official vernacular of the courts of law in North India. It is the moment in history, when Hindi and Urdu started to emerge as Hindu and Muslim identities.

the Perso-Arabic script, as shown in Figure 3(b). Similar kinds of ambiguities also arise for other pairs of scripts, dialects or languages. Thus these ambiguities increase the complexity and difficulty of *scriptural translation*.

3.6 Distinctive sound inventories

Sound inventories across dialects or languages can be different. Consider the English–Japanese pair. Japanese make no distinction between the 'L' [l] and 'R' [r] sounds so that these two English sounds collapse onto the same Japanese sound (Knight and Al-Onaizan, 1998).

For Indo-Pak languages, Punjabi/Gurmukhi (a dialect of Punjabi spoken in India) possesses two additional sounds than Punjabi/Shahmukhi (a dialect of Punjabi spoken in Pakistan). Similarly, Hindi, Punjabi, Sindhi and Seraiki have the retroflex form [ɳ], but Urdu and Kashmiri do not. Marathi has 14 vowels in contrast to Hindi's 11 vowels, shown in Table 2.

Hindi Vowels
अ [ə] आ [ɑ] इ [ɪ] ई [i] उ [ʊ] ऊ [u] ऋ [r̥] ए [e] ऐ [æ] ओ [o] औ [ɔ]
Marathi Vowels
अ [ə] आ [ɑ] इ [ɪ] ई [i] उ [ʊ] ऊ [u] ऋ [r̥] ए [e] ऐ [æ] ओ [o] औ [ɔ] अं [əŋ] अः [əh] ऌ [l̥]

Table 2: Hindi and Marathi vowel comparison

Scriptural translation approximates the pronunciation of the source language or dialect in the target due to different sound inventories. Thus a distinctive sound inventory across scripts, dialects or languages increases ambiguities and adds to the complexity of the *scriptural translation* problem.

4 Universal Intermediate Transcription

UIT (Universal Intermediate Transcription) is a multipurpose pivot. In the current study, it is used as a *phonetico-morphotactic* pivot for the *surface morphotactic translation* or scriptural translation.

Although we have not used IPA as encoding scheme, we have used the IPA coding associated with each character as the encoding principle for our ASCII encoding scheme. We selected the printable ASCII characters to base the UIT encoding scheme because it is universally portable to all computer systems and operating systems without any problem (Boitet and Tchéou, 1990; Hieronymus, 1993; Wells, 1995). UIT is a deterministic and unambiguous scheme of transcription for Indo-Pak languages in ASCII range 32–126, since a text in this rage is portable across computers and operating systems (Hieronymus, 1993; Wells, 1995).

Speech Assessment Methods Phonetic Alphabet (SAMPA)[2] is a widely accepted scheme for encoding IPA into ASCII. The purpose of SAMPA was to form the basis of an international standard machine-readable phonetic alphabet for the purpose of international collaboration in speech research (Wells, 1995). The UIT encoding of Indo-Pak languages is developed as an extension of the SAMPA and X-SAMPA that covers all symbols on the IPA chart (Wells, 1995).

4.1 UIT encodings

All characters of the Indo-Pak languages are subdivided into three categories, consonants, vowels and other symbols (punctuations and digits).

Consonants are further divided into aspirated consonants and non-aspirated consonants. For aspiration, in phonetic transcription a simple 'h' following the base consonant symbol is considered adequate (Wells, 1995). In the Indo-Pak languages, we have two characters with IPA [h]. Thus to distinguish between the 'h' consonants and the aspiration, we use *underscore* '_' to mark the aspirate and we encode an aspiration as '_h'. For example, the aspirated consonants ٹھ [t^h], پھ [p^h] and ڇھ [$ʧ^h$] of the Indo-Pak languages are encoded as 't`_h', 'p_h' and 't_S_h' respectively. Similarly for the dental consonants, we use the '_d' marker. For example, the characters د [d̪] and ت [t̪] are encoded as 'd_d' and 't_d' in UIT. Table 3 shows the UIT encodings of Hindi and Urdu aspirated consonants.

Hindi	Urdu	UIT	Hindi	Urdu	UIT
भ	بھ [b^h]	b_h	ह्र	رھ [r^h]	r_h
फ	پھ [p^h]	p_h	ढ़	ڑھ [$ɽ^h$]	r`_h
थ	تھ [$t̪^h$]	t_d_h	ख	کھ [k^h]	k_h
ठ	ٹھ [$ʈ^h$]	t`_h	घ	گھ [g^h]	g_h
झ	جھ [$ʤ^h$]	d_Z_h	ल्ह	لھ [l^h]	l_h
छ	چھ [$ʧ^h$]	t_S_h	म्ह	مھ [m^h]	m_h

[2] http://www.phon.ucl.ac.uk/home/sampa/

ध	دھ [d̪ʰ]	d_d_h	न्ह	نھ [nʰ]	n_h
ढ	ڈھ [ɖʰ]	d`_h			

Table 3: UIT encodings of Urdu aspirated consonants

Similarly, we can encode all characters of Indo-Pak languages. Table 4 gives UIT encodings of Hindi and Urdu non-aspirated consonants. We cannot give all encoding tables here due to shortage of space.

Hindi	Urdu	UIT	Hindi	Urdu	UIT
ब	ب [b]	b	स	ص [s]	s2
प	پ [p]	p	ज़	ض [z]	z2
त	ت [t̪]	t_d	त	ط [t̪]	t_d1
ट	ٹ [ʈ]	t`	ज़	ظ [z]	z3
स	ٹ [s]	s1	-	ع [ʔ]	?
ज	ج [ʤ]	d_Z	ग़	غ [ɣ]	X
च	چ [ʧ]	t_S	फ़	ف [f]	f
ह	ح [h]	h1	क़	ق [q]	q
ख़	خ [x]	x	क	ک [k]	k
द	د [d̪]	d_d	ग	گ [g]	g
ड	ڈ [ɖ]	d`	ल	ل [l]	l
ज़	ذ [z]	z1	म	م [m]	m
र	ر [r]	r	न	ن [n]	n
ड़	ڑ [ɽ]	r`	व	و [v]	v
ज़	ز [z]	z	ह	ہ [h]	h
ज़	ژ [ʒ]	Z	य	ی [j]	j
स	س [s]	s	त	ۃ [t̪]	t_d2
श	ش [ʃ]	S	ण	- [ɳ]	n`
ष	ش [ʃ]	S1	ं	ں [ŋ]	~

Table 4: UIT encodings of Urdu non-aspirated consonants

5 Finite-state Scriptural Translation Model

Figure 4 shows the system architecture of our finite-state scriptural translation system.

`Text Tokenizer` receives and converts the input source language text into constituent words or tokens. This list of the source language tokens is then passed to the `UIT Encoder` that encodes these tokens into a list of UIT tokens using the source language to UIT conversion transducer from the repertoire of *Finite-State Transducers*. These UIT tokens are given to the `UIT Decoder` that decodes them into target language tokens using the UIT to target language conversion transducer from the repertoire of Transducers. Finally, `Text Generator` generates the target language text from the translated target language tokens.

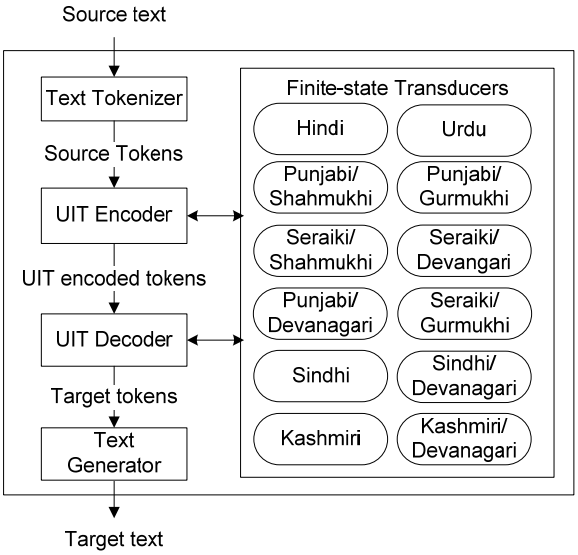


Figure 4: System Architecture of fintie-state scriptural translation

5.1 Finite-state Transducers

Both conversions of the source language text into the UIT encoded text and from the UIT encoded text into the target language text are regular relations on strings. Moreover, regular relations are closed under serial composition and a finite set of conversion relations when applied to each other's output in a specific order, also defines a regular expression (Kaplan and Kay, 1994). Thus we model the conversions from the source language to UIT and from UIT to the target language as finite-state transducers. These translational transducers can be deterministic and non-deterministic.

Character Mappings: Table 5 shows regular relations for converting Hindi and Urdu aspirated consonants into UIT.

IPA	Hindi to UIT	Urdu to UIT
b^h	भ → b_h	ﮪ → b_h
p^h	फ → p_h	ﭫ → p_h
$\underset{\frown}{t}^h$	थ → t_d_h	ﺗ → t_d_h
$ʈ^h$	ठ → t`_h	ﭨ → t`_h
$ʤ^h$	झ → d_Z_h	ﺟ → d_Z_h
$ʧ^h$	छ → t_S_h	ﭽ → t_S_h

$d̪^h$	ध → d_d_h	دھ → d_d_h
$ɖ^h$	ढ → d`_h	ڈھ → d`_h
r^h	ह़ → r_h	رھ → r_h
$ɽ^h$	ढ़ → r`_h	ڑھ → r`_h
k^h	ख → k_h	کھ → k_h
g^h	घ → g_h	گھ → g_h
l^h	ल्ह → l_h	لھ → l_h
m^h	म्ह → m_h	مھ → m_h
n^h	न्ह → n_h	نھ → n_h

Table 5: Regular rules for aspirated consonants of Hindi and Urdu

By interchanging the UIT encodings before the arrow sign and the respective characters of Hindi and Urdu after the arrow, we can construct regular conversion relations from UIT to Hindi and Urdu. We have used XFST (Xerox finite-state engine) to build finite-state transducers. Table 6 shows a sample XFST code.

Contextual Mappings: A contextual mapping is a contextual rule that determines a desired output when a character appears in a certain context. The third command of Table 6 models another contextual mapping saying that 'ह्' is translated by '_h' when it is preceded by any of the characters र, ल, म, and न. The second last rule of Table 6 models the contextual mapping rule that 'A1' is translated into 'ہ' when it is at the end of a word and preceded by a consonant.

```
clear stack
set char-encoding UTF-8
read regex [ि -> I];
read regex [ख -> [k "_" h], घ -> [g
  "_" h], छ -> [t "_" S "_" h], झ -
  > [d "_" Z "_" h], ठ -> [t "`" "_"
  h], ढ -> [d "`" "_" h], थ -> [t
  "_" d "_" h], ध -> [d "_" d "_"
  h], फ -> [p "_" h], भ -> [b "_"
  h], ढ़ -> [r "`" "_" h], स -> s, त
  -> [t "_" d], र -> r, ल -> l, म ->
  m, न -> n, व -> v, ह -> h];
read regex [[् ह] -> ["_" h] || [र |
ल | म | न] _ ];
compose net
```

Table 6: Sample XFST code

Vowel representations in Urdu, Punjabi/Shahmukhi, Sindhi, Seraiki/Shahmukhi and Kashmiri are highly context-sensitive (Malik *et al.*, 2010).

6 Experiments and Results

A sample run of our finite-state scriptural translation system on the Hindi to Urdu example sentence of Figure 1 is shown in Table 7.

Text Tokenizer	UIT Encoder	UIT Decoder	
		Unique output	*Ambiguous outputs*
दुनिया	dUnIjA1	دُنِیا	[دُنِیہ , دُنیا]
को	ko	کو	[قو , کو]
अमन	@mn	امن	[امن]
की	ki	کی	[قی , کی]
ज़रूरत	zrurt_d	زرُوت	[زرُرت , ضرُورت , ذرُورت , ظرُورت , ژرُورت , …]
है	h{	ہَے	[حَے , ہے]

Table 7: Sample run of finite-state scriptural translation model on Hindi to Urdu example

`Text Generator` converts the unique output of the UIT Decoder into an Urdu sentence with one error in the fifth word (highlighted), shown in Figure 5.

دنِیا کو امن کی زرورت ہے

Figure 5: Unique output of the sample run by deterministic FSTs

On the other hand, from the ambiguous output of the UIT Decoder, we can generate 240 output sentences, but only one is the correct scriptural translation of the source Hindi sentence in Urdu. The correct sentence is shown in Figure 6. The sole difference between the output of the deterministic FST and the correct scriptural translation is highlighted in both sentences shown in Figure 5 and 6.

دنِیا کو امن کی ضرورت ہے

Figure 6: Correct scriptural translation of the example

6.1 Test Data

Table 8 shows test sets for the evaluation of our finite-state scriptural translation system.

Data set	Language pair	No. of words	No. of sentences	Source
HU 1	Hindi–Urdu	52,753	-	Platts dictionary
HU 2	Hindi–Urdu	4,281	200	Hindi corpus
HU 3	Hindi–Urdu	4,632	226	Urdu corpus
PU	Punjabi/Shahmukhi–Punjabi/Gurmukhi	5,069	500	Classical poetry
SE	Seraiki/Shahmukhi–Seraiki/Devanagari	2,087	509	Seraiki poetry

Table 8: Test Sets of Hindi, Urdu, Punjabi and Seraiki

HU 1 is a word list obtained from the Platts dictionary[3] (Platts, 1884).

6.2 Results

For Hindi to Urdu scriptural translation, we have applied the finite-state model to all Hindi inputs of HU Test sets 1, 2 and 3. In general, it gives us an Urdu output with the necessary diacritical marks. To evaluate the performance of Hindi to Urdu scriptural translation of our finite-state system against the Urdu without diacritics, we have created a second Urdu output by removing all diacritical marks from the default Urdu output of the finite-state system. We have calculated the *Word Accuracy Rate* (WAR) and *Sentence Accuracy Rate* (SAR) for the default and the processed Urdu outputs by comparing them with the Urdu references with and without diacritics respectively. To compute WAR and SAR, we have used the SCLITE utility from the Speech Recognition Scoring Toolkit (SCTK)[4] of NIST. The results of Hindi to Urdu scriptural translation are given in Table 24.

Test Set	Default output		Processed output	
	Word Level	Sentence Level	Word Level	Sentence Level
HU 1	32.5%	-	78.9%	-
HU 2	90.8%	26.5%	91.0%	27%
HU 3	81.2%	8.8%	82.8%	9.7%

Table 9:Hindi to Urdu scriptural translation restuls

The finite-state scriptural translation system for Hindi to Urdu produces an Urdu output with diacritics. However, we know that the Urdu community is used to see the Urdu text without diacritics. Thus, we removed all diacritical marks from the Urdu output text that is more acceptable to the Urdu community. By this post-processing, we gain more than 40% accuracy in case of HU Test Set 1. We also gain in accuracy for the other test sets.

For the classification of our scriptural translation systems, we have devised two scales. One corresponds to the word accuracy rate and the other corresponds to the sentence level accuracy. They are shown in Figure 7 and 8.

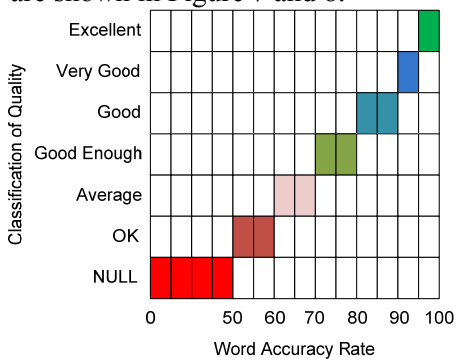


Figure 7: Classification scale based on the word accuracy rate for scriptural transaltion

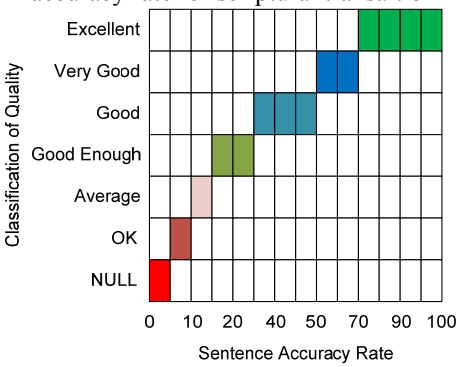


Figure 8: Classification scale based on the sentence accucary rate for scriptural translation

According to the scale of Figure 7 and 8, the Hindi to Urdu scriptural translation system is classified as 'Good' and 'Good Enough', respectively.

The subjective evaluations like usability, effectiveness and adequacy depend on several factors. A user with a good knowledge of Hindi and Urdu languages would rate our Hindi to Urdu system quite high and would also rate the Urdu output very usable. Another user who wants to read a Hindi text, but does not know Hindi, would also rate this system and the Urdu output quite high and very usable respectively, because it serves its purpose.

On the other hand, a user who wants to publish a Hindi book in Urdu, would rate this system not very good. This is because he has to localize the Hindi vocabulary of Sanskrit origin as the acceptance of the Hindi vocabulary in the Urdu

[3] Shared by University of Chicago for research purposes.

[4] http://www.itl.nist.gov/iad/mig//tools/

community, target of his published book, is very low. Thus the subjective evaluation depends on various factors and it is not easy to compute such measures for the evaluation of a scriptural translation system, but they are vital in real life.

For Urdu to Hindi scriptural translation, we have two inputs for each HU Test Set. One input contains all diacritical marks and the other does not contain any. On Hindi side, we have a single Hindi reference with which we will compare both Hindi outputs. We already know that it will give us less accuracy rates for the Urdu input without diacritical marks that are mandatory for correct Urdu to Hindi scriptural translation. The results for Urdu to Hindi scriptural translation are given in Table 10.

Test Set	With diacritics		Without diacritics	
	Word Level	Sentence Level	Word Level	Sentence Level
HU 1	68.0%	-	31.2%	-
HU 2	83.9%	10%	53.0%	1%
HU 3	98.4%	73.9%	58.9%	0.4%

Table 10: Urdu to Hindi scriptural translation results

For the Urdu input with diacritics, the accuracy of the Urdu to Hindi finite-state scriptural translation system is 83.9% at word level for HU Test Set 2 and it is classified as 'GOOD' the classification scale of Figure 7. On the other hand, it shows a sentence-level accuracy of 10% for the same test set and is classified as 'AVERAGE' by the classification scale of Figure 8.

For the Urdu input without diacritics, the Urdu to Hindi scriptural translation system is classified as 'OK' by the scale of Figure 7 for HU Test set 2 and 3. It is classifies as 'NULL' for HU Test Set 1. According to the scale of Figure 8, it is classified as 'NULL' for all three test sets.

For Punjabi scriptural translation, we also developed two types of output default and processed for Gurmukhi to Shahmukhi translation. In the reverse direction, it has two types of inputs, one with diacritics and the other without diacritics. Table 11 and 12 shows results of Punjabi scriptural translation.

Test Set	Default output		Processed output	
	Word Level	Sentence Level	Word Level	Sentence Level
PU	84.2%	27.8%	85.2%	29.9%

Table 11: Gurmukhi to Shahmukhi scriptural translation results

Test Set	With diacritics		Without diacritics	
	Word Level	Sentence Level	Word Level	Sentence Level
PU	98.8%	90.3%	67.3%	6.4%

Table 12: Shahmukhi to Gurmukhi scriptural translation results

Compared to the Hindi–Urdu pair, the Punjabi/Shahmukhi–Punjabi/Gurmukhi pair is computationally less hard. The post-processing to the default out of the finite-state scriptural translation systems for Punjabi/Gurmukhi to Punjabi/Shahmukhi also helps to gain an increase of approximately 1% and 2% at word and sentence levels respectively. The Shahmukhi to Gurmukhi scriptural translation system is classified as 'GOOD' by both scales of Figure 7 and 8. Thus the usability of the Punjabi finite-state scriptural translation system is higher than the Hindi–Urdu finite-state scriptural translation system.

In the reverse direction, the Shahmukhi to Gurmukhi scriptural translation system gives an accuracy of 98.8% and 67.3% for the Shahmukhi input text with and without diacritics respectively. For the Shahmukhi input text with diacritics, the scriptural translation system is classified as 'EXCELLENT' by both scales. On the other hand, it is classified as 'NULL' according to the scale of Figure 8 for the Shahmukhi input text without diacritical marks.

Similar to Hindi–Urdu and Punjabi finite-state scriptural translation, we have applied our finite-state system to the Seraiki test set. Here again, we have developed a processed Seraiki/Shahmukhi output from the default output of our finite-state system by removing the diacritics. The results are given in Table 13 and 14.

Test Set	Default output		Processed output	
	Word Level	Sentence Level	Word Level	Sentence Level
SE	81.3%	19.4%	83.7%	20.3%

Table 13: Seraiki/Devanagari to Seraiki/Shahmukhi scriptural translation results

Test Set	With diacritics		Without diacritics	
	Word Level	Sentence Level	Word Level	Sentence Level
SE	95.2%	76.4%	58.6%	8.6%

Table 14: Seraiki/Shahmukhi to Seraiki/Devanagari scriptural translation results

In the case of the Seraiki/Devanagari to Seraiki/Shahmukhi scriptural translation system, the post-processing also helps to gain an increase in word accuracy of approximately 1 to 2 percent

both at the word and the sentence levels. The accuracy for both the default and the processed Seraiki/Shahmukhi outputs is also more than 80% at word level. The system is classified as 'GOOD' and 'GOOD ENOUGH' according to the scale of Figure 7 and 8 respectively.

The absence of diacritical marks in the Seraiki/Shahmukhi has a very bad effect on the accuracy of the finite-state scriptural translation system. The scriptural translation system is classified as 'NULL' for the Seraiki/Shahmukhi input text without diacritics.

7 Conclusion

Finite-state methods are robust and efficient to implement scriptural translation rules in a very precise and compact manner.

The missing information and the diacritical marks in the source text proved to be very critical, crucial and important for achieving high and accurate results. The above results support our hypothesis that lack of important information in the source texts considerably lowers the quality of scriptural translation. They are crucial and their absence in the input texts decreases the performance considerably, from more than 80% to less than 60% at word level. Thus restoration of the missing information and the diacritical marks or reducing the effect of their absence on the scriptural translation is one of the major questions for further study and work.

In general, only word accuracy rates are reported. We have observed that only word accuracy rates may depict a good performance, but the performance of the same system at sentence-level may be not very good. Therefore, subjective evaluations and usage of translation results in real life should also be considered while evaluating the translation quality.

Acknowledgments

This study is supported by Higher Education Commission (HEC), Government of Pakistan under its overseas PhD scholarship scheme. We are also thankful to Digital South Asian Library, University of Chicago for sharing Platts dictionary data (Platts, 1884).

References

AbdulJaleel, N. and L. S. Larkey. 2003. Statistical Transliteration for English-Arabic Cross Language Information Retrieval. 12th international Conference on information and Knowledge Management (CIKM 03), New Orleans. 139-146.

Al-Onaizan, Y. and K. Knight. 2002. Machine Transliteration of Names in Arabic Text. Workshop on Computational Approaches To Semitic Languages, the 40th Annual Meeting of the ACL, Philadelphia, Pennsylvania, 1-13.

Arbabi, M., S. M. Fischthal, V. C. Cheng and E. Bart. 1994. Algorithms for Arabic Name Transliteration. *IBM J. Res. Dev.* 38(2): 183-193.

Boitet, C. and F. X. Tchéou. 1990. On a Phonetic and Structural Encoding of Chinese Characters in Chinese texts. ROCLING III, Taipeh. 73-80.

Fujii, A. and T. Ishikawa. 2001. Japanese/English Cross-Language Information Retrieval: exploration of query translation and transliteration. *Computers and the Humanities* 35(4): 389-420.

Halpern, J. 2002. Lexicon-based Orthographic Disambiguation in CJK Intelligent Information Retrieval. 3rd workshop on Asian language resources and international standardization, the 19th International Conference on Computational Linguistics (COLING), Taipei, Taiwan. 1-7.

Hieronymus, J. 1993. ASCII Phonetic Symbols for the World's Languages: Worldbet. AT&T Bell Laboratories.

Jeong, K. S., S. H. Myaeng, J. S. Lee and K.-S. Choi. 1999. Automatic Identification and Back-transliteration of Foreign Words for Information Retrieval. *Information Processing and Management* 35: 523-540.

Kang, B. and K. Choi. 2000. Automatic Transliteration and Back Transliteration by Decision Tree Learning. 2nd International Conference on Evaluation and Language Resources (ELRC), Athens.

Kaplan, R. M. and M. Kay. 1994. Regular Models of Phonological Rule Systems. 20(3).

Knight, K. and Y. Al-Onaizan. 1998. Translation with Finite-State Devices 3rd Conference of the Association for Machine Translation in the Americas on Machine Translation and the Information Soup (AMTA-98), Pennsylvania. 421-437.

Lee, J. S. and K. S. Choi. 1998. English to Korean Statistical Transliteration for Information Retrieval. *Computer Processing of Oriental languages* 12(1): 17-37.

Malik, M. G. A. 2005. Towards a Unicode Compatible Punjabi Character Set. 27th Internationalization and Unicode Conference, Berlin.

Malik, M. G. A., L. Besacier, C. Boitet and P. Bhattacharyya. 2009. A Hybrid Model for Urdu Hindi Transliteration. Joint conference of the 47th Annual Meeting of the Association of Computational Linguistics and the 4th

International Joint Conference on Natural Language Processing of the Asian Federation of NLP ACL/IJCNLP Workshop on Named Entities (NEWS-09), Singapore, 177–185.

Malik, M. G. A., C. Boitet and P. Bhattacharyya. 2008. Hindi Urdu Machine Transliteration using Finite-state Transducers. 22nd International Conference on Computational Linguistics (COLING), Manchester, 537-544.

Malik, M. G. A., C. Boitet and P. Bhattacharyya. 2010. Analysis of Noori Nast'aleeq for Major Pakistani Languages. 2nd Workshop on Spoken Language Technologies for Under-resourced Languages SLTU-2010, Penang, Malaysia.

Mohri, M. 1997. Finite-state Transducers in Language and Speech Processing. 23(2).

Pirkola, A., J. Toivonen, H. Keskustalo, K. Visala and K. Järvelin. 2003. Fuzzy Translation of Cross-lingual Spelling Variants. 26th Annual international ACM SIGIR Conference on Research and Development in informaion Retrieval, Toronto.

Platts, J. T. 1884. A Dictionary of Urdu, Classical Hindi and Englsih. W. H. Allen & Co.

Rahman, T. 1997. *Language and Politics in Pakistan*. Oxford University Press, Lahore.

Rahman, T. 2004. Language Policy and Localization in Pakistan: Proposal for a Paradigmatic Shift. Crossing the Digital Divide, SCALLA Conference on Computational Linguistics, Katmandu.

Rai, A. 2000. *Hindi Nationalism*. Orient Longman Private Limited, New Delhi.

Roche, E. and Y. Schabes, Eds. 1997. Finite-state Language Processing. MIT Press, Cambridge.

Sakai, T., A. Kumano and T. Manabe. 2002. Generating Transliteration Rules for Cross-language Information Retrieval from Machine Translation Dictionaries. IEEE Conference on Systems, Man and Cybernatics.

Stall, B. and K. Knight. 1998. Translating Names and Technical Terms in Arabic Text. Workshop on Computational Approaches to Semitic Languages, COLING/ACL, Montreal, 34-41.

Virga, P. and S. Khudanpur. 2003. Transliteration of Proper Names in Cross-language Applications. 26th Annual international ACM SIGIR Conference on Research and Development in informaion Retrieval, Toronto.

Wells, J. C. 1995. Computer-coding the IPA: a proposed extension of SAMPA. University College London.

Yan, Q., G. Gregory and A. E. David. 2003. Automatic Transliteration for Japanese-to-English Text Retrieval. 26th annual international ACM SIGIR conference on Research and development in information retrieval, 353-360.

Zia, K. 1999. Standard Code Table for Urdu. 4th Symposium on Multilingual Information Processing (MLIT-4), Yangon.

Dimensionality Reduction for Text using Domain Knowledge

Yi Mao and **Krishnakumar Balasubramanian** and **Guy Lebanon**
Georgia Institute of Technology

Abstract

Text documents are complex high dimensional objects. To effectively visualize such data it is important to reduce its dimensionality and visualize the low dimensional embedding as a 2-D or 3-D scatter plot. In this paper we explore dimensionality reduction methods that draw upon domain knowledge in order to achieve a better low dimensional embedding and visualization of documents. We consider the use of geometries specified manually by an expert, geometries derived automatically from corpus statistics, and geometries computed from linguistic resources.

1 Introduction

Visual document analysis systems such as INSPIRE have demonstrated their applicability in managing large text corpora, identifying topics within a document and quickly identifying a set of relevant documents by visual exploration. The success of such systems depends on several factors with the most important one being the quality of the dimensionality reduction. This is obvious as visual exploration can be made possible only when the dimensionality reduction preserves the structure of the original space, i.e., documents that convey similar topics are mapped to nearby regions in the low dimensional 2D or 3D space.

Standard dimensionality reduction methods such as principal component analysis (PCA), locally linear embedding (LLE) (Roweis and Saul, 2000), or t-distributed stochastic neighbor embedding (t-SNE) (van der Maaten and Hinton, 2008) take as input a set of feature vectors such as bag of words. An obvious drawback is that such methods ignore the textual nature of documents and instead consider the vocabulary words $v_1, \ldots, v_n$ as abstract orthogonal dimensions.

In this paper we introduce a framework for incorporating domain knowledge into dimensionality reduction for text documents. Our technique does not require any labeled data, therefore is completely unsupervised. In addition, it applies to a wide variety of domain knowledge.

We focus on the following type of non-Euclidean geometry where the distance between document x and y is defined as

$$d_T(x,y) = \sqrt{(x-y)^\top T(x-y)}. \qquad (1)$$

Here $T \in \mathbb{R}^{n\times n}$ is a symmetric positive semidefinite matrix, and we assume that documents x, y are represented as term-frequency (tf) column vectors. Since T can always be written as $H^\top H$ for some matrix $H \in \mathbb{R}^{n\times n}$, an equivalent but sometimes more intuitive interpretation of (1) is to compose the mapping $x \mapsto Hx$ with the Euclidean geometry

$$d_T(x,y) = d_I(Hx,Hy) = \|Hx - Hy\|_2. \quad (2)$$

We can view T as encoding the semantic similarity between pairs of words and H as smoothing the tf vector by mapping observed words to related but unobserved words. Therefore, the geometry realized by (1) or (2) may be used to derive novel dimensionality reduction methods that are customized to text in general and to specific text domains in particular. The main challenge is to obtain the matrices H or T that describe the relationship among vocabulary words appropriately.

We consider three general ways of obtaining H or T using domain knowledge. The first corresponds to manually specifying H or T based on the semantic relationship among words (determined by domain expert). The second corresponds to constructing H or T by analyzing relationships between different words using corpus statistics. The third is based on knowledge obtained from linguistic resources. Whether to specify H directly or indirectly by specifying T =

Coling 2008: Poster Volume, pages 801–809,
Beijing, August 2010

$H^\top H$ depends on the knowledge type and is discussed in detail in Section 4.

We investigate the performance of the proposed dimensionality reduction methods for three text domains: sentiment visualization for movie reviews, topic visualization for newsgroup discussion articles, and visual exploration of ACL papers. In each of these domains we evaluate the dimensionality reduction using several different quantitative measures. All the techniques mentioned in this paper are unsupervised, making use of labels only for evaluation purposes.

Our take home message is that all three approaches mentioned above improves dimensionality reduction for text upon standard embedding ($H = I$). Furthermore, geometries obtained from corpus statistics are superior to manually constructed geometries and to geometries derived from standard linguistic resources such as WordNet. Combining heterogenous types of knowledge provides the best results.

2 Related Work

Despite having a long history, dimensionality reduction is still an active research area. Broadly speaking, dimensionality reduction methods may be classified as projective or manifold based (Burges, 2009). The first projects data onto a linear subspace (e.g., PCA and canonical correlation analysis) while the second traces a low dimensional nonlinear manifold on which data lies (e.g., multidimensional scaling, isomap, Laplacian eigenmaps, LLE and t-SNE). The use of dimensionality reduction for text documents is surveyed by Thomas and Cook (2005) who also describe current homeland security applications.

Dimensionality reduction is closely related to metric learning. Xing et al. (2003) is one of the earliest papers that focus on learning metrics of the form (1). In particular they try to learn matrix T in an supervised way by expressing relationships between pairs of samples. A representative paper on unsupervised metric learning for text documents is Lebanon (2006) which learns a metric on the simplex based on the geometric volume of the data.

We focus in this paper on visualizing a corpus of text documents using a 2-D scatter plot. While this is perhaps the most popular and practical text visualization technique, other methods such as Spoerri (1993), Hearst (1997), Havre et al. (2002), Paley (2002), Blei et al. (2003), Mao et al. (2007) exist. Techniques developed in this paper may be ported to enhance these alternative visualization methods as well.

3 Non-Euclidean Geometries

Dimensionality reduction methods often assume, either explicitly or implicitly, Euclidean geometry. For example, PCA minimizes the reconstruction error for a family of Euclidean projections. LLE uses the Euclidean geometry as a local metric. t-SNE is based on a neighborhood structure, determined again by the Euclidean geometry. The generic nature of the Euclidean geometry makes it somewhat unsuitable for visualizing text documents as the relationship between words conflicts with Euclidean orthogonality. We consider in this paper several alternative geometries of the form (1) or (2) which are more suited for text and compare their effectiveness in visualizing documents.

As mentioned in Section 1, H smooths the tf vector x by mapping the observed words into observed and non-observed (but related) words. In case H is nonnegative, it can be further decomposed into a product of a non-negative column normalized matrix $R \in \mathbb{R}^{n\times n}$ and a non-negative diagonal matrix $D \in \mathbb{R}^{n\times n}$. The decomposition $H = RD$ shows that H has two key roles. It smooths related vocabulary words (realized by R) and it emphasizes some words over others (realized by D). Setting R_{ij} to a high value if w_i, w_j are similar and 0 if they are unrelated maps an observed word to a probability vector over related words in the vocabulary. The value D_{ii} captures the importance of v_i and therefore should be higher for important content words than for less important words or stop-words[1].

It is instructive to examine the matrices R and D in the case where the vocabulary words cluster in some meaningful way. Figure 1 gives an example where vocabulary words form two clusters. The matrix R may become block-diagonal with non-zero elements occupying diagonal blocks representing within-cluster word

[1]The nonnegativity assumption of H is useful when constructing H by domain experts such as the method A in Section 4. In general, H needs not to be nonnegative for dimensionality reduction as in (2).

$$\begin{pmatrix} 0.8 & 0.1 & 0.1 & 0 & 0 \\ 0.1 & 0.8 & 0.1 & 0 & 0 \\ 0.1 & 0.1 & 0.8 & 0 & 0 \\ 0 & 0 & 0 & 0.9 & 0.1 \\ 0 & 0 & 0 & 0.1 & 0.9 \end{pmatrix} \begin{pmatrix} 5 & 0 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 & 0 \\ 0 & 0 & 5 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 3 \end{pmatrix}$$

Figure 1: An example of a decomposition $H = RD$ in the case of two word clusters $\{v_1, v_2, v_3\}$, $\{v_4, v_5\}$. The block diagonal elements in R represent the fact that words are mostly mapped to themselves, but sometimes are mapped to other words in the same cluster. The diagonal matrix indicates that the first cluster is more important than the second cluster for the purposes of dimensionality reduction.

blending, i.e., words within each cluster are interchangeable to some degree. The diagonal matrix D represents the importance of different clusters. The word clusters are formed with respect to the visualization task at hand. For example, in the case of visualizing the sentiment content of reviews we may have word clusters labeled as "positive sentiment words", "negative sentiment words" and "objective words".

In general, the matrices R, D may be defined based on the language or may be specific to document domain and visualization purpose. It is reasonable to expect that the words emphasized for visualizing topics in news stories might be different than the words emphasized for visualizing writing styles or sentiment content.

Applying the geometry (1) or (2) to dimensionality reduction is easily accomplished by first mapping document tf vectors $x \mapsto Hx$ and proceeding with standard dimensionality reduction techniques such as PCA or t-SNE. The resulting dimensionality reduction is Euclidean in the transformed space but non-Euclidean in the original space. In many cases, the vocabulary contains tens of thousands of words or more making the specification of T or H a complicated and error prone task. We describe in the next section several techniques for specifying these matrices in practice.

4 Domain Knowledge

Method A: Manual Specification

In this method, a domain expert manually specifies $H = RD$ by specifying (R, D) based on the perceived relationship among the vocabulary words. More specifically, the user first constructs a hierarchical word clustering that may depend on the current text domain, and then specifies the matrices (R, D) based on the clustering.

Denoting the clusters by $C_1, \ldots, C_r$ (a partition of $\{v_1, \ldots, v_n\}$), R is set to

$$R_{ij} \propto \begin{cases} \rho_a, & i = j, v_i \in C_a \\ \rho_{ab}, & i \neq j, v_i \in C_a, v_j \in C_b \end{cases}.$$

The values $\rho_{ab}, a \neq b$ capture the semantic similarity between two clusters and the value ρ_{aa} captures the similarity of two different words within the cluster a. These values may be set manually by domain expert or automatically computed based on the clustering hierarchy (for example ρ_{ab} can be the inverse of the minimal number of tree edges traversed in moving from a to b). To maintain a probabilistic interpretation, the matrix R should be normalized so that its columns sum to 1. The diagonal matrix D is specified by setting the values

$$D_{ii} = d_a, \quad v_i \in C_a$$

according to the importance of word cluster C_a to the current visualization task.

We emphasize that as with the rest of the methods in this paper, the manual specification is done without access to labeled data. Since manual clustering assumes some form of human intervention, it is reasonable to also consider cases where the user specifies H or T in an interactive manner. For example, the expert specifies an initial clustering of words and values for (R, D), views the resulting embeddings and adjusts the selection interactively until reaching a satisfactory embedding.

Method B: Contextual Diffusion

An alternative to manually specifying $T = DR^\top RD$ is to construct it based on similarity between the contextual distributions of the vocabulary words. The contextual distribution of word v is defined as

$$q_v(w) = p(w \text{ appears in } x | v \text{ appears in } x) \quad (3)$$

where x is a randomly drawn document. In other words q_v is the distribution governing the words appearing in the context of word v.

A natural similarity measure between distributions is the Fisher diffusion kernel proposed by Lafferty and Lebanon (2005). Applied to contextual distributions as in Dillon et al. (2007) we arrive at the following similarity matrix

$$T(u,v) = \exp\left(-c \arccos^2\left(\sum_w \sqrt{q_u(w) q_v(w)}\right)\right).$$

where $c > 0$. Intuitively, the word u will be diffused into v depending on the geometric diffusion between the distributions of likely contexts.

We use the following formula to estimate the contextual distribution from a corpus

$$\begin{aligned} q_v(w) &= \sum_{x'} p(w, x'|v) = \sum_{x'} p(w|x', v) p(x'|v) \\ &= \sum_{x'} \mathrm{tf}(w, x') \frac{\mathrm{tf}(v, x')}{\sum_{x''} \mathrm{tf}(v, x'')} \qquad (4) \\ &= \left(\frac{1}{\sum_{x'} \mathrm{tf}(v, x')}\right) \left(\sum_{x'} \mathrm{tf}(w, x') \mathrm{tf}(v, x')\right) \end{aligned}$$

where $\mathrm{tf}(w, x)$ is the number of times word w appears in document x divided by the length of the document x. The contextual distribution q_v or diffusion matrix T above may be computed in an unsupervised manner without labels.

Method C: Web n-Grams

In method B the contextual distribution is computed using a large external corpus that is similar to the text being analyzed. An alternative that is especially useful when such a corpus is not easily available is to use generic resources to estimate the contextual distribution (3)-(4). One option is to use the publicly available Google n-gram dataset (Brants and Franz, 2006) to estimate T. More specifically, we compute the contextual distribution by considering the proportion of times two words appear together within the n-grams e.g., for $n = 2$ we have

$$q_v(w) = \frac{\#\text{ of bigrams containing both } w \text{ and } v}{\#\text{ of bigrams containing } v}.$$

Method D: Word-Net

In the last method, we consider using Word-Net, a standard linguistic resource, to specify T. This is similar to manual specification (method A) in that it builds upon experts' knowledge rather than corpus statistics. In contrast to method A, however, Word-Net is a carefully built resource containing more accurate and comprehensive linguistic information such as synonyms, hyponyms and holonyms. On the other hand, its generality puts it at a disadvantage as method A may be adapted to a specific text domain.

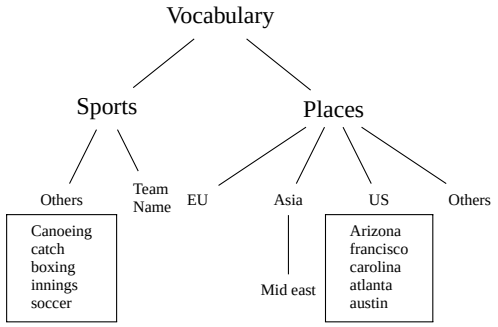


Figure 2: Manually specified hierarchical word clustering for the 20 newsgroup domain. The words in the frames are examples of words belonging to several bottom level clusters.

We follow Budanitsky and Hirst (2001) who compared five similarity measures between words based on Word-Net. In our experiments we use the measure of Jiang and Conrath (1997) (see also Jurafsky and Martin (2008))

$$T(u,v) = \log \frac{p(u)p(v)}{2p(\mathrm{lcs}(u,v))}$$

as it was shown to outperform the others. Above, lcs stands for the lowest common subsumer, i.e., the lowest node in the hierarchy that subsumes (is a hypernym of) both u and v. The quantity $p(u)$ is the probability that a randomly selected word in a corpus is an instance of the synonym set that contains word u.

Combination of Methods

In addition to individual methods we also consider their convex combinations

$$H^* = \sum_i \alpha_i H_i \quad \text{s.t.} \quad \alpha_i \geq 0, \sum_i \alpha_i = 1 \qquad (5)$$

where H_i are matrices from methods A-D (obtained implicitly by specifying R and D for method A and T for methods B-D). Doing so allows us to combine heterogeneous types of domain knowledge including experts' knowledge

and corpus statistics, leverage their diverse nature and potentially achieve better performance than any of the methods on its own.

5 Experiments

We evaluate the proposed methods by experimenting on two text datasets where domain knowledge is relatively easy to obtain (especially for method A and B). Preprocessing includes lower-casing, stop words removal, stemming, and selecting the most frequent 2000 words for both datasets.

The first is the Cornell sentiment scale dataset of movie reviews from 4 critics (Pang and Lee, 2004). The visualization in this case focuses on the sentiment quantity of either 1 (very bad) or 4 (very good) (Pang et al., 2002). For method A, we use the General Inquirer resource[2] to partition the vocabulary into three clusters conveying positive, negative or neutral sentiment. While visualizing documents from one particular author, the rest of the reviews from other three authors can be used as an estimate of contextual distribution for method B.

The second text dataset is the 20 newsgroups. It consists of newsgroup articles from 20 distinct newsgroups and is meant to demonstrate topic visualization. In this case one of the authors designed a hierarchical clustering of the vocabulary words based on general knowledge of English language (see Figure 2 for a partial clustering hierarchy) without access to labels. The contextual distribution for method B is estimated from the Reuters RCV1 dataset (Lewis et al., 2004) which consists of news articles from Reuters.com in the year 1996 and 1997.

Method C uses Google n-gram which provides a massive scale resource for estimating the contextual distribution. In the case of Word-Net (method D) we used Pedersen's implementation of Jiang and Conrath's similarity measure[3]. Note, for these two methods, the obtained information is not domain specific but rather represents general semantic relationships between words.

In our experiments below we focused on two dimensionality reduction methods: PCA and t-SNE. PCA is a well known classical method while t-SNE (van der Maaten and Hinton, 2008) is a recent dimensionality reduction technique for visualization purposes. The use of t-SNE is motivated by the fact that it was shown to outperform LLE, CCA, MVU, Isomap, and Laplacian eigenmaps when the dimensionality of the data is reduced to two or three.

To measure the dimensionality reduction quality, we visualize the data as a scatter plot with different data groups (topics, sentiments) displayed with different markers and colors. Our quantitative evaluation of the visualization is based on the fact that documents belonging to different groups (topics, sentiments) should be spatially separated in the 2-D space. Specifically, we used the following indices:

(i) The weighted intra-inter criteria is a standard clustering quality index that is invariant to non-singular linear transformations of the embedded data. It equals $\mathrm{tr}(S_T^{-1}S_W)$ where S_W is the within-cluster scatter matrix, $S_T = S_W + S_B$ is the total scatter matrix, and S_B is the between-cluster scatter matrix (Duda et al., 2001).

(ii) The Davies Bouldin index is an alternative to (i) that is similarly based on the ratio of within-cluster scatter to between-cluster scatter (Davies and Bouldin, 2000).

(iii) Classification error rate of a k-NN classifier that applies to data groups in the 2-D embedded space. Despite the fact that we are not interested in classification per se (otherwise we would classify in the original high dimensional space), it is an intuitive and interpretable measure of cluster separation.

(iv) An alternative to (iii) is to project the embedded data onto a line which is the direction returned by applying Fisher's linear discriminant analysis to the embedded data. The projected data from each group is fitted to a Gaussian whose separation is used as a proxy for visualization quality. In particular, we summarize the separation of the two Gaussians by measuring the overlap area. While (iii) corresponds to the performance of a k-NN classifier, method (iv) corresponds to the performance of Fisher's LDA classifier.

Labeled data is not used during the dimensionality reduction stage but it is used in each of the above measures for evaluation purposes.

[2] http://www.wjh.harvard.edu/∼inquirer/

[3] http://wn-similarity.sourceforge.net/

Figure 3 displays both qualitative and quantitative evaluation of PCA and t-SNE for the sentiment and newsgroup domains for $H = I$ (left column), manual specification (middle column) and contextual distribution (right column). In general for both domains, methods A and B perform better both qualitatively and quantitatively (indicating by the numbers in the top two rows) than the original dimensionality reduction with method B outperforming method A.

Tables 1-2 compare evaluation measures (i) and (iii) for different types of domain knowledge. Table 1 corresponds to the sentiment domain where we conducted separate experiments for four movie critics. Table 2 corresponds to the newsgroup domain where two tasks were considered. The first involves three newsgroups (comp.sys.mac.hardware vs. rec.sports.hockey vs. talk.politics.mideast) and the second involves four newsgroups (rec.autos vs. rec.motorcycles vs. rec.sports.baseball vs. rec.sports.hockey). It is clear from these two tables that the contextual diffusion, Google n-gram, and Word-Net generally outperform the original $H = I$ matrix. The best method varies from task to task but the contextual diffusion and Google n-gram in general result in good performance.

	PCA (1)	PCA (2)	t-SNE (1)	t-SNE (2)
$H = I$	1.5391	1.4085	1.1649	1.1206
B	1.2570	**1.3036**	1.2182	1.2331
C	**1.2023**	1.3407	**0.7844**	**1.0723**
D	1.4475	1.3352	1.1762	1.1362
	PCA (1)	PCA (2)	t-SNE (1)	t-SNE (2)
$H = I$	0.8461	0.5630	0.9056	0.7281
B	0.7381	**0.6815**	0.9110	0.6724
C	0.8420	0.5898	**0.9323**	0.7359
D	**0.8532**	0.5868	0.9013	**0.7728**

Table 2: Quantitative evaluation of dimensionality reduction for visualization for two tasks in the news article domain. The numbers in the top five rows correspond to measure (i) (lower is better), and the numbers in the bottom five rows correspond to measure (iii) ($k = 5$) (higher is better). We conclude that contextual diffusion (B), Google n-gram (C), and Word-Net (D) tend to outperform the original $H = I$.

We also examined convex combinations

$$\alpha_1 H_A + \alpha_2 H_B + \alpha_3 H_C + \alpha_4 H_D \qquad (6)$$

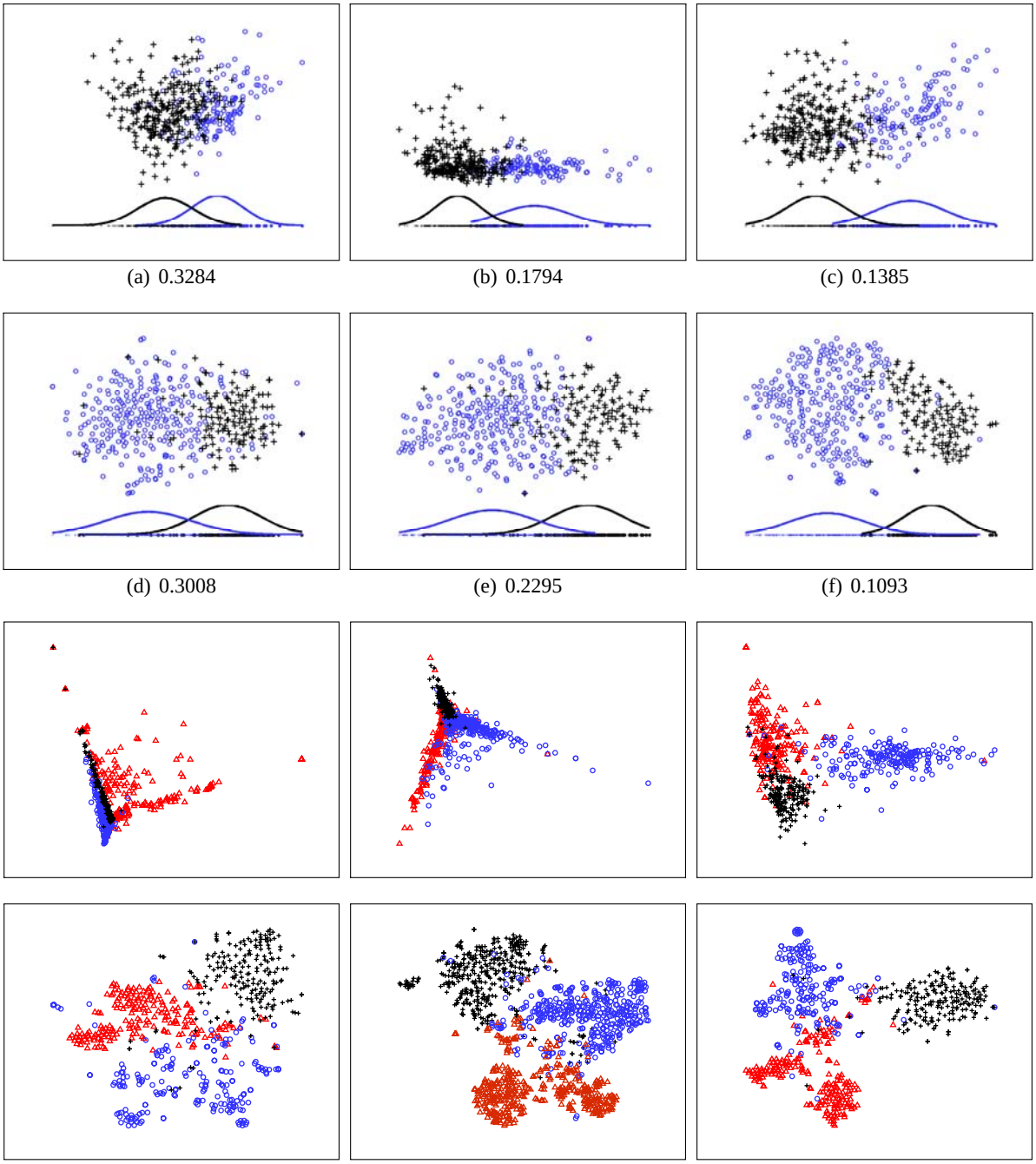
with $\sum \alpha_i = 1$ and $\alpha_i \geq 0$. Table 3 displays quantitative results using evaluation measures (i), (ii) and (iii) where k is chosen to be 5 for (iii). The first four rows correspond to method A, B, C and D and the bottom row corresponds to a convex combination found which minimizes the unsupervised evaluation measure (ii) (i.e. the search for the optimal combination is based on (ii) that does not require labeled data). Note that the convex combination also outperforms method A, B, C, and D for measure (i) and more impressively for measure (iii) which is a supervised measure that uses labeled data. In general, by combining heterogeneous types of domain knowledge, we may further improve the quality of dimensionality reduction for visualization, and the search for such a combination may be accomplished without the use of labeled data.

$(\alpha_1, \alpha_2, \alpha_3, \alpha_4)$	(i)	(ii)	(iii) (k=5)
(1,0,0,0)	0.5756	-3.9334	0.7666
(0,1,0,0)	0.5645	-4.6966	0.7765
(0,0,1,0)	0.5155	-5.0154	0.8146
(0,0,0,1)	0.6035	-3.1154	0.8245
(0.3,0.4,0.1,0.2)	**0.4735**	**-5.1154**	**0.8976**

Table 3: Three evaluation measures (i), (ii), and (iii) (see the beginning of the section for description) for convex combinations (6) using different values of α. The first four rows represent methods A, B, C, and D. The bottom row represents a convex combination whose coefficients were obtained by searching for the minimizer of measure (ii). Interestingly the minimizer also performs well on measure (i) and more impressively on the labeled measure (iii).

Finally, we demonstrate the effect of domain knowledge on a new dataset that consists of all oral papers appearing in ACL 2001 – 2009. For the purpose of manual specification, we obtain 1545 unique words from paper titles, and assign for each word relatedness scores for the following clusters: morphology/phonology, syntax/parsing, semantics, discourse/dialogue, generation/summarization, machine translation, retrieval/categorization and machine learning. The score takes value from 0 to 2, where 2 represents the most relevant. The score information is then used to generate the transformation matrix R. We also assign for each word an importance value ranging from 0 to 3 (larger the value, more important the word). This information is used to generate the diagonal matrix D.

Figure 4 shows the projection of all 2009 papers using t-SNE (papers from 2001 to 2008 are used to estimate contextual diffusion). Using Euclidean geometry $H = I$ (Figure 4 left) results in a Gaussian like distribution which does not provide much insight into the data. Using a manually



(a) 0.3284 (b) 0.1794 (c) 0.1385

(d) 0.3008 (e) 0.2295 (f) 0.1093

Figure 3: Qualitative evaluation of dimensionality reduction for the sentiment domain (top two rows) and the newsgroup domain (bottom two rows). The first and the third rows display PCA reduction while the second and the fourth display t-SNE. The left column correspond to no domain knowledge ($H = I$) reverting PCA and t-SNE to their original form. The middle column corresponds to manual specification (method A). The right column corresponds to contextual diffusion (method B). Different groups (sentiment labels or newsgroup labels) are marked with different colors and marks.
In the sentiment case (top two rows) the graphs were rotated such that the direction returned by applying Fisher linear discriminant onto the projected 2D coordinates aligns with the positive x-axis. The bell curves are Gaussian distributions fitted from the x-coordinates of the projected data points (after rotation). The numbers displayed in each sub-figure are computed from measure (iv).

	Dennis Schwartz		James Berardinelli		Scott Renshaw		Steve Rhodes	
	PCA	t-SNE	PCA	t-SNE	PCA	t-SNE	PCA	t-SNE
$H = I$	1.8625	1.8781	1.4704	1.5909	1.8047	1.9453	1.8013	1.8415
A	1.8474	1.7909	1.3292	1.4406	1.6520	1.8166	**1.4844**	1.6610
B	**1.4254**	**1.5809**	**1.3140**	**1.3276**	**1.5133**	**1.6097**	1.5053	**1.6145**
C	1.6868	1.7766	1.3813	1.4371	1.7200	1.8605	1.7750	1.7979
$H = I$	0.6404	0.7465	0.8481	0.8496	0.6559	0.6821	0.6680	0.7410
A	0.6011	0.7779	**0.9224**	0.8966	0.7424	0.7411	**0.8350**	0.8513
B	**0.8831**	**0.8554**	0.9188	**0.9377**	**0.8215**	**0.8332**	0.8124	**0.8324**
C	0.7238	0.7981	0.8871	0.9093	0.6897	0.7151	0.6724	0.7726

Table 1: Quantitative evaluation of dimensionality reduction for visualization in the sentiment domain. Each of the four columns corresponds to a different movie critic from the Cornell dataset (see text). The top five rows correspond to measure (i) (lower is better) and the bottom five rows correspond to measure (iii) ($k = 5$, higher is better). Results were averaged over 40 cross validation iterations. We conclude that all methods outperform the original $H = I$ with the contextual diffusion and manual specification generally outperforming the others.

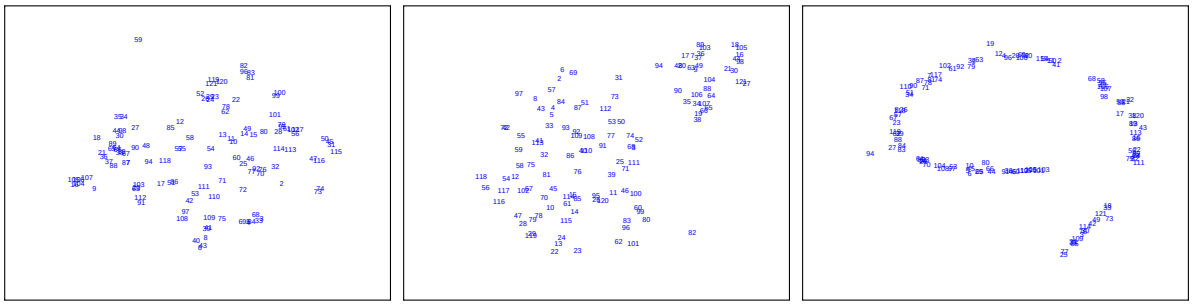


Figure 4: Qualitative evaluation of dimensionality reduction for the ACL dataset using t-SNE. Left: no domain knowledge ($H = I$); Middle: manual specification (method A); Right: contextual diffusion (method B). Each document is labeled by its assigned id from ACL anthology. See text for more details.

specified H (Figure 4 left) we get two clear clusters, the smaller containing papers dealing with machine translation and multilingual tasks. Interestingly, the contextual diffusion results in a one-dimensional manifold. Investigating the papers along the curve (from bottom to top) we find that it starts with papers discussing semantics and discourse (south), continues to structured prediction and segmentation (east), continues to parsing and machine learning (north), and then moves to sentiment prediction, summarization and IR (west) before returning to the center. Another interesting insight that we can derive is the relative discontinuity between the bottom part (semantics and discourse) and the rest of the curve. It seems spatial separability is higher in that area than in the other areas where the curve nicely traverses different regions continuously.

6 Discussion

In this paper we introduce several ways of incorporating domain knowledge into dimensionality reduction for visualizing text documents. The proposed methods all outperform in general the baseline $H = I$, which is the one currently used in most text visualization systems.

The answer to the question of which method is best depends on both the domain and the task at hand. For small tasks with limited vocabulary, manual specification could achieve best results. A large vocabulary size makes manual specification less accurate and effective. In cases where we have access to a large external corpus that is similar to the one we are interested in visualizing, contextual diffusion is an excellent choice. Lacking such a domain specific dataset estimating the contextual distribution using the generic Google n-gram is a good substitute. Word-Net captures relationships (such as synonyms and hyponyms) other than occurrence statistics between vocabulary words, and could be useful for certain tasks. Finally, the effectiveness of dimensionality reduction methods can be increased further by carefully combining different types of domain knowledge ranging from semantic similarity to occurrence statistics.

References

Blei, D., A. Ng, , and M. Jordan. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.

Brants, T. and A. Franz. 2006. Web 1T 5-gram Version 1.

Budanitsky, A. and G. Hirst. 2001. Semantic distance in wordnet: An experimental, application-oriented evaluation of five measures. In *NAACL Workshop on WordNet and other Lexical Resources*.

Burges, C. 2009. Dimension reduction: A guided tour. Technical Report MSR-TR-2009-2013, Microsoft Research.

Davies, D. L. and D. W. Bouldin. 2000. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1(4):224–227.

Dillon, J., Y. Mao, G. Lebanon, and J. Zhang. 2007. Statistical translation, heat kernels, and expected distances. In *Uncertainty in Artificial Intelligence*, pages 93–100. AUAI Press.

Duda, R. O., P. E. Hart, and D. G. Stork. 2001. *Pattern classification*. Wiley New York.

Havre, S., E. Hetzler, P. Whitney, and L. Nowell. 2002. Themeriver: Visualizing thematic changes in large document collections. *IEEE Transactions on Visualization and Computer Graphics*, 8(1).

Hearst, M. A. 1997. Texttiling: Segmenting text into multi-paragraph subtopic passages. *Computational Linguistics*, 23(1):33–64.

Jiang, J. J. and D. W. Conrath. 1997. Semantic similarity based on corpus statistics and lexical taxonomy. In *International Conference Research on Computational Linguistics (ROCLING X)*.

Jurafsky, D. and J. H. Martin. 2008. *Speech and Language Processing*. Prentice Hall.

Lafferty, J. and G. Lebanon. 2005. Diffusion kernels on statistical manifolds. *Journal of Machine Learning Research*, 6:129–163.

Lebanon, G. 2006. Metric learning for text documents. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(4):497–508.

Lewis, D., Y. Yang, T. Rose, and F. Li. 2004. RCV1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 5:361–397.

Mao, Y., J. Dillon, and G. Lebanon. 2007. Sequential document visualization. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1208–1215.

Paley, W. B. 2002. TextArc: Showing word frequency and distribution in text. In *IEEE Symposium on Information Visualization Poster Compendium*.

Pang, B. and L. Lee. 2004. A sentimental eduction: sentiment analysis using subjectivity summarization based on minimum cuts. In *Proc. of the Association of Computational Linguistics*.

Pang, B., L. Lee, and S. Vaithyanathan. 2002. Thumbs up?: sentiment classification using machine learning techniques. In *Proc. of the Conference on Empirical Methods in Natural Language Processing*.

Roweis, S. and L. Saul. 2000. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290:2323–2326.

Spoerri, A. 1993. InfoCrystal: A visual tool for information retrieval. In *Proc. of IEEE Visualization*.

Thomas, J. J. and K. A. Cook, editors. 2005. *Illuminating the Path: The Research and Development Agenda for Visual Analytics*. IEEE Computer Society.

van der Maaten, L. and G. Hinton. 2008. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605.

Xing, E., A. Ng, M. Jordan, and S. Russel. 2003. Distance metric learning with applications to clustering with side information. In *Advances in Neural Information Processing Systems, 15*.

Varro: An Algorithm and Toolkit for Regular Structure Discovery in Treebanks

Scott Martens
Centrum voor Computerlinguïstiek, KU Leuven
scott@ccl.kuleuven.be

Abstract

The *Varro* toolkit is a system for identifying and counting a major class of regularity in treebanks and annotated natural language data in the form of tree-structures: frequently recurring unordered subtrees. This software has been designed for use in linguistics to be maximally applicable to actually existing treebanks and other stores of tree-structurable natural language data. It minimizes memory use so that moderately large treebanks are tractable on commonly available computer hardware. This article introduces *condensed canonically ordered trees* as a data structure for efficiently discovering frequently recurring unordered subtrees.

1 Credits

This research is supported by the AMASS++ Project[1] directly funded by the *Institute for the Promotion of Innovation by Science and Technology in Flanders (IWT)* (SBO IWT 060051).

2 Introduction

Treebanks and similarly enhanced corpora are increasingly available for research, but these more complex structures are resistant to the techniques used in NLP for the statistical analysis of strings. This paper introduces a new treebank analysis suite *Varro*, named after Roman philologist Marcus Terentius Varro (116 BC-27 BC), who made linguistic regularity and irregularity central to his philosophy of language in *De Lingua Latina*. (Harris and Taylor, 1989)

The *Varro* toolkit focuses on a general problem in performing statistical analyses on treebanks: identifying, counting and extracting the distributions of frequently recurring unordered subtrees in treebanks. From this base, it is possible to construct more linguistically motivated schemes for performing treebank analysis. Complex statistical analyses are constructed from knowledge about frequency and distribution, so this constitutes a low level task on top of which higher level analyses can be performed.

An algorithm that can efficiently extract frequently recurring subtrees from treebanks has a number of immediate applications in computational linguistics:

- Speeding up treebank search algorithms like Tgrep2. (Rohde, 2001)
- Rule discovery for tree transducers used in parsing and machine translation. (Knight and Graehl, 2005; Knight, 2007)
- Generalizing lexical statistics techniques in NLP – e.g., collocation – to a broader array of linguistic structures. (Sinclair, 1991)
- Efficiently identifying useful features for tree kernel methods. (Moschitti, 2006)

3 Theory and Previous Work

For the purposes of this paper, a treebank is any collection of disjoint labeled trees. While in practice this mostly means parsed natural language sentences, the approach described here is equally applicable to other kinds of data, including semantic feature structures, morphological analyses, and

[1] http://www.cs.kuleuven.be/~liir/projects/amass/

Coling 2008: Poster Volume, pages 810–818,
Beijing, August 2010

doubtless many other kind of linguistically motivated structures. Figure 1 is an example of a parse tree from a Dutch-language treebank.

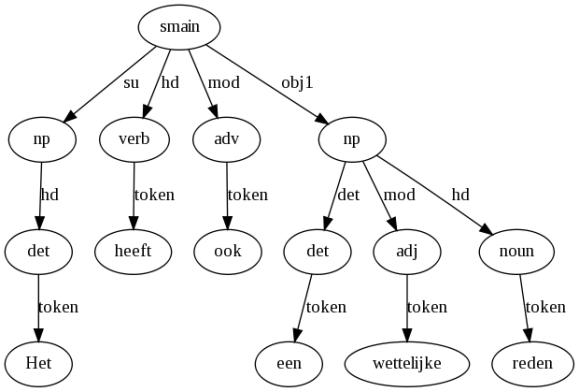


Figure 1: A tree from the Europarl Dutch corpus. (Koehn, 2005) It has been parsed and labeled automatically by the Alpino parser. (van Noord, 2006) A word-for-word translation is "*It also has a legal reason.*" (≈ "*There is also a legal reason (for that).*")

In this paper, we are concerned with identifying and counting *frequent induced unordered subtrees* in treebanks. The term *subtree* has a number of definitions, but this paper will follow the terminology of Chi et al. (2004). Figure 2 contains three examples of *induced unordered subtrees* of the tree in Figure 1. Note that the ordering of the vertices in the subtrees is different from that of Figure 1. This is what makes them *unordered subtrees*. *Induced subtrees* are more formally described in Section 4.

3.1 Apriori

The research builds on frequent subtree discovery algorithms based on the well-known *Apriori* algorithm, which is used to discover frequent itemsets in databases. (Agrawal et al., 1993) As a brief summary of *Apriori*, consider a collection of ordered itemsets $\mathbb{C} = \{\{a,b,c\},\{a,b,d\},\{b,c,d,e\}\}$. *Apriori* discovers all the subsets of those elements that appear at least some user-determined θ times. As an example, let us set $\theta = 2$, and then count the number of times each unique item appears in $\mathbb{C}$. Any single element in $\mathbb{C}$ that appears less than two times cannot be a member of a set of elements that appears at least θ times (since $\theta = 2$), so those are rejected. Each of the remaining set elements $\{a,b,c,d\}$ is extended by counting the number of two-element sets that include it and some element to the right in the ordered itemsets in $\mathbb{C}$. For b, these are $\{\{b,c\},\{b,d\},\{b,e\}\}$. Of this set, only those that appear at least θ times are retained: $\{\{b,c\},\{b,d\}\}$. This process is repeated for size three sets, and iterated over and over for increasingly large subsets, until there are no extensions that appear at least θ times. This whole procedure is then repeated for each unique item. Finally, *Apriori* will have extracted and counted all itemsets that appear at least θ times in $\mathbb{C}$.

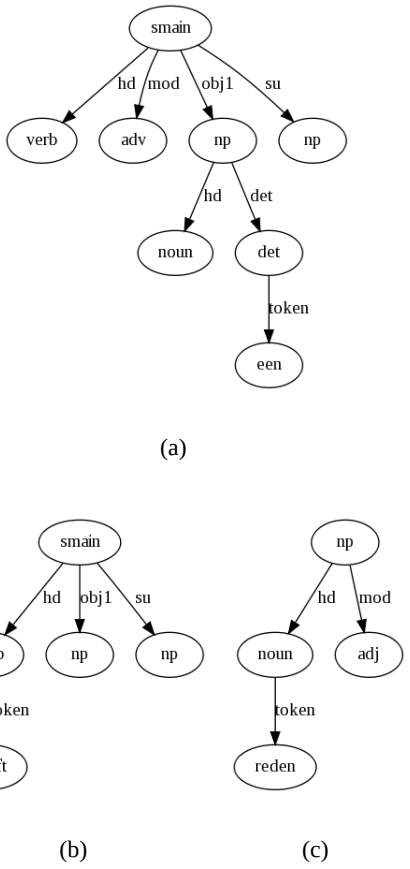


Figure 2: Three induced unordered subtrees of the tree in Figure 1

Extending *Apriori* to frequent subtree discovery dates to the work of Zaki (2002) and Asai et al. (2002). Chi et al. (2004) summarizes much of this line of research. In *Apriori*, larger and less frequent itemsets are discovered

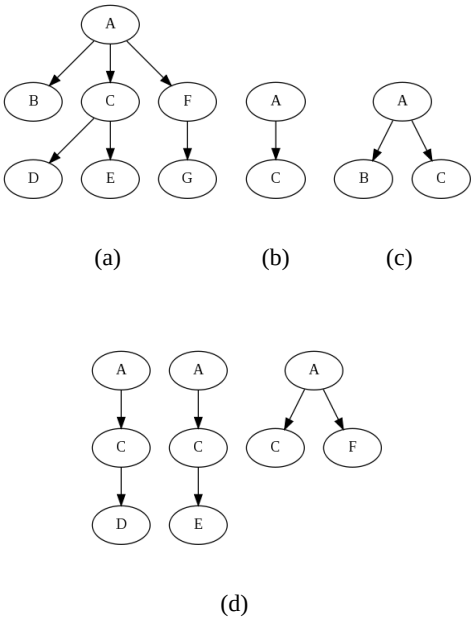


Figure 3: 3(b) and 3(c) are a subtrees of 3(a). The subtrees in 3(d) are possible extensions to 3(b), while 3(c) is not.

and counted by adding items to shorter and more frequent ones. This extends naturally to trees by initially locating and counting all the one-vertex trees in a treebank, and then constructing larger trees by adding vertices and edges to their right sides.

In Figure 3, subtree 3(b) has as valid extensions subtrees 3(d), all of which extend 3(b) to the right. An extension like subtree 3(c), which adds a node to the left of the rightmost node of 3(b), is not a valid extension.

3.2 Treebank applications

Applying these algorithms to natural language treebanks, however, presents a number of challenges.

The approach described above, because it constructs and tests subtrees by moving from left to right, is well-suited to finding *ordered subtrees*. However, this paper will consider *unordered subtrees* as better motivated linguistically. Word order is not completely fixed in any language, and can be very free in many important contexts.

But there are other problems as well. Apriori-style algorithms have the general property that their run-time is proportionate to the size of the output. Given a data-set D and a user-determined minimum frequency threshold θ, this class of solution outputs all the patterns that appear at least θ times in D. If D contains n patterns that appear at least θ times, $\mathbb{P} = \{p_1, p_2, ..., p_n\}; \forall p_i \in \mathbb{P} : freq(p_i) \geq \theta$, then the time necessary to identify and count all the patterns in $\mathbb{P}$ is proportionate to $\sum_{i=1}^{n} freq(p_i)$. In weakly correlated data, this is a very efficient method of finding patterns. In highly correlated data, however, the number of patterns present can become prohibitively large and extend run-time to unacceptable lengths, especially for small θ or large data-sets. Each frequent pattern may have any number of sub-patterns, each of which is also frequent and must be separately counted.

If we identify patterns with subtrees, a subtree with n vertices will, depending on its structure, have a minimum of $n(n-1)$ and a maximum of $(n-1)! + 1$ subtrees. If each of those subtrees is also a pattern that must be counted, then runtime grows very rapidly even for very small data-sets. Since natural language data is highly correlated, simple subtree-discovery extensions of *Apriori*, like those proposed in (Zaki, 2002) and (Asai, 2002), are not feasible for linguistic use. As reported in Martens (2009b), run-times become intractably long very quickly as data size increases for really existing treebanks.

However, there are compact representations of frequent patterns that are better suited to highly-correlated data and which can be efficiently discovered by modified *Apriori* schemes. This paper will only address one such representation: *frequent closures*. (Boulicaut and Bykowski, 2000) Frequent closures are widely used in subtree discovery and have an intuitive meaning when discussing natural language.

Given a treebank D, and a tree T that has a support of $freq(T) = \theta$, then T is *closed* if there is no supertree $T' \supset T$ where $freq(T') = \theta$. In Figure 3, if subtree 3(c) is as frequent in some treebank as 3(b), then 3(b) is not a closed subtree, nor can any further extension of it to the right be a closed subtree.

As a natural language example, given a corpus

of English sentences, let us assume we have found a pattern of the form *"NP make up NP to VP"*, such as in *"He has made up his mind to study linguistics."* If every time this pattern appears in the corpus, the second NP contains *"mind"*, then the pattern is *not* closed. A larger pattern appears just as often and in exactly the same places.

This makes the notion of frequent closed subtree discovery a generalization of *collocation* and *coligation* - well known in corpus-based lexicography - to arbitrary tree structures. (Sinclair, 1991) J.R. Firth famously said, "You shall know a word by the company it keeps." (Firth, 1957) Frequent subtree discovery tells us exactly what company entire linguistic structures keep.

3.3 Efficient closed subtree discovery

Chi et al. (2005a) outlines a general method for efficiently finding frequent closed subtrees without finding all frequent subtrees first. Their approach requires each subtree found to be aligned with its supertree before checking for closure and extensions. However, the alignment between a subtree and its supertree - the map from subtree vertices to supertree vertices - is not necessarily unique. A subtree may have a number of possible alignments with its supertree, even if one or more of the vertex alignments is specificed, as shown in Figure 4, which uses an example from the hand-corrected Alpino Treebank of Dutch.[2]

This can only be avoided by adding a restriction to trees: the combination of edge and vertex labels for each child of a vertex must be unique. This guarantees that specifying just one vertex in the alignment of a subtree to its supertree is enough to determine the entire unique mapping, but it is incompatible with most linguistic theories. Processes like tree binarization can meet this requirement, but only with some loss of generality: Some frequent closed subtrees in a collection of trees like Figure 4(a) will no longer be frequent, or will be less frequent, in a collection of binary trees.

Martens (2009a) describes an alternative method of checking for closure which does not require alignment and can, consequently, be much faster. It has, however, two drawbacks: First, it does not find all frequent closed unordered subtrees. Figure 5 shows the kind of tree where that approach is unable to correctly identify and count an unordered subtree. Second, it requires a great deal more memory than solutions that align each subtree discovered and check directly for closure, and is therefore of limited use with very large corpora.

[2] http://www.let.rug.nl/vannoord/trees/

4 Definitions

A *fully-labeled rooted tree* is a rooted tree in which each vertex and each edge has a label: $T := \langle V, E, L_V, L_E \rangle$, where V is the set of vertices, E is the set of edges, L_V is a map $L_V : V \to \mathbb{L}_V$ from the vertices to a set of labels; and similarly L_E maps the edges to labels $L_E : E \to \mathbb{L}_E$. We will designate an edge e connecting vertex v_1 to its child v_2 by the notation $e = \langle v_1, v_2 \rangle$. $\mathbb{L}_V$ and $\mathbb{L}_E$ constitute collectively the *lexicon*. Figure 1 is an example of a fully-labelled, rooted tree from a Dutch-language treebank. This formalization is broadly applicable to all linguistic formalisms whose structures are *tree-based* or can be converted one-to-one into trees without loss of generality. This may require some degree of restructuring of the tree formats used in particular linguistic theories. For example, in many formal linguistic theories, labels are not atomic symbols, but may have many parts or even whole structured feature sets. In general, these can be mapped to trees with atomic labels by inserting additional vertices, or by taking advantage of edge labelling.

The algorithm described here is insufficient for formal structures that require more powerful graph formalisms like directed acyclic graphs.

The relations *parent*, *child* and *sibling* are taken here in their ordinary sense in discussing trees. In Figure 1, the vertex labeled *adv* is a *child* of the vertex labeled *smain*, the *parent* of the vertex labeled *ook*, and a sibling of the vertex labeled *verb* and the two vertices labeled *np*. To simplify definitions, the operator $label(x)$ will indicate the label of vertex or edge x.

An *induced unordered subtree* is a connected subset of the vertices of some tree that preserves the vertex and edge labels and the parent-child relations of that tree but need not preserve the ordering of siblings. Given a fully-labeled tree $T := \langle V_T, E_T, L_{V_T}, L_{E_T} \rangle$, an *induced subtree* S of T is

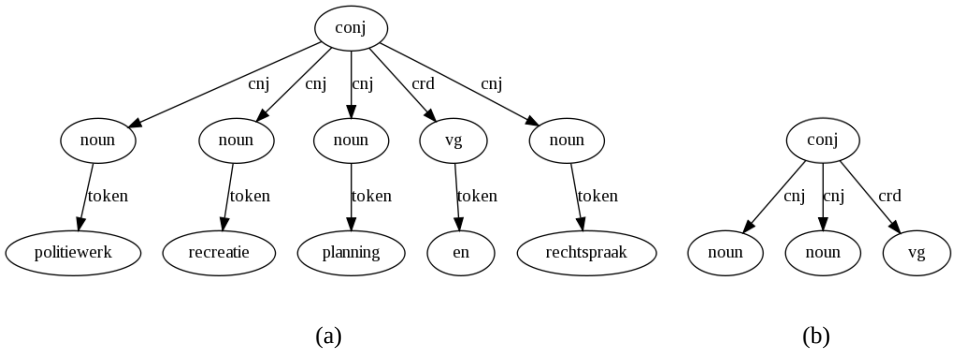


Figure 4: In 4(a) is a Dutch phrase conjoining multiple nouns. It translates as *"police work, recreation, planning and court activities"*. 4(b) has six unique unordered alignments with 4(a).

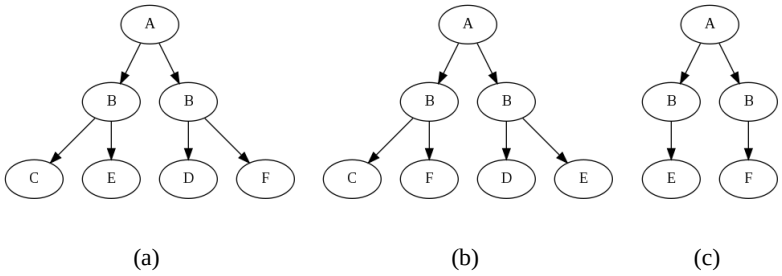


Figure 5: Subtree 5(c) is an unordered subtree of both 5(a) and 5(b), but the algorithm described in Martens (2009a) is unable to capture this in all cases.

a fully-labeled tree $S := \langle V_S, E_S, L_{V_S}, L_{E_S} \rangle$ for which there is an injection $M : V_S \rightarrow V_T$ from the vertices of S to some subset of the vertices of T, and for which:

$\forall v \in V_S$:

a. $label(v) = label(M(v))$
b. $e = \langle parent(v), v \rangle \in E_S \rightarrow$
   $e' = \langle M(parent(v)), M(v) \rangle \in E_T$
c. $label(e) = label(e')$

See Figures 1 and 2 for examples of subtrees of a particular tree.

We will further define all subtrees that are identical except in the ordering of their vertices to be *unordered isomorphic*. If a tree T is a subtree of tree T', we will follow set notation by denoting this relation as $T \subseteq T'$.

4.1 Canonical Ordering

Using canonical orderings to solve frequent unordered subtree problems was first proposed in Luccio et al. (2001) and expanded by other researchers in frequent subtree discovery techniques, notably in Chi et al. (2005b). Since the *Apriori*-style approaches described in Section 3.1 are suited only to finding subtrees whose vertices appear in a particular order, this paper will describe a mechanism for converting fully-labeled trees into canonical forms that guarantee that all instances of any unordered subtree will have an identical order to their vertices.

We must first define a strict total ordering over vertex and edge labels. Given lexica for the edge and vertex labels, $\mathbb{L}_E$ and $\mathbb{L}_V$ respectively, we define a strict total ordering on each such that $\forall l_i, l_j \in \mathbb{L}$ either $l_i \prec l_j$ or $l_i \succ l_j$ or $l_i = l_j$ and if $l_i \prec l_j$ and $l_j \prec l_k$, then $l_i \prec l_k$.

In a collection of fully-labeled trees, every vertex v that is not the root of some tree can be associated with a *full label* which is the pair $fullLabel(v) = \langle label(\langle parent(v), v \rangle), label(v) \rangle$, containing the label of the edge leading to its parent and the label of the vertex itself. For any pair of vertices where the edge to their parent is different, we

order the vertices by the order of those edges. Where the edges are the same, we order them by the ordering of their vertex labels. Where we have two sibling vertices v_i and v_j such that $fullLabel(v_i) = fullLabel(v_j)$, we recursively order the descendants of v_i and v_j, and then compare them. In this way, two nodes can only have an undefined order if they have both exactly the same full labels and identical descendants.

A *canonically ordered tree* is a tree $T := \langle V_T, E_T, L_{V_T}, L_{E_T} \rangle$, where for each $v \in V_T$, the children of v are ordered in just that fashion.

4.2 Condensed trees

A *condensed tree* is a fully-labeled tree $T := \langle V_T, E_T, L_{V_T}, L_{E_T} \rangle$ with two additional properties:

a. Each vertex $v \in V$ is associated to a list of indices $parentIndex(v) = \{i_1, i_2, ..., i_n\}$, which we will call its *parent index*. Each entry $i_1, i_2, ..., i_n$ is a non-negative integer.
b. No vertex $v \in V$ has two children with the same full label.

Condensed trees are constructed from non-condensed trees as follows:

Given a tree $T := \langle V, E, L_V, L_E \rangle$, we first canonically order it, as described in the previous section. Then, we attach a parent index to each vertex $v \in V$ which is not the root of T. The initial parent index of each node consists of a single zero.

We then traverse the vertices of the now ordered tree T in breadth-first order from the the root downwards and from left to right. Given some $v_j \in V$, if it has no sibling to its right, or if the sibling to its immediate right has a different vertex label or a different edge label on the edge to its parent, we do nothing. Otherwise, if v_j has a sibling to its immediate right v_i with the same full label, we set ℓ_i to the size of $parentIndex(v_i)$, and then we append the $parentIndex(v_j)$ to $parentIndex(v_i)$. Then, we take the children of v_j, and for each one, we increment each value in its parent index by ℓ_i, and then insert it under v_i as one of v_i's children. We delete v_j and then we reorder the children of v_i into the canonical order defined in Section 4.1.

This is performed in breadth-first order over T. The result is guaranteed to be a tree where each vertex never has two children with the same edge and vertex labels. Figure 6 shows how the trees in Figure 5 look after they are converted into condensed trees. We will denote condensed trees as $\mathfrak{T} = cond(T)$, to indicate that $\mathfrak{T}$ has been constructed from T.

If two non-condensed trees are unordered isomorphic, then their condensed forms will be identical, including in their vertex orderings and parent indexes. If two condensed trees are identical, then the non-condensed trees from which they are constructed are always unordered isomorphic.

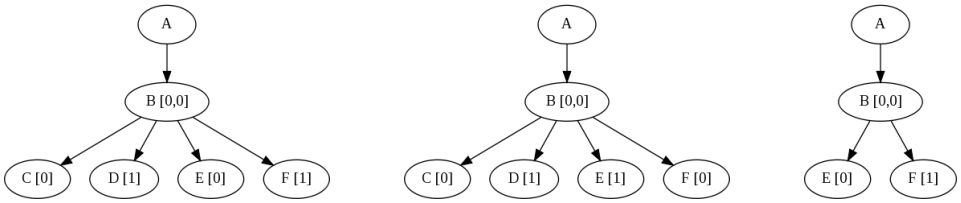
Each vertex $\mathfrak{v}$ of a condensed tree $\mathfrak{T} = cond(T)$ has a parent index containing some number of entries corresponding to a set of vertices in non-condensed tree T. We will designate that set as $orig(\mathfrak{v})$, a subset of the vertices in T. Given a condensed tree vertex $\mathfrak{v}$ and its parent $\mathfrak{p}$, if the size of $orig(\mathfrak{p})$ is larger than one, then the vertices in $\mathfrak{v}$ may have different parents in T. We can interpret the integers in the parent index of each condensed tree vertex as indicating which parent each member of $orig(\mathfrak{v})$ has.

In this way, given $\mathfrak{T} = cond(T)$, there is a one-to-one mapping from the vertices of T to a pair $\langle \mathfrak{v}, i \rangle$ consisting of some vertex in $\mathfrak{T}$ and an index to an entry in its parent index. If some vertex v in T maps to $\langle \mathfrak{v}, i \rangle$, then all the children of v, $c \in children(v)$ map to pairs $\langle \mathfrak{c}, j \rangle$ such that $parent(\mathfrak{c}) = \mathfrak{v}$ and the jth entry in $parentIndex(\mathfrak{c})$ is i. We can use this to define parent-child operations over condensed trees that perfectly match parent-child operations in non-condensed ones.

We will define a *skeleton tree* as a condensed tree stripped of its parent indices, and denote it as $skel(\mathfrak{T})$. Note that for any non-condensed tree T and any non-condensed subtree $S \subseteq T$, $skel(cond(T))$ will always contain $skel(cond(S))$ as an *ordered* subtree, including in cases like Figure 5, as shown in Figure 6.

4.3 Alignment

An alignment of a condensed subtree $\mathfrak{S}$ with a condensed tree $\mathfrak{T}$ has two parts:



(a) (b) (c)

Figure 6: The trees in Figure 5 transformed into their condensed equivalents, with their parent arrays. Note that 6(c) is visibly an ordered subtree of both 6(a) and 6(b) if you ignore the parent arrays.

a. Skeleton Alignment:
   An injection $M : V_{\mathfrak{S}} \rightarrow V_{\mathfrak{T}}$ from the vertices of $\mathfrak{S}$ to the vertices of $\mathfrak{T}$.
b. Index Alignment:
   For each vertex $v_{\mathfrak{S}} \in V_{\mathfrak{S}}$, a bipartite mapping from the vertices in $orig(v_{\mathfrak{S}})$ to the vertices in $orig(M(v_{\mathfrak{S}}))$.

The first part is an alignment of $skel(\mathfrak{S})$ with $skel(\mathfrak{T})$. Given an alignment from the root of $\mathfrak{S}$ to some vertex in $\mathfrak{T}$, this can be performed in time proportionate, in the worst case, to the number of vertices in $skel(\mathfrak{T})$. If all the parent indices of the aligned vertices in the subtree and supertree have only one index in them, then the index alignment is trivial and the alignment of $\mathfrak{S}$ to $\mathfrak{T}$ is complete.

In other cases, index alignment is non-trival. The method here draws on the procedure for unordered subtree alignment proposed by Kilpeläinen (1992). In the worst case, it resolves to the same algorithm, but can perform better on the average because of the structure of condensed trees.

Alignment proceeds from the bottom-up, starting with the leaves of $\mathfrak{S}$. If vertex $\mathfrak{s}$ is a leaf of $\mathfrak{S}$ and is aligned to some vertex $\mathfrak{t}$ in $\mathfrak{T}$, then we initially assume any member of $orig(\mathfrak{s})$ can map to any member of $orig(\mathfrak{t})$. We then proceed upwards in $\mathfrak{S}$, checking each vertex $\mathfrak{s}$ in $\mathfrak{S}$ to find a mapping from $orig(\mathfrak{s})$ to $orig(\mathfrak{t})$ such that if some $s \in orig(\mathfrak{s})$ can be mapped to some $t \in orig(\mathfrak{t})$, then the children of s can be mapped to children of t.

Once we reach the root of $\mathfrak{S}$, we proceed back downwards, removing those mappings from each $orig(\mathfrak{s})$ to its corresponding $orig(\mathfrak{t})$ that are impossible because their parents do not align.

The remaining index alignments must still be checked to verify that each one can form a part of a one-to-one mapping from $orig(\mathfrak{s})$ to $orig(\mathfrak{t})$. This is equivalent to finding a maximal bipartite matching from $orig(\mathfrak{s})$ to $orig(\mathfrak{t})$ for each possible alignment from $orig(\mathfrak{s})$ to $orig(\mathfrak{t})$. Bipartite matching is a problem with a number of well-documented solutions. (Dijkstra (1959), Lovász (1986), among others)

5 Algorithm

Having outlined condensed trees and how to align them, we can build an algorithm for extracting all frequent closed unordered subtrees from a treebank of condensed trees, given a minimum frequency threshold θ. Space restrictions preclude a full formal description of the algorithm, but it closely follows the general outline for closed tree discovery schemes advanced by Chi et al. (2005a):

1. Pass through the treebank collecting all the subtrees that consist of a single vertex label and all their locations.
2. Remove those that appear less than θ times.
3. Loop over each remaining subtree, aligning it to each place it appears in the treebank
4. Collect all the possible extensions, creating a new list of two vertex subtrees and all their locations.
5. Use the extensions to the left of the rightmost vertex in each alignment to check if the subtree is closed to the left, and reject it if it is not.
6. Use the extensions to the right of the rightmost vertex to check if the subtree is closed to the right, and output it if it is.

7. Retain the extensions to the right of the rightmost vertex and their locations if those extensions appear at least θ times.
8. Repeat for those subtrees.

6 Implementation and Performance

The *Varro* toolkit implements condensed trees and the algorithm described above in Python 3.1 and has been applied to treebanks as large as several hundred thousand sentences. The software and source code is available from sourceforge.net[3] and includes a small treebank of parsed Latin texts provided by the Perseus Digital Library. (Bamman and Crane, 2007)

The worst case memory performance of this algorithm is $O(nm)$ where n is the number of vertices in the treebank and m is the largest frequent subtree found in it. However, only the most pathologically structured treebank could come close to this ceiling, and in practice, the current implementation has so far never used as much twice the memory required to store the original treebank.

The runtime performance is, as described in Section 3.2, proportionate to the size of the output. However, aligning each occurrence of each subtree adds an additional factor. Given a condensed subtree $\mathfrak{S}$ and its condensed supertree $\mathfrak{T}$ containing $size(\mathfrak{T})$ vertices, and one already aligned vertex, the worst case alignment time is $O(size(\mathfrak{T})^{2.5})$, but only a highly pathological tree structure would approach this. The best case alignment time is $O(size(\mathfrak{S}))$. Therefore, it always takes more time to align larger subtrees, and since larger subtrees are less frequent than smaller ones, setting lower minimum frequency thresholds increases the average time required to process a subtree.

Processing even the small Alpino Treebank produces very large numbers of frequent closed subtrees. After removing punctuation and the tokens themselves, leaving just parts-of-speech and consituency labels - the Alpino treebank's 7137 sentences are reduced to 206,520 vertices. Within this small set, *Varro* took 1252 seconds to find 7307 frequent closed subtrees that appear at least 100 times. This is both considerably more subtrees than reported by Martens (2009b) on the same data and considerably more time.

Speed and memory performance are the major practical issues in this line of research. Choosing to design *Varro* with memory footprint minimization in mind is a source of some performance bottlenecks. Using Python also takes a heavy toll on speed and a C++ implementation is planned. The fast alignment-free closure checking scheme in Martens (2009b) can also be implemented using condensed trees. On small treebanks this will improve speed without loss of precision, but has limited applicability to large treebanks.

7 Conclusions

The trade-off between memory usage, run-time and completeness for this kind of algorithm is *punitive*. The user must balance *very* long runtimes against excessive memory usage if they want to accurately count all frequent unordered induced subtrees. The *Varro* toolkit is designed to make it possible to choose what tradeoffs to make. Since any subtree can be extended and checked for closure independently of other subtrees, *Varro* can easily implement heuristics designed to further reduce the number of subtrees extracted. We believe the future of this line of research lies in large part in that direction and hope that public release of *Varro* will aid in its development.

We have also discovered that there is a very strong relationship between the concision and consistency of linguistic formalisms and *Varro*'s performance. We restructured the Alpino data by promoting the head of each constituent, creating dependency-style trees along the lines described by Tesnière (1959) and Mel'čuk (1988). This reduced the number of subtrees found by 50%-60% and reduced run-times consistently by 60%-70% across a range of minimum frequency thresholds and treebank sizes. As a general rule, increasing the degree of linguistic abstraction increases the number of frequent subtrees, and consequently slows *Varro* down dramatically. Identifying linguistic formalisms that lend themselves to efficient and productive subtree discovery is another significant direction for this research, and one with immediate impact on other areas in linguistics.

[3] http://varro.sourceforge.net/

References

Agrawal, Rakesh, Tomasz Imielinski and Arun Swami. 1993. Mining association rules between sets of items in large databases. *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, pp. 207–216.

Asai, Tatsuya, Kenji Abe, Shinji Kawasoe, Hiroki Arimura, Hiroshi Sakamoto and Setsuo Arikawa. 2002. Efficient substructure discovery from large semi-structured data. *Proceedings of the Second SIAM International Conference on Data Mining*, 158–174.

Bamman, David and Gregory Crane. 2007. The Latin Dependency Treebank in a Cultural Heritage Digital Library. *Proceedings of the Workshop on Language Technology for Cultural Heritage Data*, LaTeCH 2007: pp. 33–40. http://nlp.perseus.tufts.edu/syntax/treebank/

Boulicaut, J.-F. and A. Bykowski. 2000. Frequent closures as a concise representation for binary data mining. *Knowledge discovery and data mining: current issues and new applications*, PAKDD 2000: pp. 62–73.

Chi, Yun, Richard R. Muntz, Siegfried Nijssen and Joost N. Kok. 2004. Frequent Subtree Mining - An Overview. *Fundamenta Informaticae*, 66(1-2):161–198.

Chi, Yun, Yi Xia, Yirong Yang and Richard R. Muntz. 2005a. Mining Closed and Maximal Frequent Subtrees from Databases of Labeled Rooted Trees. *IEEE Transactions on Knowledge and Data Engineering*, 17(2):190–202.

Chi, Yun, Yi Xia, Yirong Yang and Richard R. Muntz. 2005b . Canonical forms for labelled trees and their applications in frequent subtree mining. *Knowledge and Information Systems*, 8(2):203–234.

Dijkstra, E. W. 1959. A note on two problems in connexion with graphs. *Numerische Mathematik* 1:269–271.

Firth, J.R. 1957. *Papers in Linguistics*. London: OUP.

Harris, Roy and Talbot J. Taylor. 1989/1997. Varro on Linguistic Regularity. In *Harris and Taylor, Landmarks in Linguistic Thought I: The Western Tradition from Socrates to Saussure*. 2nd ed. London: Routledge. pp. 47-59.

Kilpeläinen, Pekka. 1992. *Tree Matching Problems with Applications to Structured Text Databases.* PhD dissertation. Univ. Helsinki, Dept. of Computer Science.

Knight, Kevin. 2007. Capturing practical natural language transformations. *Machine Translation*, 21:121–133.

Knight, Kevin and Graehl, Jonathan. 2005. An Overview of Probabilistic Tree Transducers for Natural Language Processing. *Proceedings of the 6th CICLing*, 1–24.

Koehn, Philipp. 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. *Proceedings of the 10th Machine Translation Summit*, 79–86.

Lovász, László and M.D. Plummer. 1986. *Matching Theory*. Amsterdam: Elsevier Science.

Luccio, Fabrizio, Antonio Enriquez, Pablo Rieumont and Linda Pagli. 2001. *Exact Rooted Subtree Matching in Sublinear Time*. Università Di Pisa Technical Report TR-01-14.

Moschitti, Alessandro. Making tree kernels practical for natural language learning. *Proceedings of the 11th Conference of the European Association for Computational Linguistics* (EACL 2006), 113–120.

Mel'čuk, Igor A. 1988. *Dependency syntax: Theory and practice*. Albany, NY: SUNY Press.

Martens, Scott. 2009a. Frequent Structure Discovery in Treebanks. *Proceedings of the 19th Computational Linguistics in the Netherlands* (CLIN 19).

Martens, Scott 2009b. Quantitative analysis of treebanks using frequent subtree mining methods. *Proceedings of the 2009 Workshop on Graph-based Methods for Natural Language Processing (TextGraphs-4)*, 84–92.

Rohde, Douglas. 2001. *Tgrep2 User Manual*. http://tedlab.mit.edu/∼dr/Tgrep2

Sinclair, John. 1991. *Corpus, Concordance, Collocation*. Oxford: OUP.

Tesnière, Lucien. 1959. *Éléments de syntaxe structurale*. Paris: Éditions Klincksieck.

van Noord, Gertjan. 2006. At last parsing is now operational. *Verbum Ex Machina. Actes de la 13e conférence sur le traitement automatique des langues naturelles* (TALN6), 20–42.

Mohammed J. Zaki. 2002. Efficiently mining frequent trees in a forest. *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1021–1035.

Instance Sense Induction from Attribute Sets

Ricardo Martin-Brualla
Google Inc
rmbrualla@gmail.com

Enrique Alfonseca
Google Inc
ealfonseca@google.com

Marius Pasca
Google Inc
mars@google.com

Keith Hall
Google Inc
kbhall@google.com

Enrique Robledo-Arnuncio
Google Inc
era@google.com

Massimiliano Ciaramita
Google Inc
massi@google.com

Abstract

This paper investigates the new problem of automatic sense induction for instance names using automatically extracted attribute sets. Several clustering strategies and data sources are described and evaluated. We also discuss the drawbacks of the evaluation metrics commonly used in similar clustering tasks. The results show improvements in most metrics with respect to the baselines, especially for polysemous instances.

1 Introduction

Recent work on information extraction increasingly turns its attention to the automatic acquisition of open-domain information from large text collections (Etzioni et al., 2008). The acquired information typically includes instances (e.g. *barack obama* or *hillary clinton*), class labels (e.g. *politician* or *presidential candidate*) and relations and attributes of the instances (e.g. *president-country* or *date-of-birth*) (Sekine, 2006; Banko et al., 2007).

Within the larger area of relation extraction, the acquisition of instance attributes (e.g. *president* for instances of countries, or *side effects* for instances of drugs) plays an important role, since attributes may serve as building blocks in any knowledge base constructed around open-domain classes of instances. Thus, a variety of attribute extraction methods mine textual data sources ranging from unstructured (Tokunaga et al., 2005) or structured (Cafarella et al., 2008) text within Web documents, to human-compiled encyclopedia (Wu et al., 2008; Cui et al., 2009) and Web search query logs (Paşca and Van Durme, 2007), attempting to extract, for a given class, a ranked list of attributes that is as comprehensive and accurate as possible.

Previous work on attribute extraction, however, does not capture or address attributes of polysemous instances. An instance may have different meanings, and the extracted attributes may not apply to all of them. For example, the most salient meanings of *darwin* are the scientist *Charles Darwin*, an Australian city, and an operating system, plus many less-known meanings. For these ambiguous instances, it is common for the existing procedures to extract mixed lists of attributes that belong to incompatible meanings, e.g. {*biography, population, hotels, books*}.

This paper explores the problem of automatically inducing instance senses from the learned attribute lists, and describes several clustering solutions based on a variety of data sources. For that, it brings together research on attribute acquisition and on word sense induction. Results show that we can generate meaninful groupings of attributes for polysemous instance names, while not harming much the monosemous instance names by generating unwanted clusters for them. The results are much better than for a random baseline, and are superior to the one-in-all and the all-singleton baselines.

2 Previous Work

Previous work on **attribute extraction** uses a variety of types of textual data as sources for mining attributes. Some methods take advantage of structured and semi-structured text available within Web documents. Examples of this are the use of markup information in HTML documents to ex-

Coling 2008: Poster Volume, pages 819–827,
Beijing, August 2010

tract patterns and clues around attributes (Yoshinaga and Torisawa, 2007; Wong and Lam, 2009; Ravi and Paşca, 2008), or the use of articles within online encyclopedia as sources of structured text for attribute extraction (Suchanek et al., 2007; Nastase and Strube, 2008; Wu and Weld, 2008). Regarding unstructured text in Web documents, the method described in (Tokunaga et al., 2005) takes various class labels as input, and applies manually-created lexico-syntactic patterns to document sentences to extract candidate attributes ranked using several frequency statistics. In (Bellare et al., 2007), the extraction is guided by a set of seed instances and attributes rather than handcrafted patterns, with the purpose of generating training data and extract new instance-attribute pairs from text.

Web search queries have also been used as a data source for attribute extraction, using lexico-syntactic patterns (Paşca and Van Durme, 2007) or seed attributes (Paşca, 2007) to guide the extraction, and leading to attributes of higher accuracy than those extracted with equivalent techniques from Web documents (Paşca et al., 2007).

Another related area to this work is the field of **word sense induction**: the task of identifying the possible senses of a word in a corpus using unsupervised methods (Yarowsky, 1995), as opposed to traditional disambiguation methods which rely on the availability of a finite and static list of possible meanings. In (Agirre and Soroa, 2007) a framework is proposed for evaluating such systems. Word sense induction can be naturally formulated as a clustering task. This introduces the complication of choosing the right number of possible senses, hence a Bayesian approach to WSI was proposed which deals with this problem within a principled generative framework (Brody and Lapata, 2009). Another related line of work is the disambiguation of people names (Mann and Yarowsky, 2003). In SEMEVAL-1, a shared task was introduced dedicated to this problem, the Web People Search task (Artiles et al., 2007; Artiles et al., 2009). Disambiguating names is also often approached as a clustering problem. One challenge shared by word sense induction and name disambiguation (and most unsupervised settings), is the evaluation. In both tasks, simple baselines such as predicting one single cluster tend to outperform more sophisticated approaches (Agirre and Soroa, 2007; Artiles et al., 2007).

Turkey Attributes		Darwin Attributes	
maps$_1$	capital$_1$	maps$_1$	definition$_{1,3}$
recipes$_2$	culture$_1$	awards$_2$	jobs$_1$
pictures$_{1,2}$	history$_1$	shoes$_1$	tourism$_1$
calories$_2$	tourism$_1$	evolution$_3$	biography$_3$
facts$_{1,2}$	nutrition facts$_2$	theory$_3$	attractions$_1$
nutrition$_2$	beaches$_1$	weather$_1$	hotels$_1$
cooking time$_2$	brands$_2$	pictures$_{1,3}$	ports$_4$
religion$_1$	language$_1$	quotes$_3$	population$_1$

Table 1: Attributes extracted for the instances *Turkey* and *Darwin*.

3 Instance Sense Induction

3.1 Problem description

This paper assumes the existence of an attribute extraction procedure. Using those attributes, our aim is to identify the coarse-grained meanings with which each attribute is associated. As an example, Table 1 shows the top 16 attributes extracted using the procedure described in (Paşca and Van Durme, 2007). Salient meanings for *turkey* are the country name (labeled as 1 in the table), and the bird name (labeled as 2). Some attributes are applicable to both meanings (*pictures* and *facts*). The second example, *darwin*, can refer to a city (sense 1), the Darwin Awards (sense 2), the person (sense 3), and an operating system (sense 4).

Examples of applications that need to discriminate between the several meanings of instances are user-facing applications requiring the attributes to be organized logically and information extraction pipelines that depend on the extracted attributes to find values in documents.

The problem we are addressing is the automatic induction of instance senses from the attribute sets, by grouping together the attributes that can be applied to a particular sense. As in related work on sense induction (Agirre and Soroa, 2007; Artiles et al., 2007), we approach this as a clustering problem: finding the right similarity metrics and clustering procedures to identify sets of related attributes in an instance. We propose a clustering based on the Expectation-Maximization (EM) algorithm (Dempster et al., 1977), exploring different parameters, similarity sources, and prior distributions.

3.2 Instance and attributes input data

The input data of instances and attributes has been obtained, in a fully automated way, following the method described in (Paşca and Van Durme, 2007). The input dataset is a set of fully anonymized set of English queries submitted to a popular (anonymized) search engine. The set contains millions of unique isolated, individual queries that are independent from one another. Each query is accompanied by its frequency of occurrence in the query logs. The sum of frequencies of all queries in the dataset is hundreds of millions. Other sources of similar data are available publicly for research purposes (Gao et al., 2007). This extraction method applies a few patterns (e.g., *the* $\mathcal{A}$ *of* $\mathcal{I}$, or $\mathcal{I}$*'s* $\mathcal{A}$, or $\mathcal{A}$ *of* $\mathcal{I}$) to queries within query logs, where an instance $\mathcal{I}$ is one of the most frequent 5 million queries from the repository of isolated queries, and $\mathcal{A}$ is a candidate attribute. For each instance, the method extracts ranked lists containing zero, one or more attributes, along with frequency-based scores. For this work, only the top 32 attributes of each instance were used, in order to have an input set for the clustering with a reasonable size, but to keep precision at high levels.

3.3 Per-attribute clustering information

For each (instance, attribute) pair, the following information is collected:

Search results: The top 20 search results (including titles and snippets) returned by a popular search engine for a query created by concatenating the instance and the attribute. The motivation for this data source is that the attributes that refer to the same meaning of the instance should help the search engine in selecting web pages that refer to that meaning. The titles and snippets of these search results are expected to contain other terms related to that meaning. For example, for the queries *[turkey maps]* and *[turkey culture]* the search results will contain information related to the country, whereas *[turkey recipes]* and *[turkey nutritional value]* should share many terms about the poultry.

Query sessions: A query session is a series of queries submitted by a single user within a small range of time (Silverstein et al., 1999). Information stored in the session logs may include the text of the queries and metadata, such as the time, the type of query (e.g., using the normal or the advance form), and user settings such as the Web browser used (Silverstein et al., 1999).

For each (instance, attribute) pair:

- Retrieve all the sessions that contained the query *[instance attribute]*.
- Collect the set of all the queries that appeared in the same session and which are a superstring of *instance*.
- Remove *instance* from each of those queries, and output the resulting set of query words.

Figure 1: Algorithm to collect session phrases associated to attributes.

Users often search for related queries within a session: queries on the *culture* of the country Turkey will tend to be surrounded by queries about topics related to the country; similarly, queries about turkey *recipes* will tend to be surrounded by other queries on recipes. Therefore, if two attributes refer to the same meaning of the instance, the distributions of terms that co-occur with them in the same search sessions is expected to be similar. To ensure that the user did not change intent during the session, we also require the queries from which we extract phrases to contain the instance of interest. The pseudocode of the procedure is shown in Figure 1.

Class labels: As described in (Paşca and Van Durme, 2008), we collect for each instance (e.g., *turkey*), a ranked list of class labels (e.g., *country, location, poultry, food*). The procedure uses a collection of Web documents and applies some IsA extraction patterns selected from (Hearst, 1992). Using the (instance, ranked-attributes) and the (instance, ranked-class labels) lists, it is possible to aggregate the two datasets to obtain, for each attribute, the class labels that are most strongly associated to it (Figure 2).

3.4 EM clustering

We run a set of EM clusterings separately for the attributes of each instance. The model implemented is the following: given an instance, let $\mathcal{A} = \{a_1, a_2, ..., a_n\}$ be the set of attributes associated with that instance. Let $\mathcal{T}$ be the vocabulary for the terms found in the search results, $\mathcal{S}$ the vocabulary of session log terms co-occurring with

For each attribute:

- Collect all the instances that contain that attribute.
- For each class label, average its ranks for those instances. If an instance does not contain a particular class label, use as rank the size of the longest list of class labels plus one.
- Rank the class labels from smaller to larger average rank.

Figure 2: Algorithm to collect class labels associated to attributes.

the attribute, and $\mathcal{C}$ be the set of all the possible class labels. Let $\mathcal{K}$ be the cluster function which assigns cluster indexes to the attributes.

We assume that the distributions for snippet terms, session terms and class labels are conditionally independent given the clustering. Furthermore, we assume that the distribution of terms for queries in a cluster are also conditionally independent given the cluster assignments:

$$p_\theta(\mathcal{T}|\mathcal{K},\mathcal{A}) \approx \prod_j p_\theta(t_j|\mathcal{K},\mathcal{A})$$
$$p_\theta(\mathcal{S}|\mathcal{K},\mathcal{A}) \approx \prod_k p_\theta(s_k|\mathcal{K},\mathcal{A})$$
$$p_\theta(\mathcal{C}|\mathcal{K},\mathcal{A}) \approx \prod_l p_\theta(c_l|\mathcal{K},\mathcal{A})$$

The clustering model for each instance (the expectation step) is, therefore:

$$p_\theta(\mathcal{K}\mathcal{T}\mathcal{S}\mathcal{C}|\mathcal{A},\Theta) =$$
$$\prod_i^N p_\theta(\mathcal{K}|\mathcal{A})p_\theta(\mathcal{T}|\mathcal{K},\mathcal{A})p_\theta(\mathcal{S}|\mathcal{K},\mathcal{A})p_\theta(\mathcal{C}|\mathcal{K},\mathcal{A})$$

To estimate the parameters of the model, we must be able to estimate the following distributions during the maximization step:

- $p_\theta(t_j|\mathcal{K},\mathcal{A}) = \frac{E_\theta(t_j,\mathcal{K}|\mathcal{A})}{E_\theta(\mathcal{K}|\mathcal{A})}$
- $p_\theta(s_k|\mathcal{K},\mathcal{A}) = \frac{E_\theta(s_k,\mathcal{K}|\mathcal{A})}{E_\theta(\mathcal{K}|\mathcal{A})}$
- $p_\theta(c_l|\mathcal{K},\mathcal{A}) = \frac{E_\theta(c_l,\mathcal{K}|\mathcal{A})}{E_\theta(\mathcal{K}|\mathcal{A})}$

One advantage of this approach is that it allows using a subset of the available data sources to evaluate their relative influence on the clustering quality. In the experiments we have tried all possible combinations of the three data sources to find the settings that give the best results.

3.5 Initialization strategies

The initial assignment of attributes to clusters is important, since a bad seed clustering can lead EM to local optima. We have tried the following two strategies:

Random assignment: the attributes are assigned to clusters randomly. To make the results repeatable, for each instance we use the instance name as the seed for the random number generator.

K-means: the initial assignments of attributes to clusters is performed using K-means. In this model, we use a simple vector-space-model in the following way:

1. Each attribute is represented with a bag-of-words of the snippets of the search results for a concatenation of the instance name and the attribute. This is the same data already collected for EM.
2. Each of the snippet terms in these bag-of-words is weighted using the $tf \times idf$ score, with inverse document frequencies estimated from an English web corpus with hundreds of millions of documents.
3. The cosine of the angle of the vectors is used as the similarity metric between each pair of attributes.

Several values of K have been tried in our experiments, as mentioned in Section 4.

3.6 Post-processing

EM works with a fixed set of clusters. In order to decide which is the optimal number of clusters, we have run all the experiments with a number of clusters K that is large enough to accommodate most of the queries in our dataset, and we run a post-processing step that merges clusters for instances that have less than K meanings.

Since we have, for each attribute, a distribution of the most likely class labels (Section 3.3), the post-processing performs as follows:

1. Generate a list of class labels per cluster, by combining the ranked lists of per-attribute class labels as was done in Section 3.3.
2. Merge together all the clusters such that their sets of top k class labels are the same.

The values of K and k are chosen by doing several runs with different values on the development set, as described in Section 4.

4 Evaluation and Results

4.1 Evaluation metrics

There does not exist a fully agreed evaluation metric for clustering tasks in NLP (Geiss, 2009; Amigó et al., 2009). Each metric has its own idiosyncrasies, so we have chosen to compute six different evaluation metrics as described in (Amigó et al., 2009). Empirical results show they are highly correlated, i.e., tuning a parameter by hill-climbing on F-score typically also improves the B^3 F-score.

Purity (Zhao and Karypis, 2002): Let C be the clusters to evaluate, L the set of categories (the clusters in the gold-standard), and N the number of clustered items. Purity is the average of the precision values: Purity $= \sum_i \frac{|C_i|}{N} \max_j \text{Prec}(C_i, L_j)$, where the precision for cluster C_i with respect to category L_j is $\text{Prec}(C_i, L_j) = \frac{|C_i \cap L_j|}{|C_i|}$. Purity is a precision metric. Inverting the roles of the categories L and the clusters C gives a recall metric, **inverse purity**, which rewards grouping items together. The two metrics can be combined in an F-score.

B^3 Precision (Bagga and Baldwin, 1998): Let $L(e)$ and $C(e)$ denote the gold-standard-category and the cluster of an item e. The correctness of the relation between e and other element e' is defined as

$$\text{Correctness}(e, e') = \begin{cases} 1 & \text{iff} L(e) = L(e') \Leftrightarrow C(e) = C(e') \\ 0 & \text{otherwise} \end{cases}$$

The B^3 Precision of an item is the proportion of items in its cluster which belong to its category, including itself. The total precision is the average of the item precisions: B^3 Prec $= \text{avg}_e[\text{avg}_{e':C(e)=C(e')}\text{Correctness}(e, e')]$

B^3 Recall: is calculated in a similar way, inverting the roles of clusters and categories. The **B^3 F-score** is obtained by combining B^3 precision and B^3 recall.

4.2 Gold standards

We have built two annotated sets, one to be used as a development set for adjusting the parameters, and a second one as a test set. The evaluation settings were chosen without knowledge of the test set. Each of the two sets contains 75 instances chosen randomly from the complete set of instances with ranked attributes (Section 3.2 described the input data). For the random sampling, the instances were weighted with their frequency in the query logs as full queries, so that more frequent instances have higher chance to be chosen. This ensures that uncommon instances are not overrepresented in the gold-standard.

The annotators contributed 50 additional instances (25 for development and 25 for testing) that they considered interesting to study, e.g., because of having several salient meanings.

Five human annotators were shown the top 32 attributes for each instance, and they were asked to cluster them. We decided to start with a simplified version of the problem by considering it a hard clustering task.

Table 2 shows that the average agreement scores between judge pairs, measured with the same evaluation metrics used for the system output, are quite high. In the first three metrics, the F-score is not an average of precision and recall, but a weighted average calculated separately for each cluster, so it may have a value that is not between the values of precision and recall.

The annotated instances were classified as monosemous/polysemous, depending on wether or not they had more than one cluster with enough (five) attributes. This classification allows to report separate results for the whole set (where instances with just one major sense dominate) and for the subset of polysemous instances. Table 3 shows examples of polysemous instances. Exam-

Purity	Inv. Purity	F-score	B^3 Precision	B^3 Recall	B^3 F-score
0.94	0.95	0.92	0.90	0.92	0.91

Table 2: Inter-judge agreement scores.

Polysemous	Main meanings
airplane	machine, movie
apple	fruit, company
armstrong	unit, company, person
chain reaction	company, film, band, chemistry
chf	airport, currency, heart attack
darwin	person, city
david copperfield	book, performer, movie
delta	letter, airways

Table 3: Examples of polysemous instances.

		All instances						polysemous instances					
	Weights	**Purity**	**Inv. Purity**	**F score**	**B^3 Prec.**	**B^3 Recall**	**B^3 F score**	**Purity**	**Inv. Purity**	**F score**	**B^3 Prec.**	**B^3 Recall**	**B^3 F score**
	All-in-one	0.797	1.000	0.766	0.700	1.000	0.797	0.558	1.000	0.540	0.410	1.000	0.573
	All-singletons	1.000	0.145	0.187	1.000	0.145	0.242	1.000	0.205	0.266	1.000	0.205	0.333
	Random	0.888	0.322	0.451	0.851	0.246	0.373	0.685	0.362	0.447	0.595	0.276	0.373
Random Init.	Only snippets	0.809	0.374	0.417	0.737	0.311	0.410	0.596	0.430	0.401	0.483	0.361	0.399
	Only sessions	0.797	0.948	0.728	0.700	0.944	0.753	0.558	1.000	0.540	0.410	1.000	0.573
	Only class labels	0.798	0.983	0.760	0.701	0.969	0.785	0.561	0.990	0.541	0.415	0.981	0.574
	No snippets	0.798	0.934	0.723	0.702	0.918	0.744	0.561	0.990	0.541	0.415	0.981	0.574
	No sessions	0.809	0.374	0.417	0.737	0.311	0.410	0.596	0.430	0.401	0.483	0.361	0.399
	No class labels	0.809	0.374	0.417	0.737	0.311	0.410	0.596	0.430	0.401	0.483	0.361	0.399
	All	0.809	0.380	0.420	0.736	0.316	0.414	0.596	0.430	0.400	0.483	0.361	0.399
K-Means Init.	Only snippets	0.844	0.765	0.700	0.771	0.654	0.675	0.671	0.806	0.587	0.556	0.719	0.611
	Only sessions	0.798	0.957	0.736	0.702	0.949	0.759	0.558	1.000	0.540	0.410	1.000	0.573
	Only class labels	0.824	0.656	0.622	0.747	0.568	0.604	0.641	0.768	0.565	0.519	0.699	0.575
	No snippets	0.824	0.655	0.622	0.748	0.562	0.598	0.640	0.768	0.565	0.518	0.698	0.574
	No sessions	0.843	0.770	0.701	0.769	0.661	0.677	0.671	0.806	0.587	0.556	0.719	0.611
	No class labels	0.844	0.762	0.698	0.771	0.651	0.673	0.671	0.806	0.587	0.556	0.719	0.611
	All	0.843	0.767	0.699	0.770	0.657	0.675	0.671	0.806	0.587	0.556	0.719	0.611

Table 4: Scores over all instances and over polysemous instances.

ples of monosemous instances are *activision*, *amctheaters*, *american airlines*, *ask.com*, *bebo*, *disney* or *einstein*. 22% of the instances in the development set and 13% of the instances in the test set are polysemous.

4.3 Parameter tuning

We tuned the different parameters of the algorithm using the development set. We performed several EM runs including all three data sources, modifying the following parameters: the smoothing ϵ added to the cluster soft-assignment in the Maximization step (Manning et al., 2008), the number K of clusters for K-Means and EM, and the number k of top ranked class labels that two clusters need to have in common in order to be merged at the post-processing step. The best results were obtained with $\epsilon = 0.4$, $K = 5$ and $k = 1$. These are the values used in the experiments mentioned from now on.

4.4 EM initialization and data sources

Table 4 shows the results after running EM over the development set, using every possible combination of data sources, and the two initialization strategies (random and K-Means). Several observations can be drawn from this table:

First, as mentioned in Section 2, the evaluation metrics are biased towards the all-in-one solution. This is worsened by the fact that the majority of the instances in our dataset are monosemous. Therefore, the highest F-scores and B^3 F-scores are obtained by the all-in-one baseline, although it is not the most useful clustering.

When using only class labels, EM tends to produce results similar to the all-in-one baseline This can be explained by the limited class vocabulary which makes most of the attributes share class labels. The bad results when using only sessions are caused by the presence of attributes with no session terms, due to insufficient data.

The random clustering baseline (third line in Table 4) tends to give smaller clusters than EM, because it distributes instances uniformly across the clusters. This leads to better precision scores, and much worse recall and F-score metrics.

From these results, we conclude that snippet terms are the most useful resource for clustering. The other data sources do not provide a significant improvement over it. The best results overall for the polysemous instances, and the highest results for the whole dataset (excluding the outliers that are too similar to the all-in-one baseline) are obtained using snippet terms. For these configurations, as we expected, the K-Means initialization does a better job in avoiding local optima during EM than the random one.

4.5 Post-processing

Table 5 includes the results on the development set after post-processing, using the best configuration for EM (K-Means initialization and snippet terms for EM). Post-processing slightly hurts the B^3 F-score for polysemous terms, but it improves results for the whole dataset, as it merges many clusters for the monosemous instances.

Data	Method	Purity	Inv. Purity	F-score	B^3 Prec.	B^3 Recall	B^3 F-score
All instances	All-in-one	0.797	1.000	0.766	0.700	1.000	0.797
	All-singletons	1.000	0.145	0.187	1.000	0.145	0.242
	K-Means + EM (snippets)	0.844	0.765	0.700	0.771	0.654	0.675
	K-Means + EM (snippets) + postprocessing	0.825	0.837	0.728	0.743	0.761	0.722
Polysemous	All-in-one	0.558	1.000	0.540	0.410	1.000	0.573
	All-singletons	1.000	0.205	0.266	1.000	0.205	0.333
	K-Means + EM (snippets)	0.671	0.806	0.587	0.556	0.719	0.611
	K-Means + EM (snippets) + postprocessing	0.644	0.846	0.592	0.518	0.777	0.607

Table 5: Scores only over all and polysemous instances, without and with postprocessing.

K-Means output	EM output	Post-processing
pictures, family, logo, biography inauguration, song, lyrics, foods, quotes, timeline, shoes, health care maps, art, kids, history, speeches official website, facts, scandal economy, blog, music, flag, camping	pictures, biography, inauguration song, lyrics, foods, timeline, camping, shoes, maps, art, history, official website, facts, speeches scandal, blog, music	pictures, biography, inauguration song, lyrics, goods, timeline, camping, shoes, maps, art, history official website, facts, speeches scandal, blog, music, family, kids daughters
approval rating	approval rating, health care, economy	approval rating, health care, economy
daughters	family, kids, daughters	
symbol	logo, quotes, symbol, flag	logo, quotes, symbol, definition religion, slogan, books, flag
definition, religion, slogan, books	definition, religion, slogan, books	

Table 6: Attributes extracted for the monosemous instance *obama*, using snippet terms for EM.

4.6 Clustering examples

Tables 6 and 7 show examples of clustering results for three instances chosen as representatives of the monosemous and the polysemous subsets. These show that the output of the K-Means initialization can uncover some meaningful clusters, but tends to generate a dominant cluster and a few small or singleton clusters. EM distributes the attributes more evenly across clusters, combining attributes that are closely related.

For monosemous instances like *obama*, EM generates small clusters of highly related attributes (e.g, *family*, *kids* and *daughters*). Postprocessing merges some of the clusters together, but it fails to merge all into a single cluster.

For *darwin*, two of the small clusters given by K-Means are actually good, as *ports* is the only attribute of the operating system, and *lyrics* is one of the two attributes referring to a song titled Darwin. EM again redistributes the attributes, creating two large and mostly correct clusters.

For *david copperfield*, EM creates two clusters for the performer, one for the book, one for the movie, and one for *tattoo* (off-topic for this instance). The two clusters referring to the performer are merged in the post-processing, with some errors remaining, e.g, *trailer* and *second wife* are in the wrong cluster.

4.7 Results on the test set

Table 8 show the results of the EM clustering and the postprocessing step when executed on the test set. The settings are those that produced the best results on the development set: using EM initialized with K-Means, and using only snippet terms for the generative model.

As mentioned above, the test set has a higher proportion of monosemous queries than the development set, so the all-in-one baseline produces better results than before. Still, we can see the same trend happening: for the whole dataset the F-score metrics are somewhat worse than the best baseline, given that the evaluation metrics all overvalue the all-in-one baseline, but this can be considered an artifact of the metrics. As with the development set, using EM produces the best precision scores (except for the all-singletons baseline), and the postprocessing improves precision and F-score over the all-in-one baseline. The whole system improves considerably the F-score for the polysemous terms.

5 Conclusions

This paper investigates the new task of inducing instance senses using ranked lists of attributes as input. It describes a clustering procedure based on the EM model, capable of integrating differ-

Instance	K-Means output	EM output	Post-processing
Darwin	maps, shoes, logo, awards, weather pictures, quotes, definition, jobs, tourism, biography, hotels, attractions, beaches, accommodation, tv show, clothing, postcode, music facts, review, history side effects, airlines, prices, lighting	maps, shoes, logo, weather jobs, tourism hotels, attractions, beaches, accommodation, tv show, clothing, postcode, music, review side effects, airlines, prices, lighting	maps, shoes, logo, weather jobs, tourism hotels, attractions, beaches, accommodation, tv show, clothing, postcode, music, review side effects, airlines, prices, lighting
	ports	awards, ports	definition, population
	evolution, theory, books	evolution, theory, quotes pictures, biography, facts, history, books	awards, ports
	lyrics	lyrics	evolution, theory, quotes pictures, biography, facts, history, books
	population	definition, population	lyrics
David Copperfield	summary, biography, pictures, quotes, strokes, book review, tricks, tour dates, characters, lyrics, plot, synopsis, dating, logo, themes, author, filmography, cast members, official website, trailer, setting, religion	biography, pictures, quotes, strokes, tricks, tour dates, lyrics, dating, logo, filmography, cast members, official website, trailer, setting, religion	biography, pictures, girlfriend quotes, strokes, tricks, tattoo tour dates, secrets, lyrics, wives, music, dating, logo, filmography, blog, cast members, official website, trailer, setting, religion
	house, reviews	book review, review, house, reviews	book review, review, house, reviews
	tattoo	tattoo	summary, second wife, characters, plot, synopsis, themes, author
	second wife	summary, second wife, characters, plot, synopsis, themes, author	girlfriend, secrets, wives, music, blog
	girlfriend, secrets, wives, review, music, blog	girlfriend, secrets, wives, music, blog	

Table 7: Attributes extracted for three polysemous instances, using snippet terms for EM.

Set	Solution	Purity	Inverse Purity	F-score	B^3 Precision	B^3 Recall	B^3 F-score
All	All-in-one	0.907	1.000	0.892	0.858	1.000	0.908
	All-singletons	1.000	0.076	0.114	1.000	0.076	0.136
	Random	0.936	0.325	0.463	0.914	0.243	0.377
	EM	0.927	0.577	0.664	0.896	0.426	0.561
	EM+postprocessing	0.919	0.806	0.804	0.878	0.717	0.764
Polysemous	All-in-one	0.588	1.000	0.586	0.457	1.000	0.613
	All-singletons	1.000	0.141	0.210	1.000	0.141	0.239
	Random	0.643	0.382	0.441	0.549	0.288	0.369
	EM	0.706	0.631	0.556	0.626	0.515	0.547
	EM+postprocessing	0.675	0.894	0.650	0.564	0.842	0.661

Table 8: Scores in the test set.

ent data sources, and explores cluster initialization and post-processing strategies. The evaluation shows that the most important of the considered data sources is the snippet terms obtained from search engine results to queries made by concatenating the instance and the attribute. A simple post-processing that merges attribute clusters that have common class labels can improve recall for monosemous queries. The results show improvements across most metrics with respect to a random baseline, and F-score improvements for polysemous instances.

Future work includes extending the generative model to be applied across the board, linking the clustering models of different instances with each other. We also intend to explore applications of the clustered attributes in order to perform extrinsic evaluations on these data.

References

Agirre, Eneko and Aitor Soroa. 2007. Semeval-2007 task 02: Evaluating word sense induction and discrimination systems. In *Proceedings of SemEval-2007*, pages 7–12. Association for Computational Linguistics.

Amigó, E., J. Gonzalo, J. Artiles, and F. Verdejo. 2009. A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Information Retrieval*, 12(4):461–486.

Artiles, Javier, Julio Gonzalo, and Satoshi Sekine. 2007. The semeval-2007 WePS evaluation: Establishing a benchmark for the web people search task. In *Proceedings of SemEval-2007*, pages 64–69.

Artiles, J., J. Gonzalo, and S. Sekine. 2009. Weps 2 evaluation campaign: overview of the web people search clustering task. In *2nd Web People Search Evaluation Workshop (WePS 2009), 18th WWW Conference*.

Bagga, A. and B. Baldwin. 1998. Entity-based cross-document co-referencing using the vector space model, Proceedings of the 17th international conference on Computational linguistics. In *Proceedings of ACL-98*.

Banko, M., Michael J Cafarella, S. Soderland, M. Broadhead, and O. Etzioni. 2007. Open information extraction from the Web. In *Proceedings of IJCAI-07*, pages 2670–2676, Hyderabad, India.

Bellare, K., P.P. Talukdar, G. Kumaran, F. Pereira, M. Liberman, A. McCallum, and M. Dredze. 2007. Lightly-Supervised Attribute Extraction. In *NIPS 2007 Workshop on Machine Learning for Web Search*.

Brody, Samuel and Mirella Lapata. 2009. Bayesian word sense induction. In *Proceedings of EACL '09*, pages 103–111.

Cafarella, M.J., A. Halevy, D.Z. Wang, and Y. Zhang. 2008. Webtables: Exploring the Power of Tables on the Eeb. *Proceedings of the VLDB Endowment archive*, 1(1):538–549.

Cui, G., Q. Lu, W. Li, and Y. Chen. 2009. Automatic Acquisition of Attributes for Ontology Construction. In *Proceedings of the 22nd International Conference on Computer Processing of Oriental Languages*, pages 248–259. Springer.

Dempster, A.P., N.M. Laird, D.B. Rubin, et al. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38.

Etzioni, O., M. Banko, S. Soderland, and S. Weld. 2008. Open Information Extraction from the Web. *Communications of the ACM*, 51(12), December.

Gao, W., C. Niu, J. Nie, M. Zhou, J. Hu, K. Wong, and H. Hon. 2007. Cross-lingual query suggestion using query logs of different languages. In *Proceedings of SIGIR-07*, pages 463–470, Amsterdam, The Netherlands.

Geiss, J. 2009. Creating a Gold Standard for Sentence Clustering in Multi-Document Summarization. *ACL-IJCNLP 2009*.

Hearst, M. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of COLING-92*, pages 539–545, Nantes, France.

Mann, Gideon S. and David Yarowsky. 2003. Unsupervised personal name disambiguation. In *Proceedings of HLT-NAACL 2003*, pages 33–40. Association for Computational Linguistics.

Manning, C.D., P. Raghavan, and H. Schtze. 2008. *Introduction to Information Retrieval*. Cambridge University Press New York, NY, USA.

Nastase, V. and M. Strube. 2008. Decoding wikipedia categories for knowledge acquisition. In *Proceedings of AAAI-08*, pages 1219–1224, Chicago, Illinois.

Paşca, M. and B. Van Durme. 2007. What you seek is what you get: Extraction of class attributes from query logs. In *Proceedings of IJCAI-07*, pages 2832–2837, Hyderabad, India.

Paşca, M. and B. Van Durme. 2008. Weakly-supervised acquisition of open-domain classes and class attributes from web documents and query logs. In *Proceedings of ACL-08*, pages 19–27, Columbus, Ohio.

Paşca, M., B. Van Durme, and N. Garera. 2007. The role of documents vs. queries in extracting class attributes from text. In *Proceedings of CIKM-07*, pages 485–494, Lisbon, Portugal.

Paşca, M. 2007. Organizing and searching the World Wide Web of facts - step two: Harnessing the wisdom of the crowds. In *Proceedings of WWW-07*, pages 101–110, Banff, Canada.

Ravi, S. and M. Paşca. 2008. Using Structured Text for Large-Scale Attribute Extraction. In *CIKM*. ACM New York, NY, USA.

Sekine, S. 2006. On-Demand Information Extraction. In *Proceedings of the COLING/ACL on Main conference poster sessions*, pages 731–738. Association for Computational Linguistics Morristown, NJ, USA.

Silverstein, C., H. Marais, M. Henzinger, and M. Moricz. 1999. Analysis of a very large web search engine query log. In *ACM SIGIR Forum*, pages 6–12. ACM New York, NY, USA.

Suchanek, F., G. Kasneci, and G. Weikum. 2007. Yago: a core of semantic knowledge unifying WordNet and Wikipedia. In *Proceedings of WWW-07*, pages 697–706, Banff, Canada.

Tokunaga, K., J. Kazama, and K. Torisawa. 2005. Automatic discovery of attribute words from Web documents. In *Proceedings of the 2nd International Joint Conference on Natural Language Processing (IJCNLP-05)*, pages 106–118, Jeju Island, Korea.

Wong, T.L. and W. Lam. 2009. An Unsupervised Method for Joint Information Extraction and Feature Mining Across Different Web Sites. *Data & Knowledge Engineering*, 68(1):107–125.

Wu, F. and D. Weld. 2008. Automatically refining the Wikipedia infobox ontology. In *Proceedings of WWW-08*, pages 635–644, Beijing, China.

Wu, F., R. Hoffmann, and D. Weld. 2008. Information extraction from Wikipedia: Moving down the long tail. In *Proceedings of KDD-08*, pages 731–739, Las Vegas, Nevada.

Yarowsky, David. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of ACL-95*, pages 189–196. Association for Computational Linguistics.

Yoshinaga, N. and K. Torisawa. 2007. Open-Domain Attribute-Value Acquisition from Semi-Structured Texts. In *Proceedings of the Workshop on Ontolex*, pages 55–66.

Zhao, Y. and G. Karypis. 2002. Criterion functions for document clustering. Technical report, Experiments and Analysis University of Minnesota, Department of Computer Science/Army HPC Research Center.

A Power Mean Based Algorithm for Combining Multiple Alignment Tables

Sameer Maskey, Steven J. Rennie, Bowen Zhou
IBM T.J. Watson Research Center
{smaskey, sjrennie, zhou}@us.ibm.com

Abstract

Most existing techniques for combining multiple alignment tables can combine only two alignment tables at a time, and are based on heuristics (Och and Ney, 2003), (Koehn et al., 2003). In this paper, we propose a novel mathematical formulation for combining an arbitrary number of alignment tables using their power mean. The method frames the combination task as an optimization problem, and finds the optimal alignment lying between the intersection and union of multiple alignment tables by optimizing the parameter p: the affinely extended real number defining the order of the power mean function. The combination approach produces better alignment tables in terms of both F-measure and BLEU scores.

1 Introduction

Machine Translation (MT) systems are trained on bi-text parallel corpora. One of the first steps involved in training a MT system is obtaining alignments between words of source and target languages. This is typically done using some form of Expectation Maximization (EM) algorithm (Brown et al., 1993), (Och and Ney, 2003), (Vogel et al., 1996). These unsupervised algorithms provide alignment links between english words e_i and the foreign words f_j for a given $e-f$ sentence pair. The alignment pairs are then used to extract phrases tables (Koehn et al., 2003), hierarchical rules (Chiang, 2005), or tree-to-string mappings (Yamada and Knight, 2001). Thus, the accuracy of these alignment links has a significant impact in overall MT accuracy.

One of the commonly used techniques to improve the alignment accuracy is combining alignment tables obtained for source to target ($e2f$) and target to source ($f2e$) directions (Och and Ney, 2003). This combining technique involves obtaining two sets of alignment tables A_1 and A_2 for the same sentence pair $e - f$, and producing a new set based on union $A_\cup = A_1 \cup A_2$ or intersection $A_\cap = A_1 \cap A_2$ or some optimal combination A_o such that it is subset of $A_1 \cup A_2$ but a superset of $A_1 \cap A_2$. How to find this optimal A_o is a key question. $A_\cup$ has high precision but low recall producing fewer alignments and $A_\cap$ has high recall but low precision.

2 Related Work

Most existing methods for alignment combination (symmetrization) rely on heuristics to identify reliable links (Och and Ney, 2003), (Koehn et al., 2003). The method proposed in (Och and Ney, 2003), for example, interpolates the intersection and union of two asymmetric alignment tables by adding links that are adjacent to intersection links, and connect at least one previously unaligned word. Another example is the method in (Koehn et al., 2003), which adds links to the intersection of two alignment tables that are the diagonal neighbors of existing links, optionally requiring that any added links connect two previously unaligned words.

Other methods try to combine the tables during alignment training. In (Liang et al., 2006), asymmetric models are jointly trained to maximize the similarity of their alignments, by opti-

Coling 2008: Poster Volume, pages 828–836,
Beijing, August 2010

mizing an EM-like objective function based on agreement heuristics. In (Ayan et al., 2004), the authors present a technique for combining alignments based on various linguistic resources such as parts of speech, dependency parses, or bilingual dictionaries, and use machine learning techniques to do alignment combination. One of the main disadvantages of (Ayan et al., 2004)'s method, however, is that the algorithm is a supervised learning method, and so requires human-annotated data. Recently, (Xiang et al., 2010) proposed a method that can handle multiple alignments with soft links which are defined by confidence scores of alignment links. (Matusov et al., 2004) on the other hand, frame symmetrization as finding a set with minimal cost using use a graph based algorithm where costs are associated with local alignment probabilities.

In summary, most existing alignment combination methods try to find an optimal alignment set A_o that lies between $A_\cap$ and $A_\cup$ using heuristics. The main problems with methods based on heuristics are:

1. they may not generalize well across language pairs
2. they typically do not have any parameters to optimize
3. most methods can combine only 2 alignments at a time
4. most approaches are ad-hoc and are not mathematically well defined

In this paper we address these issues by proposing a novel mathematical formulation for combining an arbitrary number of alignment tables. The method frames the combination task as an optimization problem, and finds the optimal alignment lying between the intersection and union of multiple alignment tables by optimizing the parameter p of the power mean function.

3 Alignment combination using the power mean

Given an english-foreign sentence pair (e_1^I, f_1^J) the alignment problem is to determine the presence of absence of alignment links a_{ij} between the words e_i and f_j, where $i \leq I$ and $j \leq J$. In this paper we will use the convention that when $a_{ij} = 1$, words e_i and f_j are linked, otherwise $a_{ij} = 0$. Let us define the alignment tables we obtain for two translation directions as A_1 and A_2, respectively. The union of these two alignment tables $A_\cup$ contain all of the links in A_1 and A_2, and the intersection $A_\cap$ contain only the common links. Definitions 1 and 2 below define $A_\cup$ and $A_\cap$ more formally. Our goal is to find an alignment set A_o such that $|A_\cap| \leq |A_o| \leq |A_\cup|$ that maximizes some objective function. We now describe the power mean (PM) and show how the PM can represent both the union and intersection of alignment tables using the same formula.

The power mean:
The power mean is defined by equation 1 below, where p is a real number in $(-\infty, \infty)$ and a_n is a positive real number.

$$S_p(a_1, a_2, ..., a_n) = (\frac{1}{n}\sum_{k=1}^{n} a_k^p)^{\frac{1}{p}} \quad (1)$$

The power mean, also known as the generalized mean, has several interesting properties that are relevant to our alignment combination problem. In particular, the power mean is equivalent to the geometric mean G when $p \to 0$ as shown in equation 2 below:

$$\begin{aligned} G(a_1, a_2, ..., a_n) &= (\prod_{i=1}^{n} a_i)^{\frac{1}{n}} \\ &= \lim_{p\to 0}(\frac{1}{n}\sum_{k=1}^{n} a_k^p)^{\frac{1}{p}} \quad (2) \end{aligned}$$

The power mean, furthermore, is equivalent to the maximum function M when $p \to \infty$:

$$\begin{aligned} M(a_1, a_2, ..., a_n) &= \max(a_1, a_2, ..., a_n) \\ &= \lim_{p\to \infty}(\frac{1}{n}\sum_{k=1}^{n} a_k^p)^{\frac{1}{p}} \quad (3) \end{aligned}$$

Importantly, the PM S_p is a non-decreasing function of p. This means that S_p is lower bounded by G and upper-bounded by M for $p \in [0, \infty]$:

$$G < S_p < M, \quad 0 < p < \infty. \quad (4)$$

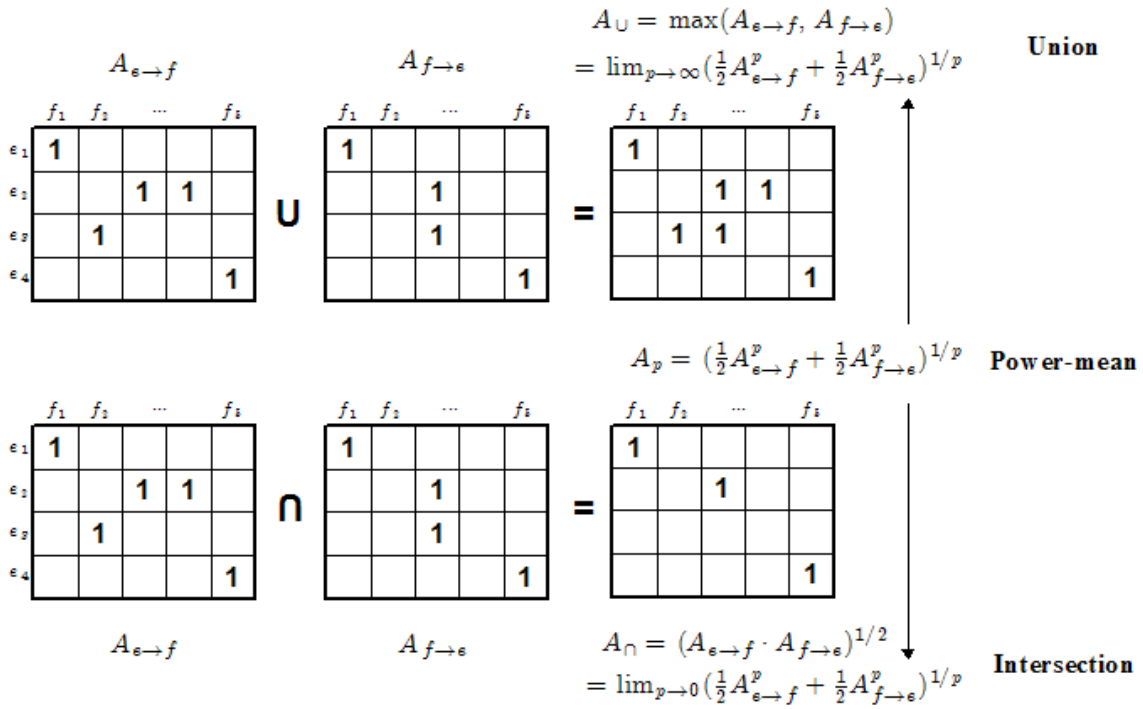


Figure 1: The power-mean is a principled way to interpolate between the extremes of union and intersection when combining multiple alignment tables.

They key insight underpinning our mathematical formulation of the alignment combination problem is that the geometric mean of multiple alignment tables is equivalent to their intersection, while the maximum of multiple alignment tables is equivalent to their union.

Let A_q be an alignment with elements a^q_{ij} such that $a^q_{ij} = 1$ if words e_i and f_j are linked, and $a^q_{ij} = 0$ otherwise. The union and intersection of a set of n alignment tables can then be formally defined as follows:

Definition 1: The union of alignments $A_1, A_2, ..., A_n$ is a set $A_\cup$ with $a^\cup_{ij} = 1$ if $a^q_{ij} = 1$ for *any* $q \in \{1, 2, ..., n\}$.

Definition 2: The intersection of alignments $A_1, A_2, ..., A_n$ is a set $A_\cap$ with $a^\cap_{ij} = 1$ if $a^q_{ij} = 1$ for *all* $q \in \{1, 2, ..., n\}$.

Figure 1 depicts a simple example of the alignment combination problem for the common case of alignment symmetrization. Two alignments tables, $A_{e\to f}$ and $A_{f\to e}$ (one-to-many alignments), need to be combined. The result of taking the union $A_\cup$ and intersection $A_\cap$ of the tables is shown. $A_\cup$ can be computed by taking the element-wise maximum of $A_{e\to f}$ and $A_{f\to e}$, which in turn is equal to the power mean A_p of the elements of these tables in the limit as $p \to \infty$. The intersection of the two tables, $A_\cap$, can similarly be computed by taking the geometric mean of the elements of $A_{e\to f}$ and $A_{f\to e}$, which is equal to the power mean A_p of the elements of these tables in the limit as $p \to 0$. For $p \in (0, \infty)$, equation 4 implies that A_p has elements with values between $A_\cap$ and $A_\cup$. We now provide formal proofs for these results when combining an arbitrary number of alignment tables.

3.1 The intersection of alignment tables $A_1..A_n$ is equivalent to their element-wise geometric mean $G(A_1, A_2, ..., A_n)$, as defined in (2).

Proof : Let $A_\cap$ be the intersection of all A_q where $q \in \{1, 2, .., n\}$. As per our definition of intersection $\cap$ between alignment tables, $A_\cap$ contains links where $a^q_{ij} = 1 \; \forall \; q$.

Let A_g be the set that contains the elements

of $G(A_1, A_2, ..., A_n)$. Then a_{ij}^g is the geometric mean of the elements a_{ij}^q where $q \in \{1, 2, .., n\}$, as defined in equation 2, that is, $a_{ij}^g = (\prod_{q=1}^{n} a_{ij}^g)^{\frac{1}{n}}$. This product is equal to 1 iff $a_{ij}^q = 1 \,\forall\, q$ and zero otherwise, since $a_{ij}^q \in \{0, 1\} \,\forall\, q$. Hence $A_g = A_\cap$. Q.E.D.

3.2 The union of alignment tables $A_1..A_n$ is equivalent to their element-wise maximum $M(A_1, A_2, ..., A_n)$, as defined in (3).

$Proof$: Let $A_\cup$ be the union of all A_q for $q \in \{1, 2, .., n\}$. As per our definition of the union between alignments $A_\cup$ has links where $a_{ij}^q = 1$ for some q.

Let A_m be the set that contain the elements of $M(A_1, A_2, ..., A_n)$. Let a_{ij}^m be the maximum of the elements a_{ij}^q where $q \in \{1, 2, .., n\}$, as defined in equation (3). The max function is equal to 1 iff $a_{ij}^q = 1$ for some q and zero otherwise, since $a_{ij}^q \in \{0, 1\} \,\forall\, q$. Hence $A_m = A_\cup$. Q.E.D.

3.3 The element-wise power mean $S_p(A_1, A_2, ..., A_n)$ of alignment tables $A_1..A_n$ has entries that are lower-bounded by the intersection of these tables, and upper-bounded by their union for $p \in [0, \infty]$.

$Proof$: We have already shown that the union and intersection of a set of alignment tables are equivalent to the maximum and geometric mean of these tables, respectively. Therefore given that the result in equation 4 is true (we will not prove it here), the relation holds. In this sense, the power mean can be used to interpolate between the intersection and union of multiple alignment tables. Q.E.D.

4 Data

We evaluate the proposed method using an English-Pashto translation task, as defined by the DARPA TransTac program. The training data for this task consists of slightly more than 100K parallel sentences. The Transtac task was designed to evaluate speech-to-speech translation systems, so all training sentences are conversational in nature. The sentence length of these utterances varies greatly, ranging from a single word to more than 50 words. 2026 sentences were randomly sampled from this training data to prepare held out development set. The held out Transtac test set consists of 1019 parallel sentences.

Method	F-measure
I	0.5979
H	0.6891
GDF	0.6712
PM	0.6984
PM$_n$	0.7276
U	0.6589

Table 1: F-measure Based on Various Alignment Combination Methods

5 Experiments and Discussion

We have shown in the previous sections that union and intersection of alignments can be mathematically formulated using the power mean. Since both combination operations can be represented with the same mathematical expression, we can search the combination space "between" the intersection and union of alignment tables by optimizing p w.r.t. any chosen objective function. In these experiments, we define the optimal alignment as the one that maximizes the objective function $f(\{a_{ijt}\}, \{\hat{a}_{ijt}\}, p)$, where f is standard F-measure, $\{\hat{a}_{ijt}\}$ is the set of all estimated alignment entries on some dataset, $\{a_{ijt}\}$ is the set of all corresponding human-annotated alignment entries, and p is the order of the power mean function. Instead of attempting to optimize the F-measure using heuristics, we can now optimize it by finding the appropriate power order p using any suitable numerical optimization algorithm. In our experiments we used the general simplex algorithm of amoeba search (Nelder and Mead, 1965), which attempts to find the optimal set of parameters by evolving a simplex of evaluated points in the direction that the F-measure is increasing.

In order to test our alignment combination formulation empirically we performed experiments on English-Pashto language with data described in Section 4. We first trained two sets of alignments, the e2f and f2e directions, based on GIZA++ (Och and Ney, 2003) algorithm. We then combined these alignments by performing intersec-

tion (I) and union (U). We obtained F-measure of 0.5979 for intersection (I), 0.6589 for union (U). For intersection the F-measure is lower presumably because many alignments are not shared by the input alignment tables so the number of links is under-estimated. We then also re-produced the two commonly used combination heuristic methods that are based on growing the alignment diagonally (GDF) (Koehn et al., 2003), and adding links based on refined heuristics (H) (Och and Ney, 2003), respectively. We obtained F-measure of 0.6891 for H, and 0.6712 for GDF as shown in Table 1.

We then used our power mean formulation for combination to maximize the F-measure function with the aforementioned simplex algorithm for tuning the power parameter p, where F-measure is computed with respect to the hand aligned development data, which contains 150 sentences. This hand aligned development set is different than the development set for training MT models. While doing so we also optimized table weights $W_q \in (0,\ 1), \sum_q W_q = 1$, which were applied to the alignment tables before combining them using the PM. The W_q allow the algorithm to weight the two directions differently. We found that the F-measure function had many local minima so the simplex algorithm was initialized at several values of p and $\{W_q\}$ to find the globally optimal F-measure.

After obtaining power mean outputs for the alignment entries, they need to be converted into binary valued alignment links, that is, $S_p(a^1_{ij}, a^2_{ij}, ...a^n_{ij})$ needs to be converted into a binary table. There are many ways to do this conversion such as simple thresholding or keeping best N% of the links. In our experiments we used the following simple selection method, which appears to perform better than thresholding. First we sorted links by PM value and then added the links from the top of the sorted list such that e_i and f_j are linked if e_{i-1} and e_{i+1} are connected to f_j, or f_{j-1} and f_{j+1} is linked to e_i, or both e_i and f_j are not connected. After tuning power mean parameter and the alignment weights the best parameter gave an F-measure of 0.6984 which is higher than commonly used GDF by 2.272% and H by 0.93% absolute respectively. We observe in Figure 2 that even though PM has higher F-measure compared with GDF it has significantly fewer number of alignment links suggesting that PM has improved precision on the finding the alignment links. The presented PM based alignment combination can be tuned to optimize any chosen objective, so it is not surprising that we can improve upon previous results based on heuristics.

One of the main advantages of the combining alignment tables using the PM is that our statements are valid for any number of input tables, whereas most heuristic approaches can only process two alignment tables at a time. The presented power mean algorithm, in contrast, can be used to combine any number of alignments in a single step, which, importantly, makes it possible to jointly optimize all of the parameters of the combination process.

In the second set of experiments the PM approach, which we call PM_n, is applied simultaneously to more than two alignments. We obtained four more sets of alignments from the Berkeley aligner (BA) (Liang et al., 2006), the HMM aligner (HA) (Vogel et al., 1996), the alignment based on partial words (PA), and alignment based on dependency based reordering (DA) (Xu et al., 2009). Alignment I was obtained by using Berkeley aligner as an off-the-shelf alignment tool. We built the HMM aligner based on (Vogel et al., 1996) and use the HMM aligner for producing Alignment II. Producing different sets of alignments using different algorithms could be useful because some alignments that are pruned by one algorithm may be kept by another giving us a bigger pool of possible links to chose from.

We produced Alignment III based on partial words. Pashto is morphologically rich language with many prefixes and suffixes. In lack of a morphological segmenter it has been suggested that keeping only first 'n' characters of a word can effectively reduce the vocabulary size and may produce better alignments. (Chiang et al., 2009) used partial words for alignment training in English and Urdu. We trained such alignments using using GIZA++ on parallel data with partial words for Pashto sentences.

The fourth type of alignment we produced, Alignment IV, was motivated by the (Xu et al.,

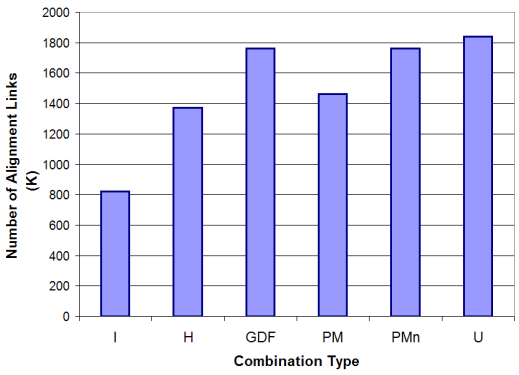


Figure 2: Number of Alignments Links for Different Combination Types

2009). (Xu et al., 2009) showed that translation between subject-verb-object (English) and subject-object-verb (Pashto) languages can be improved by reordering the source side of the parallel data. They obtained dependency tree of the source side and used high level human generated rules to reorder source side using precedence-based movement of dependency subtrees. The rules were particularly useful in reordering of verbs that moved to the end of the sentence. Making the ordering of source and target side more similar may produce better alignments for language pairs which differ in verb ordering, as many alignment algorithms penalize or fail to consider alignments that link words that differ greatly in sentence position. A Pashto language expert was hired to produce similar precedence-based rules for the English-Pashto language pair. Using the rules and algorithm described in (Xu et al., 2009) we reordered all of the source side and used GIZA++ to align the sentences.

The four additional alignment sets just described, including our baseline alignment, Alignment V, were combined using the presented PM$_n$ combination algorithm, where n signifies the number of tables being combined. As seen on Table 1, we obtained an F-measure of 0.7276 which is 12.97% absolute better than intersection and 6.87% better than union. Furthermore PM$_n$, which in these experiments utilizes 5 alignments, is better than PM by 2.92% absolute. This is an encouraging result because this not only shows that we are finding better alignments than intersection and union, but also that combining more than two alignments is useful. We note that PM$_n$ performed 3.85% absolute better than H (Och and Ney, 2003), and 5.64% better than GDF heuristics.

In the above experiments the parameters of the power mean combination method were tuned on development data to optimize alignment F-measure, and the performance of several alignment combination techniques were compared in terms of F-measure. However, it is not clear how correlated alignment F-measures are with BLEU scores, as explained in (Fraser and Marcu, 2007).

While there is no mathematical problem with optimizing the parameters of the presented PM-based combination algorithm w.r.t. BLEU scores, computationally it is not practical to do so because each iteration would require a complete training phase. To further evaluate the quality of the alignments methods being compared in this paper, we built several MT models based on them and compared the resulting BLEU scores.

E2F	Dev	Test
I	0.1064	0.0941
H	0.1028	0.0894
GDF	0.1256	0.1091
PM	0.1214	0.1094
PM$_n$	0.1378	0.1209
U	0.1062	0.0897

Table 2: E2F BLEU: PM Alignment Combination Based MT Model Comparision

We built a standard phrase-based translation system (Koehn et al., 2003) that utilizes a stack-based decoder based on an A^* search. Based on the combined alignments, we extracted phrase tables with a maximum phrase length of 6 for English and 8 for Pashto, respectively. We then trained the lexicalized reordering model that produced distortion costs based on the number of words that are skipped on the target side, in a manner similar to (Al-Onaizan and Papineni, 2006). Our training sentences are a compilation of sentences from various domains collected by DARPA, and hence we were able to build interpolated language model which weights the domains differently. We built an interpolated LM for both

English and Pashto, but for English we had significantly more monolingual sentences (1.4 million in total) compared to slightly more than 100K sentences for Pashto. We tuned our MT model using minimum error rate (Och, 2003) training.

F2E	Dev	Test
I	0.1145	0.1101
H	0.1262	0.1193
GDF	0.1115	0.1204
PM	0.1201	0.1155
PM_n	0.1198	0.1196
U	0.1111	0.1155

Table 3: F2E BLEU : PM Alignment Combination Based MT Model Comparision

We built five different MT models based on Intersection (I), Union (U), (Koehn et al., 2003) Grow Diagonal Final (GDF), (Och and Ney, 2003) H refined heuristics and Power Mean (PM_n) alignment sets where $n = 5$. We obtained BLEU (Papineni et al., 2002) scores for E2F direction as shown in Table 2. As expected MT model based on I alignment has the low BLEU score of 0.1064 on the dev set and 0.0941 on the test set on E2F direction. Intersection, though, has higher precision, but throws away many alignments, so the overall number of alignments is too small to produce a good phrase translation table. Similarly the U alignment also has low scores (0.1062 and 0.0897) on the dev and test sets, respectively. The best scores for E2F direction for both dev and test set is obtained using the model based on PM_n algorithm. We obtained BLEU scores of 0.1378 on the dev set and 0.1209 on the test set which is better than all heuristic based methods. It is better by 1.22 absolute BLEU score on the dev set and 1.18 on a test compared to commonly used GDF (Koehn et al., 2003) heuristics. The above BLEU scores were all computed based on 1 reference. Note that for the e2f direction PM, which combines only 2 alignments, is not worse than any of the heuristic based methods. Also note that the difference in the BLEU score of PM and PM_n is quite large, which indicates that combining more than two alignments using the power mean leads to substantial gains in performance.

Although we saw significant gains on E2F direction we did not see similar gains on F2E direction unfortunately. Matching our expectation Intersection (I) produced the worse results with BLEU scores of 0.1145 and 0.1101 on the dev and test set respectively, as shown in Table 3. Our PM_n algorithm obtained BLEU score of 0.1198 on the dev set and 0.1196 on test set which is better by 0.83 absolute in dev set over GDF. On the test set though performance between PM_n and GDF is only slightly different with 0.1196 for PM_n and 0.1204 for GDF. The standard deviation on test set BLEU scores for F2E direction is only 0.0042 which is one third of the standard deviation in E2F direction at 0.013 signifying that the alignment seems to make less difference in F2E direction for our models. One possible explanation for such results is that the Pashto LM for the E2F direction is trained on a small set of sentences available from training corpus while English LM for F2E direction was trained on 1.4 million sentences. Therefore the English LM, which is trained on significantly more data, is probably more robust to translation model errors.

Type	PT Size (100K)
I	182.17
H	30.73
GDF	27.65
PM	60.87
PM_n	25.67
U	24.54

Table 4: E2F Phrase Table Size

Type	PT Size (100K)
I	139.98
H	56.76
GDF	22.96
PM	47.50
PM_n	21.24
U	20.33

Table 5: F2E Phrase Table Size

Note that different alignments lead to different phrase table (PT) sizes (Figure 2). The intersection (I) method has the least number of alignment links, and tends to produce the largest phrase tables, because there are less restrictions on the

phrases to be extracted. The Union (U) method, on the other hand, tends to produce the least number of phrases, because the phrase extraction algorithm has more constraints to satisfy. We observe that PT produced by intersection is significantly larger than others as seen in Tables 4 and 5. The PT size produced by PM_n as shown in Table 4 is between I and U and is significantly smaller than the other heuristic based methods. It is 7.1% smaller than GDF heuristic based phrase table. Similarly in F2E direction as well (Table 5) we see the similar trend where PM_n PT size is smaller than GDF by 4.2%. The decrease in phrase table size and increase in BLEU scores for most of the dev and test sets show that our PM based combined alignments are helping to produce better MT models.

6 Conclusion and Future Work

We have presented a mathematical formulation for combining alignment tables based on their power mean. The presented framework allows us to find the optimal alignment between intersection and union by finding the best power mean parameter between 0 and ∞, which correspond to intersection and union operations, respectively. We evaluated the proposed method empirically by computing BLEU scores in English-Pashto translation task and also by computing an F-measure with respect to human alignments. We showed that the approach is more effective than intersection, union, the heuristics of (Och and Ney, 2003), and the grow diagonal final (GDF) algorithm of (Koehn et al., 2003). We also showed that our algorithm is not limited to two tables, which makes it possible to jointly optimize the combination of multiple alignment tables to further increase performance.

In future work we would like to address two particular issues. First, in this work we converted power mean outputs to binary alignment links by simple selection process. We are currently investigating ways to integrate the binary constraint into the PM-based optimization algorithm. Second, we do not have to limit ourselves to alignments tables that are binary. PM based algorithm can combine alignments that are not binary, which makes it easier to integrate other sources of information such as posterior probability of word translation into the alignment combination framework.

7 Acknowledgment

This work is partially supported by the DARPA TRANSTAC program under the contract number of NBCH2030007. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of DARPA.

References

Al-Onaizan, Yaser and Kishore Papineni. 2006. Distortion models for statistical machine translation. In *Proceedings of ACL.*

Ayan, Necip, Bonnie J. Dorr, , and Nizar Habash. 2004. Multi-align: Combining linguistic and statistical techniques to improve alignments for adaptable mt. In *Proceedings of the 6th Conference of the Association for Machine Translation in the Americas.*

Brown, P., V. Della Pietra, S. Della Pietra, and R. Mercer. 1993. The mathematics of statistical machine translation: parameter estimation. *Computational Linguistics*, 19(2):263–311.

Chiang, David, Kevin Knight, and Samad Echihabi. 2009. In *Presentation at NIST MT 2009 Workshop, August.*

Chiang, David. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of ACL.*

Fraser, Alexander and Daniel Marcu. 2007. Measuring word alignment quality for statistical machine translation. *Comput. Linguist.*, 33(3):293–303.

Koehn, Philipp, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of HLT/NAACL.*

Liang, Percy, Ben Taskar, and Dan Klein. 2006. Alignment by agreement. In *Proceedings of ACL.*

Matusov, Evgeny, Richard Zens, and Hermann Ney. 2004. Symmetric word alignments for statistical machine translation. In *Proceedings of COLING*, page 219, Morristown, NJ, USA.

Nelder, JA and R Mead. 1965. A simplex method for function minimization. *The Computer Journal 7: 308-313.*

Och, F. J. and H. Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Och, Franz J. 2003. Minimum error rate training in statistical machine. In *Proceedings of ACL*.

Papineni, Kishore, Salim Roukos, Todd Ward, and Wei jing Zhu. 2002. Bleu: A method for automatic evaluation of machine translation. In *In Proceedings of ACL*, pages 311–318.

Vogel, Stephan, Hermann Ney, and Christoph Tillmann. 1996. Hmm-based word alignment in statistical translation. In *COLING 96: The 16th Int. Conf. on Computational Linguistics*, pages 836–841.

Xiang, Bing, Yonggang Deng, and Bowen Zhou. 2010. Diversify and combine: Improving word alignment for machine translation on low-resource languages. In *Proceedings of ACL*.

Xu, Peng, Jaeho Kang, Michael Ringgaard, and Franz Och. 2009. Using a dependency parser to improve smt for subject-object-verb languages. In *NAACL*, pages 245–253, Morristown, NJ, USA.

Yamada, Kenji and Kevin Knight. 2001. A syntax-based statistical translation model. In *Proceedings of ACL*, pages 523–530, Toulouse, France, July. ACL.

Machine Translation with Lattices and Forests

Haitao Mi[†‡] **Liang Huang**[‡] **Qun Liu**[†]

[†]Key Lab. of Intelligent Information Processing
Institute of Computing Technology
Chinese Academy of Sciences
{htmi,liuqun}@ict.ac.cn

[‡]Information Sciences Institute
Viterbi School of Engineering
University of Southern California
{lhuang,haitaomi}@isi.edu

Abstract

Traditional 1-best translation pipelines suffer a major drawback: the errors of 1-best outputs, inevitably introduced by each module, will propagate and accumulate along the pipeline. In order to alleviate this problem, we use compact structures, *lattice* and *forest*, in each module instead of 1-best results. We integrate both lattice and forest into a single tree-to-string system, and explore the algorithms of lattice parsing, lattice-forest-based rule extraction and decoding. More importantly, our model takes into account all the probabilities of different steps, such as segmentation, parsing, and translation. The main advantage of our model is that we can make global decision to search for the best segmentation, parse-tree and translation in one step. Medium-scale experiments show an improvement of +0.9 BLEU points over a state-of-the-art forest-based baseline.

1 Introduction

Statistical machine translation (SMT) has witnessed promising progress in recent years. Typically, conventional SMT is characterized as a 1-best pipeline system (Figure 1(a)), whose modules are independent of each other and only take as input 1-best results from the previous module. Though this assumption is convenient to reduce the complexity of SMT systems. It also bring a major drawback of error propagation. The errors of 1-best outputs, introduced inevitably in each phase, will propagate and accumulate along the pipeline. Not recoverable in the final decoding step. These errors will severely hurt the translation quality. For example, if the accuracy of each module is 90%, the final accuracy will drop to 73% after three separate phases.

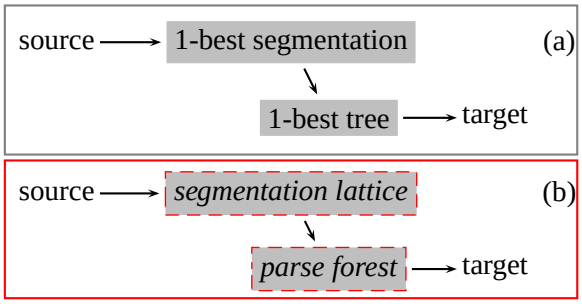


Figure 1: The pipeline of tree-based system: (a) 1-best (b) lattice-forest.

To alleviate this problem, an obvious solution is to widen the pipeline with *k*-best lists rather than 1-best results. For example Venugopal et al. (2008) use *k*-best alignments and parses in the training phase. However, with limited scope and too many redundancies, it is inefficient to search separately on each of these similar lists (Huang, 2008).

Another efficient method is to use compact data structures instead of *k*-best lists. A *lattice* or *forest*, compactly encoded exponentially many derivations, have proven to be a promising technique. For example, Mi and Huang (2008), Mi et al. (2008), Liu et al. (2009) and Zhang et al. (2009) use forests in rule extraction and decoding phases to extract more general rules and weaken the influence of parsing errors; Dyer et al. (2008) use word lattice in Chinese word segmentation and Arabic morphological variation phases to weaken the influence of segmentation errors; Huang (2008) and

Coling 2008: Poster Volume, pages 837–845,
Beijing, August 2010

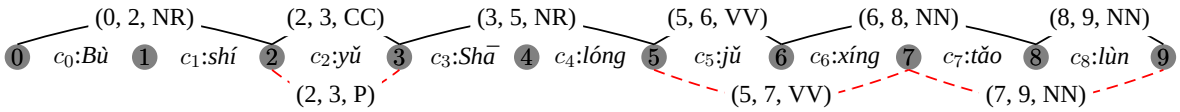


Figure 2: The lattice of the example:“ *Bù shí yǔ Shā lóng jǔ xíng tǎo lùn*.” The solid lines show the 1-best result, which is wrong.

Jiang et al. (2008b) stress the problems in reranking phase. Both lattices and forests have become popular in machine translation literature.

However, to the best of our knowledge, previous work only focused on one module at a time. In this paper, we investigate the combination of lattice and forest (Section 2), as shown in Figure 1(b). We explore the algorithms of lattice parsing (Section 3.2), rule extraction (Section 4) and decoding (Section 5). More importantly, in the decoding step, our model can search among not only more parse-trees but also more segmentations encoded in the lattice-forests and can take into account all the probabilities of segmentations and parse-trees. In other words, our model postpones the disambiguition of segmentation and parsing into the final translation step, so that we can do global search for the best segmentation, parse-tree and translation in one step. When we integrate a lattice into a forest system, medium-scale experiments (Section 6) show another improvement of +0.9 BLEU points over a state-of-the-art forest-based system.

2 Compact Structures

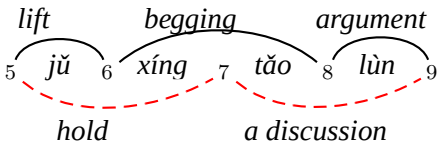
A **word lattice** (Figure 2) is a compact representation of all the possible of segmentations and POS tags, while a **parse forest** (Figure 5) is a compact representation of all parse trees.

2.1 Word Lattice

For a given input sentence $C = c_0..c_{n-1}$, where c_i denotes a character at position i, and n is the length of the sentence.

A word lattice (Figure 2), or **lattice** in short, is a set of **edges** L, where each edge is in the form of (i, j, X), which denotes a word of tag X, covering characters c_i through c_{j-1}. For example, in Figure 2, (7, 9, NN) is a noun “*tǎolùn*” of two characters.

The lattice in Figure 2 shows result of the example:“ *Bù shí yǔ Shā lóng jǔ xíng tǎo lùn* ”. One ambiguity comes from the POS tag of word “*yǔ*” (preposition (P) or conjunction (CC)). The other one is the segmentation ambiguity of the last four characters, we can segment into either “*jǔ xíngtǎo lùn*” (solid lines), which means *lift*, *begging* and *argument* separately for each word or “*jǔxíng tǎolùn*” (dashed lines), which means *hold a discussion*.



The solid lines above (and also in Figure 2) show the 1-best result, which is obviously wrong. If we feed it into the next modules in the SMT pipeline, parsing and translation will be become much more difficult, since the segmentation is not recoverable. So it is necessary to postpone error segmentation decisions to the final translation step.

2.2 Parse Forest

In parsing scenario, a parse forest (Figrure 5), or **forest** for short, can be formalized as a hypergraph H, a pair $\langle V, E \rangle$, where node $v \in V$ is in the form of $X_{i,j}$, which denotes the recognition of nonterminal X spanning the substring $c_{i:j-1}$ from positions c_i through c_{j-1}. Each hyperedge $e \in E$ is a pair $\langle tails(e), head(e) \rangle$, where $head(e) \in V$ is the **consequent node** in an instantiated deductive step, and $tails(e) \in (V)^*$ is the list of **antecedent nodes**.

For the following deduction:

$$\frac{\mathrm{NR}_{0,2} \quad \mathrm{CC}_{2,3} \quad \mathrm{NR}_{3,5}}{\mathrm{NP}_{0,5}} \qquad (*)$$

its hyperedge e^* is notated:

$$\langle(\text{NR}_{0,2},\ \text{CC}_{2,3},\ \text{NR}_{3,5}),\ \ \text{NP}_{0,5}\rangle.$$

where

$$head(e^*) = \{\text{NP}_{0,5}\}, \text{ and}$$
$$tails(e^*) = \{\text{NR}_{0,2}, \text{CC}_{2,3}, \text{NR}_{3,5}\}.$$

We also denote $IN(v)$ to be the set of **incoming hyperedges** of node v, which represents the different ways of deriving v. For simplicity, we only show a tree in Figure 5(a) over 1-best segmentation and POS tagging result in Figure 2. So the $IN(\text{NP}_{0,5})$ is $\{e^*\}$.

3 Lattice Parsing

In this section, we first briefly review the conventional CYK parsing, and then extend to lattice parsing. More importantly, we propose a more efficient parsing paradigm in Section 3.3.

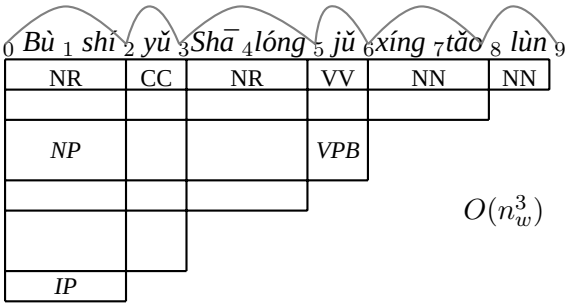
3.1 Conventional Parsing

The conventional CYK parsing algorithm in Figure 3(a) usually takes as input a single sequence of words, so the CYK cells are organized over words. This algorithm consists of two steps: initialization and parsing. The first step is to initialize the CYK cells, whose span size is one, with POS tags produced by a POS tagger or defined by the input string[1]. For example, the top line in Figure 3(a) is initialized with a series of POS tags in 1-best segmentation. The second step is to search for the best syntactic tree under a context-free grammar. For example, the tree composed by the solid lines in Figure 5(a) shows the parsing tree for the 1-best segmentation and POS tagging results.

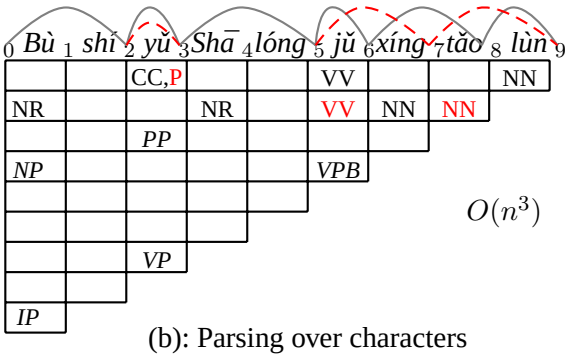
3.2 Lattice Parsing

The main differences of our lattice parsing in Figure 3(b) from conventional approach are listed in following: First, the CYK cells are organized over characters rather than words. Second, in the initialization step, we only initialize the cells with all edges L in the lattice. Take the edge (7, 9, NN) in Figure 2 for example, the corresponding cell should be $(7, 9)$, then we add a leaf node $v = \text{NN}_{7,9}$ with a word *tǎolùn*. The final initialization is shown in Figure 3(b), which shows that

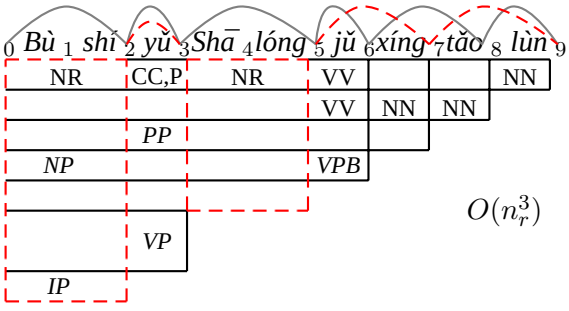
[1]For simplicity, we assume the input of a parser is a segmentation and POS tagging result



(a): Parsing over 1-best segmentation



(b): Parsing over characters



(c): Parsing over most-refined segmentation

Figure 3: CKY parsing charts (a): Conventional parsing over 1-best segmentation. (b): Lattice parsing over characters of input sentence. (c): Lattice parsing over most-refined segmentation of lattice. n_w and n_r denotes the number of tokens over the 1-best segmentation and the most-refined segmenation respectively, and $n_w \leq n_r \leq n$.

lattice parsing can initialize the cells, whose span size is larger than one. Third, in the deduction step of the parsing algorithm *i*, *j*, *k* are the indexes between characters rather than words.

We formalize our lattice parser as a deductive proof system (Shieber et al., 1994) in Figure 4.

Following the definitions of the previous Sec-

tion, given a set of edges L of a lattice for an input sentence $C = c_0..c_{n-1}$ and a PCFG grammar: a 4-tuple $\langle N, \Sigma, P, S\rangle$, where N is a set of non-terminals, Σ is a set of terminal symbols, P is a set of inference rules, each of which is in the form of $X \rightarrow \alpha : p$ for $X \in N$, $\alpha \in (N \cup \Sigma)^*$ and p is the probability, and $S \in N$ is the start symbol. The deductive proof system (Figure 4) consists of **axioms**, **goals** and **inference rules**. The axioms are converted by edges in L. Take the (5, 7, NN) associated with a weight p_1 for example, the corresponding axiom is NN $\rightarrow$ *tǎolùn* : p_1. All axioms converted from the lattice are shown in Figure 3(b) exclude the italic non-terminals. Please note that all the probabilities of the edges L in a lattice are taken into account in the parsing step. The goals are the recognition $X_{0,n} \in S$ of the whole sentence. The inference rules are the deductions in parsing. Take the deduction (*) for example, it will prove a new item $NP_{0,5}$ (italic NP in Figure 3(b)) and generate a new hyper-edge e^* (in Figure 5(b)). So the parsing algorithm starts with the axioms, and then applies the inference rules to prove new items until a goal item is proved. The final whole forest for the input lattice (Figure 2) is shown in Figure 5(b). The extra hyper-edges of lattice-forest are highlighted with dashed lines, which can inference the input sentence correctly. For example: "*yǔ*" is tagged into P rather than CC.

3.3 Faster Parsing with Most-refined Lattice

However, our statistics show that the average number of characters n in a sentence is 1.6 times than the number of words n_w in its 1-best segmentation. As a result, the parsing time over the characters will grow more than 4 times than parsing over the 1-best segmentation, since the time complexity is $O(n^3)$. In order to alleviate this problem, we reduce the parsing time by using **most-refined segmentation** for a lattice, whose number of tokens is n_r and has the property $n_w \leq n_r \leq n$.

Given a lattice with its edges L over indexes $(0, .., n)$, a index i is a **split point**, if and only if there exists some edge $(i, j, X) \in L$ or $(k, i, X) \in L$. The **most-refined segmentation**, or **ms** for short, is the segmentation result by using all split points in a lattice. For example, the corresponding **ms** of the example is "*Bùshí yǔ Shālóng jǔ xíng tǎo lùn*" since points 1 and 4 are *not* split points.

Item form: $X_{i,j}$

Axioms: $\dfrac{}{X_{i,j} : p(i,j,X)}\ (i,j,X) \in L$

Infer. rules: $\dfrac{X_{i,k} : p_1\ Y_{k,j} : p_2}{Z_{i,j} : pp_1p_2} Z \rightarrow XY : p \in P$

Goals: $X_{0,n}$

Figure 4: Lattice parsing as deductive proof system. The i, j, k are the indexes between characters.

Figure 3(c) shows the CKY parsing cells over most-refined segmentation, the average number of tokens n_r is reduced by combining columns, which are shown with red dashed boxes. As a result, the search space is reduced without losing any derivations. Theoretically, the parsing over *fs* will speed up in $O((n/n_r)^3)$. And our experiments in Section 6 show the efficiency of our new approach.

It turns out that the parsing algorithm developed in lattice-parsing Section 3.2 can be used here without any change. The non-terminals inducted are also shown in Figure 3(c) in italic style.

4 Rule Extraction with Lattice & Forest

We now explore the extraction algorithm from aligned source lattice-forest and target string[2], which is a tuple $\langle F, \tau, a\rangle$ in Figure 5(b). Following Mi and Huang (2008), we extract minimal rules from a lattice-forest also in two steps:

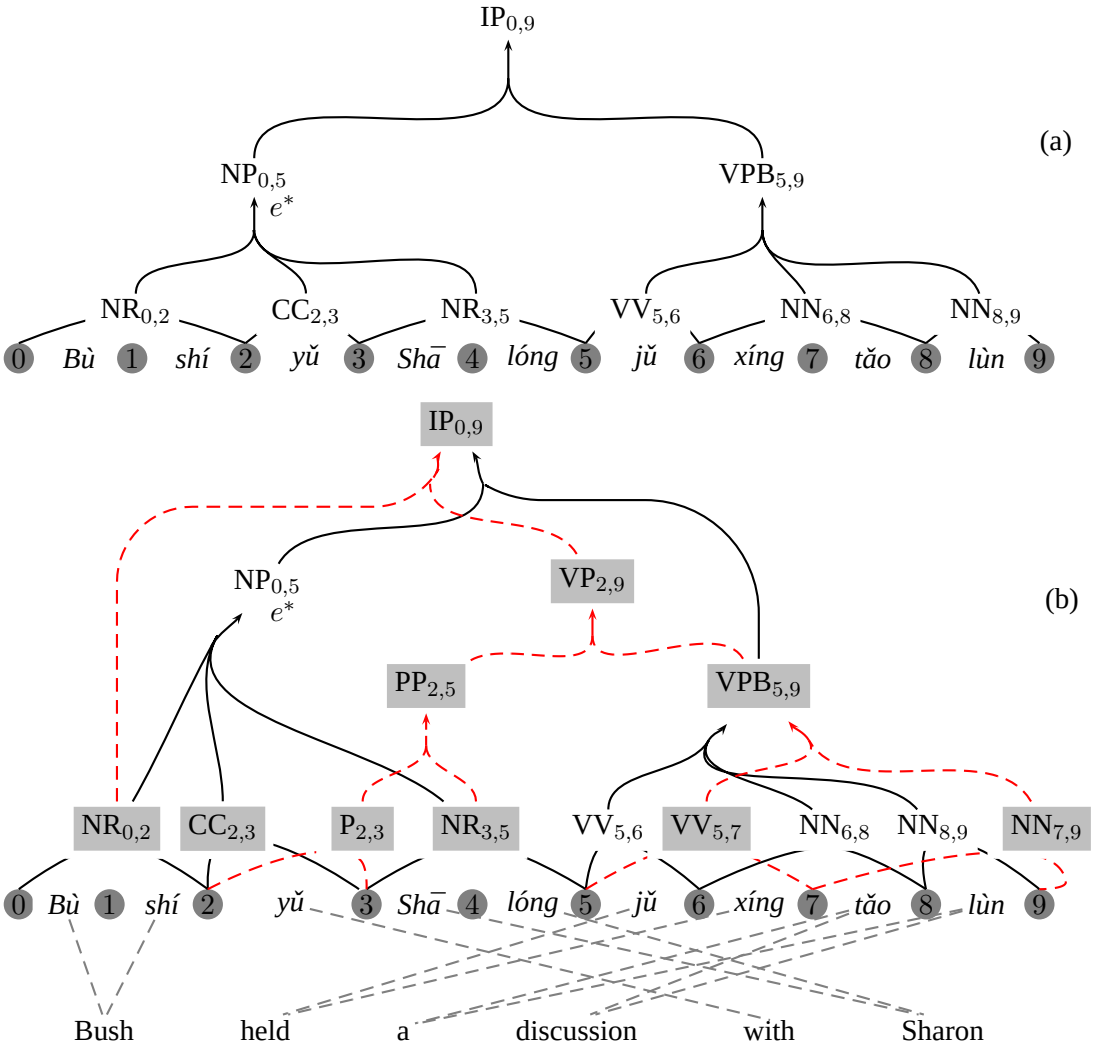
(1) frontier set computation

(2) fragmentation

Following the algorithms developed by Mi and Huang (2008) in Algorithm 1, all the nodes in *frontier set* (*fs*) are highlighted with gray in Figure 5(b).

Our process of fragmentation (lines 1- 13) is to visit each frontier node v and initial a queue ($open$) of growing fragments with a pair of empty fragment and node v (line 3). Each fragment is associated with a list of *expansion sites* ($front$) being

[2]For simplicity and consistency, we use character-based lattice-forest for the running example. The "*Bù*" and "*shí*" are aligned to the same word "*Bush*". In our experiment, we use most-refined segmentation to run lattice-parsing and word alignment.



Forest only (Minimal rules)	Lattice & forest (Extra minimal rules)
IP(NP(x_1:NR x_2:CC x_3:NR) x_4:VPB)	IP(x_1:NR x_2:VP) $\rightarrow x_1\ x_2$
$\rightarrow x_1\ x_4\ x_2\ x_3$	VP(x_1:PP x_2:VPB) $\rightarrow x_2\ x_1$
CC(*yǔ*) $\rightarrow$*with*	PP(x_1:P x_2:NR) $\rightarrow x_1\ x_2$
NR(*Shālóng*) $\rightarrow$*Sharon*	P(*yǔ*) $\rightarrow$*with*
NR(*Bùshí*) $\rightarrow$*Bush*	VPB(x_1:VV x_2:NN) $\rightarrow x_1\ x_2$
VPB(VV(*jǔ*) NN(*xíngtǎo*) NN(*lùn*))	VV(*jǔxíng*) $\rightarrow$*held*
$\rightarrow$*held a discussion*	NN(*tǎolùn*) $\rightarrow$*a discussion*

(c)

Figure 5: (a): The parse forest over the 1-best segmentation and POS tagging result. (b): Word-aligned tuple $\langle F, \tau, a\rangle$: the lattice-forest F, the target string τ and the word alingment a. The solid hyperedges form the forest in (a). The dashed hyperedges are the extra hyperedges introduced by the lattice-forest. (c): The minimal rules extracted on forest-only (left column), and the extra minimal rules extracted on lattice-forest (right column).

the subset of leaf nodes of the current fragment that are *not* in the *fs* except for the initial node v. Then we keep expanding fragments in $open$ in following way. If current fragment is complete, whose expansion sites is empty, we extract rule corresponding to the fragment and its target string

Code 1 Rule Extraction (Mi and Huang, 2008).

Input: lattice-forest F, target sentence τ, and alignment a
Output: minimal rule set $\mathcal{R}$

1: $fs \leftarrow \text{FROSET}(F, \tau, a)$ ▷ frontier set
2: **for** each $v \in fs$ **do**
3: $open \leftarrow \{\langle \varnothing, \{v\}\rangle\}$ ▷ initial queue
4: **while** $open \neq \varnothing$ **do**
5: $\langle frag, front\rangle \leftarrow open.pop()$
6: **if** $front = \varnothing$ **then** ▷ finished?
7: generate a rule r using $frag$
8: $\mathcal{R}$.append(r)
9: **else** ▷ incomplete: further expand
10: $u \leftarrow front$.pop() ▷ expand frontier
11: **for** each $e \in IN(u)$ **do**
12: $f \leftarrow front \cup (tails(e) \setminus fs)$
13: $open$.append($\langle frag \cup \{e\}, f\rangle$)

(line 7) . Otherwise we pop one expansion node u to grow and spin-off new fragments by $IN(u)$, adding new expansion sites (lines 11- 13), until all active fragments are complete and $open$ queue is empty.

The extra minimal rules extracted on lattice-forest are listed at the right bottom of Figure 5(c). Compared with the forest-only approach, we can extract smaller and more general rules.

After we get all the minimal rules, we compose two or more minimal rules into composed rules (Galley et al., 2006), which will be used in our experiments.

For each rule r extracted, we also assign a fractional count which is computed by using inside-outside probabilities:

$$c(r) = \frac{\alpha(root(r)) \cdot P(lhs(r)) \cdot \prod_{v \in yield(root(r))} \beta(v)}{\beta(\text{TOP})}, \quad (1)$$

where $root(r)$ is the root of the rule, $lhs(r)$ is the left-hand-side of rule, $rhs(r)$ is the right-hand-side of rule, $P(lhs(r))$ is the product of all probabilities of hyperedges involved in $lhs(r)$, $yield(root(r))$ is the leave nodes, TOP is the root node of the forest, $\alpha(v)$ and $\beta(v)$ are outside and inside probabilities, respectively.

Then we compute three conditional probabilities for each rule:

$$P(r \mid lhs(r)) = \frac{c(r)}{\sum_{r': lhs(r')=lhs(r)} c(r')} \quad (2)$$

$$P(r \mid rhs(r)) = \frac{c(r)}{\sum_{r': rhs(r')=rhs(r)} c(r')} \quad (3)$$

$$P(r \mid root(r)) = \frac{c(r)}{\sum_{r': root(r')=root(r)} c(r')}. \quad (4)$$

All these probabilities are used in decoding step (Section 5). For more detail, we refer to the algorithms of Mi and Huang (2008).

5 Decoding with Lattice & Forest

Given a source-side lattice-forest F, our decoder searches for the best derivation d^* among the set of all possible derivation D, each of which converts a tree in lattice-forest into a target string τ:

$$d^* = \underset{d \in D, T \in F}{\text{argmax}} P(d|T)^{\lambda_0} \cdot e^{\lambda_1 |d|} \cdot LM(\tau(d))^{\lambda_2} \cdot e^{\lambda_3 |\tau(d)|}, \quad (5)$$

where $|d|$ is the penalty term on the number of rules in a derivation, $LM(\tau(d))$ is the language model and $e^{\lambda_3|\tau(d)|}$ is the length penalty term on target translation. The $P(d|T)$ decomposes into the product of rule probabilities $P(r)$, each of which is decomposed further into

$$P(d|T) = \prod_{r \in d} P(r). \quad (6)$$

Each $P(r)$ in Equation 6 is decomposed further into the production of five probabilities:

$$\begin{aligned} P(r) = & P(r|lhs(r))^{\lambda_4} \\ & \cdot P(r|rhs(r))^{\lambda_5} \\ & \cdot P(r|root(lhs(r))^{\lambda_6} \\ & \cdot P_{lex}(lhs(r)|rhs(r))^{\lambda_7} \\ & \cdot P_{lex}(rhs(r)|lhs(r))^{\lambda_8}, \end{aligned} \quad (7)$$

where the last two are the lexical probabilities between the terminals of $lhs(r)$ and $rhs(r)$. All the weights of those features are tuned by using Minimal Error Rate Training (Och, 2003).

Following Mi et al. (2008), we first convert the lattice-forest into *lattice translation forest* with the conversion algorithm proposed by Mi et al. (2008),

and then the decoder finds the best derivation on the lattice translation forest. For 1-best search, we use the *cube pruning* technique (Chiang, 2007; Huang and Chiang, 2007) which approximately intersects the translation forest with the LM. For k-best search after getting 1-best derivation, we use the lazy Algorithm 3 of Huang and Chiang (2005) to incrementally compute the second, third, through the kth best alternatives.

For more detail, we refer to the algorithms of Mi et al. (2008).

6 Experiments

6.1 Data Preparation

Our experiments are on Chinese-to-English translation. Our training corpus is FBIS corpus with about 6.9M/8.9M words in Chinese/English respectively.

We use SRI Language Modeling Toolkit (Stolcke, 2002) to train a 4-gram language model with Kneser-Ney smoothing on the first 1/3 of the Xinhua portion of Gigaword corpus.

We use the 2002 NIST MT Evaluation test set as development set and the 2005 NIST MT Evaluation test set as test set. We evaluate the translation quality using the case-insensitive BLEU-4 metric (Papineni et al., 2002). We use the standard MERT (Och, 2003) to tune the weights.

6.1.1 Baseline Forest-based System

We first segment the Chinese sentences into the 1-best segmentations using a state-of-the-art system (Jiang et al., 2008a), since it is not necessary for a conventional parser to take as input the POS tagging results. Then we parse the segmentation results into forest by using the parser of Xiong et al. (2005). Actually, the parser will assign multiple POS tags to each word rather than one. As a result, our baseline system has already postponed the POS tagging disambiguition to the decoding step. Forest is pruned by using a marginal probability-based pruning algorithm similar to Huang (2008). The pruning threshold are $p_f = 5$ and $p_f = 10$ at rule extraction and decoding steps respectively.

We word-align the strings of 1-best segmentations and target strings with GIZA++ (Och and Ney, 2000) and apply the refinement method "grow-diag-final-and" (Koehn et al., 2003) to get the final alignments. Following Mi and Huang (2008) and Mi et al. (2008), we also extract rules from forest-string pairs and translate forest to string.

6.1.2 Lattice-forest System

We first segment and POS tag the Chinese sentences into word lattices using the same system (Jiang et al., 2008a), and prune each lattice into a reasonable size using the marginal probability-based pruning algorithm.

Then, as current GIZA++ (Och and Ney, 2000) can only handle alignment between string-string pairs, and word-alingment with the pairs of Chinese characters and target-string will obviously result in worse alignment quality. So a much better way to utilize GIZA++ is to use the most-refined segmentation for each lattice instead of the character sequence. This approach can be viewed as a compromise between character-string and lattice-string word-alignment paradigms. In our experiments, we construct the most-refined segmentations for lattices and word-align them against the English sentences. We again apply the refinement method "grow-diag-final-and" (Koehn et al., 2003) to get the final alignments.

In order to get the lattice-forests, we modified Xiong et al. (2005)'s parser into a lattice parser, which produces the pruned lattice forests for both training, dev and test sentences. Finally, we apply the rule extraction algorithm proposed in this paper to obtain the rule set. Both lattices and forests are pruned using a marginal probability-based pruning algorithm similar to Huang (2008). The pruning threshold of lattice is $p_l = 20$ at both the rule extraction and decoding steps, the thresholds for the latice-forests are $p_f = 5$ and $p_f = 10$ at rule extraction and decoding steps respectively.

6.2 Results and Analysis

Table 1 shows results of two systems. Our lattice-forest (LF) system achieves a BLEU score of 29.65, which is an absolute improvement of 0.9 points over the forest (F) baseline system, and the improvement is statistically significant at $p < 0.01$ using the *sign-test* of Collins et al. (2005).

The average number of tokens for the 1-best and most-refined segmentations are shown in second column. The average number of characters is 46.7, which is not shown in Table 1. Com-

Sys	Avg # of		Rules		BLEU
	tokens	links	All	dev&tst	
F	28.7	35.1	29.6M	3.3M	28.75
LF	37.1	37.1	23.5M	3.4M	29.65

Table 1: Results of forest (F) and lattice-forest (LF) systems. Please note that lattice-forest system only extracts 23.5M rules, which is only 79.4% of the rules extracted by forest system. However, in decoding step, lattice-forest system can use more rules after filtered on dev and test sets.

pared with the characters-based lattice parsing, our most-refined lattice parsing speeds up parsing by $(37.1/46.7)^3 \approx 2$ times, since parsing complexity is $O(n^3)$.

More interestingly, our lattice-forest model only extracts 23.5M rules, which is 79.4% percent of the rules extracted by the baseline system. The main reason lies in the larger average number of words for most-refined segmentations over lattices being 37.1 words vs 28.7 words over 1-best segmentations. With much finer granularity, more word aligned links and restrictions are introduced during the rule extraction step by GIZA++. However, more rules can be used in the decoding step for the lattice-forest system, since the lattice-forest is larger than the forest over 1-best segmentation.

We also investigate the question of how often the non 1-best segmentations are picked in the final translation. The statistic on our dev set suggests 33% of sentences choose non 1-best segmentations. So our lattice-forest model can do global search for the best segmentation and parse-tree to direct the final translation. More importantly, we can use more translation rules in the translation step.

7 Related Works

Compactly encoding exponentially many derivations, lattice and forest have been used in some previous works on SMT. To alleviate the problem of parsing error in 1-best tree-to-string translation model, Mi et al. (2008) first use forest to direct translation. Then Mi and Huang (2008) use forest in rule extraction step. Following the same direction, Liu et al. (2009) use forest in tree-to-tree model, and improve 1-best system by 3 BLEU points. Zhang et al. (2009) use forest in tree-sequence-to-string model and also achieve a promising improvement. Dyer et al. (2008) combine multiple segmentations into word lattice and then use lattice to direct a phrase-based translation decoder. Then Dyer (2009) employ a single Maximum Entropy segmentation model to generate more diverse lattice, they test their model on the hierarchical phrase-based system. Lattices and forests can also be used in Minimal Error Rate Training and Minimum Bayes Risk Decoding phases (Macherey et al., 2008; Tromble et al., 2008; DeNero et al., 2009; Kumar et al., 2009; Li and Eisner, 2009). Different from the works listed above, we mainly focus on how to combine lattice and forest into a single tree-to-string system.

8 Conclusion and Future Work

In this paper, we have proposed a lattice-forest based model to alleviate the problem of error propagation in traditional single-best pipeline framework. Unlike previous works, which only focus on one module at a time, our model successfully integrates lattice into a state-of-the-art forest tree-to-string system. We have explored the algorithms of lattice parsing, rule extraction and decoding. Our model postpones the disambiguition of segmentation and parsing into the final translation step, so that we can make a more global decision to search for the best segmentation, parse-tree and translation in one step. The experimental results show that our lattice-forest approach achieves an absolute improvement of +0.9 points in term of BLEU score over a state-of-the-art forest-based model.

For future work, we would like to pay more attention to word alignment between lattice pairs and forest pairs, which would be more principled than our current method of word alignment between most-refined segmentation and string.

Acknowledgement

We thank Steve DeNeefe and the three anonymous reviewers for comments. The work is supported by National Natural Science Foundation of China, Contracts 90920004 and 60736014, and 863 State Key Project No. 2006AA010108 (H. M and Q. L.), and in part by DARPA GALE Contract No. HR0011-06-C-0022, and DARPA under DOI-NBC Grant N10AP20031 (L. H and H. M).

References

David Chiang. 2007. Hierarchical phrase-based translation. *Comput. Linguist.*, 33(2):201–228.

Michael Collins, Philipp Koehn, and Ivona Kucerova. 2005. Clause restructuring for statistical machine translation. In *Proceedings of ACL*, pages 531–540, Ann Arbor, Michigan, June.

John DeNero, David Chiang, and Kevin Knight. 2009. Fast consensus decoding over translation forests. In *Proceedings of ACL/IJCNLP*.

Christopher Dyer, Smaranda Muresan, and Philip Resnik. 2008. Generalizing word lattice translation. In *Proceedings of ACL-08: HLT*, pages 1012–1020, Columbus, Ohio, June.

C. Dyer. 2009. Using a maximum entropy model to build segmentation lattices for mt. In *Proceedings of NAACL*.

Michel Galley, Jonathan Graehl, Kevin Knight, Daniel Marcu, Steve DeNeefe, Wei Wang, and Ignacio Thayer. 2006. Scalable inference and training of context-rich syntactic translation models. In *Proceedings of COLING-ACL*, pages 961–968, Sydney, Australia, July.

Liang Huang and David Chiang. 2005. Better k-best parsing. In *Proceedings of IWPT*.

Liang Huang and David Chiang. 2007. Forest rescoring: Faster decoding with integrated language models. In *Proceedings of ACL*, pages 144–151, June.

Liang Huang. 2008. Forest reranking: Discriminative parsing with non-local features. In *Proceedings of ACL*.

Wenbin Jiang, Liang Huang, Qun Liu, and Yajuan Lü. 2008a. A cascaded linear model for joint chinese word segmentation and part-of-speech tagging. In *Proceedings of ACL-08: HLT*.

Wenbin Jiang, Haitao Mi, and Qun Liu. 2008b. Word lattice reranking for chinese word segmentation and part-of-speech tagging. In *Proceedings of Coling 2008*.

Philipp Koehn, Franz Joseph Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of HLT-NAACL*, pages 127–133, Edmonton, Canada, May.

Shankar Kumar, Wolfgang Macherey, Chris Dyer, and Franz Och. 2009. Efficient minimum error rate training and minimum bayes-risk decoding for translation hypergraphs and lattices. In *Proceedings of the ACL/IJCNLP 2009*.

Zhifei Li and Jason Eisner. 2009. First- and second-order expectation semirings with applications to minimum-risk training on translation forests. In *Proceedings of EMNLP*, pages 40–51, Singapore, August. Association for Computational Linguistics.

Yang Liu, Yajuan Lü, and Qun Liu. 2009. Improving tree-to-tree translation with packed forests. In *Proceedings of ACL/IJCNLP*, August.

Wolfgang Macherey, Franz Och, Ignacio Thayer, and Jakob Uszkoreit. 2008. Lattice-based minimum error rate training for statistical machine translation. In *Proceedings of EMNLP 2008*.

Haitao Mi and Liang Huang. 2008. Forest-based translation rule extraction. In *Proceedings of EMNLP 2008*, pages 206–214, Honolulu, Hawaii, October.

Haitao Mi, Liang Huang, and Qun Liu. 2008. Forest-based translation. In *Proceedings of ACL-08:HLT*, pages 192–199, Columbus, Ohio, June.

Franz J. Och and Hermann Ney. 2000. Improved statistical alignment models. In *Proceedings of ACL*, pages 440–447.

Franz J. Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of ACL*, pages 160–167.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of ACL*, pages 311–318, Philadephia, USA, July.

Stuart M. Shieber, Yves Schabes, and Fernando C. N. Pereira. 1994. Principles and implementation of deductive parsing.

Andreas Stolcke. 2002. SRILM - an extensible language modeling toolkit. In *Proceedings of ICSLP*, volume 30, pages 901–904.

Roy Tromble, Shankar Kumar, Franz Och, and Wolfgang Macherey. 2008. Lattice Minimum Bayes-Risk decoding for statistical machine translation. In *Proceedings of EMNLP 2008*.

Ashish Venugopal, Andreas Zollmann, Noah A. Smith, and Stephan Vogel. 2008. Wider pipelines: N-best alignments and parses in MT training. In *Proceedings of AMTA*.

Deyi Xiong, Shuanglong Li, Qun Liu, and Shouxun Lin. 2005. Parsing the Penn Chinese Treebank with Semantic Knowledge. In *Proceedings of IJCNLP 2005*, pages 70–81.

Hui Zhang, Min Zhang, Haizhou Li, Aiti Aw, and Chew Lim Tan. 2009. Forest-based tree sequence to string translation model. In *Proceedings of the ACL/IJCNLP 2009*.

Automatic Persian WordNet Construction

Mortaza Montazery
School of Electrical and Computer Engineering
College Engineering, University of Tehran
Mortaza.gh@gmail.com

Heshaam Faili
School of Electrical and Computer Engineering
College Engineering, University of Tehran
hfaili@ut.ac.ir

Abstract

In this paper, an automatic method for Persian WordNet construction based on Prenceton WordNet 2.1 (PWN) is introduced. The proposed approach uses Persian and English corpora as well as a bilingual dictionary in order to make a mapping between PWN synsets and Persian words. Our method calculates a score for each candidate synset of a given Persian word and for each of its translation, it selects the synset with maximum score as a link to the Persian word. The manual evaluation on selected links proposed by our method on 500 randomly selected Persian words, shows about 76.4% quality respect to precision measure. By augmenting the Persian WordNet with the un-ambiguous words, the total accuracy of automatically extracted Persian WordNet is about 82.6% which outperforms the previously semi-automated generated Persian WordNet by about 12.6%.

1 Introduction

In Natural Language Processing (NLP) wide coverage lexical databases are used in different area such as information retrieval and cross-language information retrieval. WordNet is an example for a lexical database that groups words into sets of synonyms and categorizes them in four categories: noun, verb, adjective and adverb and records various relations between synonym sets. A broad overview of the different PWN applications such as "Machine Translation", "Information Retrieval", "Document Classification", "Query Answering" and "Conceptual Identification" have been presented in (Morato et al., 2004).

PWN was created and maintained since 1990s. After this WordNet for other languages have been under development and new projects start every year. PWN database contains about 150000 words organized in over 115000 synsets.

Manual construction of WordNet is a time consuming task and requires linguistic knowledge. A number of automatic methods were proposed for constructing WordNet for other languages that use PWN and other existing lexical resources. In order to help the development of WordNets for other languages rather than English, especially for European one, a project named EuroWordNet was found (Vossen, 1999), in which a number of automatic methods for construction of such databases were proposed (Farreres et al., 1998).

There have been some other efforts to create a WordNet for Persian language (Famian, 2007; Rouhizadeh et al., 2008; Shamsfard, 2008) but there exists no Persian WordNet yet that covers all Persian words in dictionary and comparable with PWN. These projects have tried to construct Persian WordNet in the manually or semi automatic manner. In (Shamsfard, 2008) a semi automatic method is proposed in which for each Persian word a number of PWN synsets are suggested by system in order to be supervised by a human annotator to select a relevant synset. Based on experiments mentioned by Shamsfard (2008), the proposed WordNet extracted automatically by the system, retrieved about 70% accuracy.

In this paper a fully automatic method for constructing a large-scale Persian WordNet from available resource such as PWN, MRDs and corpora has been proposed. Our approach uses different word similarity metrics like mutual information and WordNet similarity to map Persian words to appropriate PWN synsets.

2 Related Works

In the related field of automatic and semi automatic WordNet construction, several efforts

Coling 2008: Poster Volume, pages 846–850,
Beijing, August 2010

have been made. In (Shamsfard, 2008) a semi automatic method has been used for developing a lexical ontology called FarsNet for Persian language. About 1500 verbs and 1500 nouns have been gathered manually to make WorldNet's core. Then some heuristics and Word Sense Disambiguation (WSD) methods have been used to find the most likely related Persian synsets.

According to the first heuristic, a Persian word has only one synset if it's be translated to a single English word. In this case no ambiguity exists for the Persian word whose one of synsets will be equivalent with that of English word. In other cases, second heuristic is used: if two translations of a Persian word have only one common synset then for the Persian word this common synset is selected. The existence of a single common synset in fact implies the existence of a single common sense between the two words and therefore their Persian translations shall be connected to this synset (Shamsfard, 2008). For words whose English translations have more than one synset and second heuristic cannot find the appropriate synset, WSD methods have been used to select correct synset. For each candidate synset, a score is calculated using the measure of semantic similarity and synset gloss words. Manual evaluation of the proposed automatic method in this research shows 70% correctness and covers about 6500 Entries on WordNet.

In (Sathapornrungkij and Pluempitiwiriyawej, 2005) a semi-automatic approach has been described to construct the Thai WordNet lexical database from WordNet and LEXiTRON machine readable dictionaries. Thai WordNet synsets have been derived from the PWN. The candidate links between Thai words and synsets have been derived from semantic links which are obtained from WordNet and the translation links which are obtained from LEXiTRON. In order to derive links between Thai words and PWN synsets, 13 criteria have been used which are categorized into three groups: monosemic, polysemic and structural criteria. Monosemic criteria focus on an English word which has only one meaning. Such English word has one synset in PWN. Polysemic criteria focus on an English word which has multiple meaning. Such English word has multiple synset in PWN. Structural criteria focus on the structural relations among synsets with respect to WordNet 1.7. In order to verify links that constructed using these 13 criteria, stratified sampling technique has been applied and for each criterion 400 links have been verified manually. The results of verification show that the best criterion has 92% correctness and the lowest correctness is equal 49.25%.

In PWN, there is a gloss for each synset that can be used in automatic WordNet construction. In (Kaji and Watanabe, 2006) this information has been used for automatic construction of Japanese WordNet. Given an English synset, it calculates the score for each of its Japanese translation candidates according to the gloss appended to the synset. A pair of words is called associated if mutual information between them be larger than a threshold. The score is defined as the sum of correlations between the translation candidate and the associated words appearing in the gloss. Whereas availability of bilingual corpora is limited, for calculating pair wise correlation between the Japanese translations of an English word and its associated words an iteratively approach has been proposed that calculate this correlation without using bilingual corpora.

In (Lee et al., 2000) a set of automatic WSD techniques have been described for linking Korean words collected from a bilingual MRD to PWN synsets. For a given synset, 6 individual heuristic scores are calculated and then a decision tree is used to combine these scores to classify the synset as linking or discarding. In order to make the decision tree, a set of synsets have been labeled manually as linking or discarding and corresponding heuristic scores have been calculated and then used for training data set. To evaluate the accuracy of proposed method the candidate synsets of 3260 senses of Korean words have been classified manually as linking or discarding. This test set has been used to calculate precision of each heuristic. The results of experiments show that the precision of all heuristics is better than random mapping and the best heuristic have 75.21% precision. The combination of heuristics using decision tree shows 93.59% precision.

3 Automatic Persian WordNet Construction

Each Persian word can have several English translations and each English translation has also

several PWN synsets. For a given Persian word, a bilingual dictionary is used to extract English equivalent words, and then a set of candidate synset is generated using PWN that contains all synsets of English translations of Persian word. As in (Shamsfard, 2008), if the English translation of a given Persian word has only one synset in PWN, then the Persian word is linked to this PWN synset directly, or if for a candidate synset at least two English translations belong to it, then Persian word is linked to this PWN synset.

In other cases, a score is calculated for each remaining candidate synset and the synset with maximum score is selected as an appropriate synset of the Persian word. Note that after selecting a synset, all synsets that share English words are removed from candidate synsets.

The following resources have been used in the process of score calculation:

- PWN: synset words, synset definition and hypernymy relations have been used.
- Bilingual dictionary (Persian – English)
- Raw Persian text corpus for extracting related words of a given Persian word
- Raw English text corpus for extracting mutual information between English words

Text corpora have been used to extract the related words of any given word. To do this, Mutual Information (MI) metric between any words in corpus and given Persian word are calculated and n-best words with higher MI values are selected. Mutual Information of pair x and x' is defined as follows:

$$MI(x,x') = \frac{n(x,x')}{n(x) * n(x')} * N \qquad (1)$$

In formula 1, n(x, x') is co-occurrence frequency of x and x' in corpus. This frequency is calculated using a window with specific size. n(x) is the frequency of word x in corpus and N is the number of unique words in corpus.

So, in order to select the most related words for a given Persian word, an additional step is considered. For each Persian word *w*, other related Persian words with highest mutual information are selected and considered as a set $R = \{r_1, r_2, \ldots, r_n\}$. Then for each Persian word r_i a similar process is used and a set of words is extracted that is called R_i. If R_i contains the word *w*, then r_i is selected as the related word for *w* and otherwise discarded.

After extracting the related words of the given Persian word, a Persian to English dictionary has been used to find equivalent translation of each related word. These words are referred as Related Translation Set (RTS). In scoring algorithm words that appear in gloss of each synset and words that appear in hypernym synset are called Gloss Words (GW). These words are considered as related words to the candidate synset and distinguish each synset from other.

Now for each candidate synset of a given Persian word a score is calculated that is based on the idea that two related words in the two-side languages share the same words in the correlation set. That is, if Persian word *w* relates to English synset *e*, then other co-related Persian words $r_1, r_2, \ldots, r_n$ which have gotten the best MI respect to *w*, should be related to the same synset *e* again.

Based on the above notion, the score of each candidate synset S can be estimated as follow:

$$Score(S) = \sum_{w_i \in RTS} \sum_{e_i \in GW} Sim(w_i, S) * MI(w_i, e_i) \qquad (2)$$

The score of synset S is defined as summation on product of semantic similarity between words in RTS and synset S, and mutual information between words in RTS and words in GW. In (Pedersen et al., 2004) several methods for calculating semantic similarity based on WordNet's structure have been presented. Some of these methods are based on path lengths between concepts and some of them are based on information content. One of these methods is named path in which for each word *w* and synset *s* is defined as inverse of shortest path length between any synset of *w* and *s*. In our experiments the measure path has been used and calculated using formula 3.

$$Sim(w,S) = \frac{1}{\min_{s_i \in \text{ synsets of w}}(path(s_i,S))} \qquad (3)$$

In formula 2 the words from RTS which has less similarity to synset *s* has little effect on the amount of score in synset.

4 Experiments and Evaluations

Persian WordNet constructor components are Word Translator, Related Word Extractor, Synset Extractor and Synset Selector. Persian words and their selected synsets are input and output of this system. Persian word is given as input to the Word Translator and Related Word Extractor components. In our experiment, 10 words with highest MI to the given Persian word are extracted using Related Word Extractor. For this purpose, 3000 documents of IRNA[1] newspaper text corpus have been used. IRNA is a news agency published their news on different languages, mainly on Persian. In order to count the number of co-occurrences of words x and x', a window with the size of 20 words was considered. Translations of related words and candidate synsets are given to Synset Selector and appropriate synsets for the given Persian word are selected. In this step PWN is used for semantic similarity calculation and an English text corpus (USENET corpus) is used to calculate mutual information. Table 1 shows the number of words and documents in the Persian and English text corpora. About 30698 Persian words from Aryanpour[2] Persian to English dictionary has been used for constructing Persian WordNet.

	Num of documents	Num of Unique Words
Persian	3000	32197
English	3000	32899

Table 1: number of documents and unique words in Persian and English corpuses

As it was mentioned in the previous section, Persian words were linked to PWN synsets in the two different ways. Some links was selected directly without calculating their score by using some heuristics. We call these links as unambiguous links. Some of these links are shown in table 2. As it shown in the table, unambiguous links are wrong in some cases. For example in the case of '<barchasb>tag', a verb synset is selected while the Persian word is noun, so the selection is judged as incorrect. If the part of speech tag information of word is used in this example the correct synset would be selected.

Another type of links are ambiguous links, in which a scoring method is used for selecting the appropriate synset. Examples of these links are shown in table 3. As it's shown in the table, the word '<karmozd>commission' has been linked to 6th sense of word 'commission' that is wrong. In constructed Persian WordNet also word '<farman>commission' has been linked to this sense of word 'commission' but the word '<karmozd>commission' and the word '<farman>commission' have less similarity together. In this example link between '<farman>commission' and 6th sense of word 'commission' is an unambiguous link. Therefore we can avoid of selecting this synset for '<karmozd>commission' using this information.

In order to evaluate the quality of the selected links, 500 Persian words have been randomly selected and the accuracy of selected synsets has been evaluated manually. Table 4 summarizes the results of this evaluation. As it's shown in the table, the precision of unambiguous links is about 95.8% while this precision is 76.4% for the ambiguous links. The weighted average precision of the whole links in our automatically generated Persian WordNet is 82.6%, which outperforms the only comparable semi-automated Persian WordNet which was previously presented by (Shamsfard, 2008), about 12.5%. Also, by comparing the PWN coverage rate of these Persian WordNets, it reveals that our result covered 29716 entries on PWN which it is about 4 times more than the previously generated Persian WordNet.

	Precision
Unambiguous links	95.8%
Ambiguous links	76.4%
All links	82.6%

Table 4: accuracy of selected links for 500 words

The experimental results reveal that in PWN there is a short gloss for some synsets which makes the calculated score for those synsets to be lower than other candidate synsets of a given Persian word. This problem can be overcome by normalizing the scores of candidate synset of a given Persian word, i.e. by dividing the score of each synset by the number of words in GW. Another solution of this problem is proposed by (Kaji and Watanabe, 2006). In (Kaji and Wata-

[1] Islamic Republic News Agency (http://www.irna.ir)
[2] http://www.aryanpour.com/

Persian word	English translation	Selected synset	Gloss	Correct /incorrect
<mosen> aged	aged, elderly, old	aged, elderly	people who are old collectively	correct
<barchasb> tag	tag, label, mark	tag, label, mark	attach a tag or label to	incorrect

Table 2: Examples of unambiguous links

Persian word	English translation	Selected synset	Gloss	Correct /incorrect
<enteshar> publication	publication	publication	the communication of something to the public; making information generally known	correct
<karmozd> commission	commission	commission, charge, direction	a formal statement of a command or injunction to do something	incorrect

Table 3: Examples of ambiguous links

nabe, 2006), the gloss is given as a query to text retrieval engine and the words that appear as the answer of this query are used instead of the words of gloss. In our experiments, the first solution is chosen which retrived the results shown in table 4.

5 Conclusion

This paper explored a method for automatically linking WordNet synsets to Persian words using pre-existing lexical resources such as Persian and English text corpora and PWN. The proposed method calculates a score for each candidate synset of a given Persian word and selects the synset with maximum score to be linked to the Persian word. This score is calculated considering related words of Persian word and words that appear in gloss of synset. A preliminary experiment shows that this method can be used to construct Persian WordNet. In the proposed method for each Persian word synsets with maximum calculated score are selected without considering other Persian words. In future work we intend to adapt our method and contribute other Persian word in order to select a synset for a given Persian word.

References

Alexin, Z., Csirik, j., Szarvas, G., Kocsor, A., Mihältz, M. (2006). *Construction of the Hungarian EuroWordNet Ontology and its Application to Information Extraction.* In Proceedings of the Third International WordNet Conference, Seogwipo, Jeju Island, Korea, pages 291-293.

Famian, A. A. (2007). *Towards Building a WordNet for Persian Adjectives.* In Proceedings of the 3rd Global wordnet conference, pages 307-309.

Farreres, X., Rigau, G., Rodríguez, H. (1998). *Using WordNet for Building WordNets.* In Proceedings of COLING/ACL Workshop on the Usage of WordNet in Natural Language Processing Systems, pages 65-72 .

Kaji, H., Watanabe, M. (2006). *Automatic construction of Japanese WordNet.* In Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC'06), Genoa, Italy, May 2006.

Lee, C., Lee, G., JungYun, S. (2000). *Automatic WordNet mapping using word sense disambiguation.* In the Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC 2000), Hong Kong, pages 142-147.

Pedersen, T., Patwardhan, S., Michelizzi, J. (2004). *WordNet::Similarity - Measuring the Relatedness of Concepts.* In AAAI, pages 1024-1025.

Rouhizadeh, M.,Shamsfard M., Yarmohammadi, M. (2008). *Building a WordNet for Persian Verbs.* The Fourth Global WordNet Conference, Hungary, pages 406- 412.

Sathapornrungkij, P., Pluempitiwiriyawej, C. (2005). *Construction of Thai WordNet lexical database from machine readable dictionaries.* Conference Proceedings: the tenth Machine Translation Summit, Thailand, pages 87-92.

Shamsfard, M. (2008). *Towards Semi Automatic Construction of a Lexical Ontology for Persian.* In Proceedings of the Sixth International Language Resources and Evaluation (LREC'08), Marrakech, Morocco.

Vossen, P. (1999). *EuroWordNet General Document.* Version 3 Final University of Amsterdam EuroWordNet LE2-4003, LE4-8328 .

Morato, J., Marzal, M., Lloréns, J., Moreiro, J. (2004). *WordNet Applications.* In Proceedings of the Second Global WordNet Conference, Masaryk University, pages 270–278.

Imbalanced Classification Using Dictionary-based Prototypes and Hierarchical Decision Rules for Entity Sense Disambiguation

Tingting Mu
National Centre for Text Mining
University of Manchester
tingting.mu@man.ac.uk

Xinglong Wang
National Centre for Text Mining
University of Manchester
xinglong.wang@man.ac.uk

Jun'ichi Tsujii
Department of Computer Science
University of Tokyo
tsujii@is.s.u-tokyo.ac.jp

Sophia Ananiadou
National Centre for Text Mining
University of Manchester
Sophia.Ananiadou@man.ac.uk

Abstract

Entity sense disambiguation becomes difficult with few or even zero training instances available, which is known as imbalanced learning problem in machine learning. To overcome the problem, we create a new set of reliable training instances from dictionary, called dictionary-based prototypes. A hierarchical classification system with a tree-like structure is designed to learn from both the prototypes and training instances, and three different types of classifiers are employed. In addition, supervised dimensionality reduction is conducted in a similarity-based space. Experimental results show our system outperforms three baseline systems by at least 8.3% as measured by macro F_1 score.

1 Introduction

Ambiguities in terms and named entities are a challenge for automatic information extraction (IE) systems. The problem is particularly acute for IE systems targeting the biomedical domain, where unambigiously identifying terms is of fundamental importance. In biomedical text, a term (or its abbreviation (Okazaki et al., 2010)) may belong to a wide variety of semantic categories (e.g., gene, disease, etc.). For example, *ER* may denote protein *estrogen receptor* in one context, but cell subunit *endoplasmic reticulum* in another, not to mention it can also mean emergency room. In addition, same terms (e.g., protein) may belong to many model organisms, due to the nomenclature of gene and gene products, where genes in model organisms other than human are given, whenever possible, the same names as their human orthologs (Wain et al., 2002). On the other hand, public biological databases keep species-specific records for the same protein or gene, making species disambiguation an inevitable step for assigning unique database identifiers to entity names in text (Hakenberg et al., 2008; Krallinger et al., 2008).

One way to entity disambiguation is classifying an entity into pre-defined semantic categories, based on its context (e.g., (Bunescu and Paşca, 2006)). Existing classifiers, such as maximum entropy model, achieved satisfactory results on the "majority" classes with abundant training instances, but failed on the "minority" ones with few or even zero training instances, i.e., the knowledge acquisition bottleneck (Agirre and Martinez, 2004). Furthermore, it is often infeasible to create enough training data for all existing semantic classes. In addition, too many training instances for certain majority classes lead to increased computational complexity for training, and a biased system ignoring the minority ones. These correspond to two previously addressed difficulties in imbalanced learning: *"... either (i) you have far more data than your algorithms can deal with,*

Coling 2008: Poster Volume, pages 851–859,
Beijing, August 2010

and you have to select a sample, or (ii) you have no data at all and you have to go through an involved process to create them" (Provost, 2000). Given an entity disambiguation task with imbalanced data, this paper explores how to create more informative training instances for minority classes and how to improve the large-scale training for majority classes.

Previous research has shown that words denoting class information in the surrounding context of an entity can be an informative indicator for disambiguation (Krallinger et al., 2008; Wang et al., 2010). Such words are refered to as "cue words" throughout this paper. For example, to disambiguate the type of an entity, that is, whether it is a protein, gene, or RNA, looking at words such as *protein*, *gene* and *RNA* are very helpful (Hatzivassiloglou et al., 2001). Similarly, for the task of species disambiguation (Wang et al., 2010), the occurrence of *mouse p53* strongly suggests that *p53* is a *mouse* protein. In many cases, cue words are readily available in dictionaries. Thus, for the minority classes, instead of creating artificial training instances by commonly used sampling methods (Haibo and Garcia, 2009), we propose to create a new set of real training instances by modelling cue words from a dictionary, called dictionary-based prototypes. To learn from both the original training instances and the dictionary-based prototypes, a hierarchical classification system with a tree-like structure is designed. Furthermore, to cope with the large number of features representing each instance, supervised orthogonal locality preserving projection (SOLPP) is conducted for dimensionality reduction, by simultaneously preserving the intrinsic structures constructed from both the features and labels. A new set of lower-dimensional embeddings with better discriminating power is obtained and used as input to the classifier. To cope with the large number of training instances in some majority classes, we propose a committee machine scheme to accelerate training speed without sacrificing classification accuracy. The proposed method is evaluated on a species disambiguation task, and the empirical results are encouraging, showing at least 8.3% improvement over three different baseline systems.

2 Related Work

Construction of a classification model using supervised learning algorithms is popular for entity disambiguation. A number of researchers have tackled entity disambiguation in general text using wikipedia as a resource to learn classification models (Bunescu and Paşca, 2006). Hatzivassiloglou et al. (2001) studied disambiguating proteins, genes, and RNA in text by training various classifiers using entities with class information provided by adjacent cue words. Wang et al. (2010) proposed a "hybird" system for species disambiguation, which heuristically combines results obtained from classifying the context, and those from modeling relations between cue words and entities. Although satisfactory performance was reported, their system incurs higher computational cost due to syntactic parsing and the binary relation classifier.

Many imbalanced learning techniques, as reviewed by Haibo and Garcia (2009), can also be used to achieve the same purpose. However, to our knowledge, there is little research in applying these machine learning (ML) techniques to entity disambiguation. It is worth mentioning that although these ML techniques can improve the learning performance to some extent, they only consider the information contained in the original training instances. The created instances do not add new information, but instead utilize the original training information in a more sophisticated way. This motivates us to pursue a different method of creating new training instances by using information from a related and easily obtained source (e.g., a dictionary), similar to transfer learning (Pan and Yang, 2009).

3 Task and Corpus

In this work, we develop an entity disambiguation technique with the use of cue words, as well as a general ML algorithm for imbalanced classification using a set of newly created dictionary-based prototypes. These prototypes are represented with different features from those used by the original training instances. The proposed method is evaluated on a species disambiguation task: given a text, in which mentions of biomedical named en-

tities are annotated, we assign a *s*pecies identifier to every entity mention. The types of entities studied in this work are genes and gene products (e.g., proteins), and we use the NCBI Taxonomy[1] (taxon) IDs as species tags and to build the prototypes. Note that this paper focuses on the task of species disambiguation and makes the assumption that the named entities are already recognised.

Consider the following sentence as an example: if one searches the proteins (i.e., the underlined term) in a protein database, he/she will find they belong to many model organisms. However, in this particular context, CD200R-CD4d3+4 is *human* and *mouse* protein, while *r*CD4d3+4 is a *rat* one.[2] We call such a task of assigning species identifiers to entities, according to context, as species disambiguation.

> The amounts of *human* and *mouse* CD200R-CD4d3+4 and *r*CD4d3+4 protein on the microarray spots were similar as visualized by the red fluorescence of OX68 mAb recognising the CD4 tag present in each of the recombinant proteins.

The informative cue words (e.g., *mouse*) used to help species disambiguation are called species words. In this work, species words are defined as any word that indicates a model organism and also appears in the organism dictionaries we use. They may have various parts-of-speech, and may also contain multiple tokens (despite the name species *word*). For example, "human", "mice", "bovine" and "E. Coli" are all species words. We detect these words by automatic dictionary lookup: a word is annotated as a species word if it matches an entry in a list of organism names. Each entry in the list contains a species word and its corresponding taxon ID, and the list is merged from two dictionaries: the NCBI Taxonomy and the UniProt controlled vocabulary of species.[3] The NCBI portion is a flattened NCBI Taxonomy (i.e., without hierarchy) including only the identifiers of *genus* and *species* ranks. In total, the merged list contains 356,387 unique species words and 272,991 unique species IDs. The ambiguity in species words is low: 3.86% of species words map to multiple IDs, and on average each word maps to 1.043 IDs.

The proposed method was evaluated on the corpus developed in (Wang et al., 2010), containing $6,223$ genes and gene products, each of which was manually assigned with either a taxon ID or an "Other" tag, with *human* being the most frequent at 50.30%. With the extracted features and the species ID tagged by domain experts, each occurrence of named entities can be represented as a d-dimensional vector with a label. Species disambiguation can be modelled as a multi-classification task: Given n training instances $\{\boldsymbol{x}_i\}_{i=1}^n$, their $n \times d$ feature matrix $\mathbf{X} = [x_{ij}]$ and n-dimensional label vector $\boldsymbol{y} = [y_1, y_2, \ldots, y_n]^T$ are used to train a classifier $\mathfrak{C}(\cdot)$, where $\boldsymbol{x}_i = [x_{i1}, x_{i2}, \ldots, x_{id}]^T$, $y_i \in \{1, 2, \ldots, c\}$, and c denotes the number of existing species in total. Given m different query instances $\{\hat{\boldsymbol{x}}_i\}_{i=1}^m$, their $m \times d$ feature matrix $\hat{\mathbf{X}} = [\hat{x}_{ij}]$ are used as the input to the trained classifier, so that their labels can be predicted by $\{\mathfrak{C}(\hat{\boldsymbol{x}}_i)\}_{i=1}^m$.

We used relatively simple contextual features because this work was focused on developing a ML framework. In more detail, we used the following features: 1) 200 words surrounding the entity in question; 2) two nouns and two adjectives at the entity's left and right; 3) 5 species words at the entity's left and right. In addition, function words and words that consist of only digits and punctuations are filtered out. The final numerical dataset consists of 6,227 instances, each represented by 16,851 binary features and belonging to one of the 13 classes. The dataset is highly imbalanced: among the 13 classes, the numbers of instances in the four majority classes vary from 449 to 3,220, while no more than 20 instances are contained in the eight minority classes (see Table 1).

[1] http://www.ncbi.nlm.nih.gov/sites/entrez?db= taxonomy

[2] Prefix 'r' in "rCD4d3+4" indicates that it is a *rat* protein.

[3] http://www.expasy.ch/cgi-bin/speclist

4 Proposed Method

4.1 Dictionary-based Prototypes

For each existing species, we create a b-dimensional binary vector, given as $\boldsymbol{p}_i = [p_{i1}, p_{i2}, \ldots, p_{ib}]^T$, using b different species words listed in the dictionary as features, which is called dictionary-based prototype. The binary value p_{ij} denotes whether the jth species word belongs to the ith species in the dictionary. This leads to a $c \times b$ feature matrix $\mathbf{P} = [p_{ij}]$ for c species.

Considering that the species words preceding and appearing in the same sentence as an entity can be informative indicators for the possible species of this entity, we create two more $m \times b$ binary feature matrices for the query instances with the same b species words as features: $\hat{\mathbf{X}}_1 = [\hat{x}_{ij}^{(1)}]$ and $\hat{\mathbf{X}}_2 = [\hat{x}_{ij}^{(2)}]$, where $\hat{x}_{ij}^{(1)}$ denotes whether the jth species word is the preceding word of the ith entity, and $\hat{x}_{ij}^{(2)}$ denotes whether the jth species word appears in the same sentence as the ith entity but is not preceding word. Thus, the similarity between each query entity and existing species can be simply evaluated by calculating the inner-product between the entity instance and the corresponding prototype. This leads to the following $m \times c$ similarity matrix $\hat{\mathbf{S}} = [\hat{s}_{ij}]$:

$$\hat{\mathbf{S}} = \theta \hat{\mathbf{X}}_1 \mathbf{P}^T + (1-\theta)\hat{\mathbf{X}}_2 \mathbf{P}^T, \qquad (1)$$

where $0 \leq \theta \leq 1$ is a user-defined parameter controlling the degree of indicating reliability of the preceding word and the same-sentence word. The $n \times c$ similarity matrix $\mathbf{S} = [s_{ij}]$ between the training instances and the species can be constructed in exactly the same way. Based on empirical experience, the preceding word indicates the entity's species more accurately than the same-sentence word. Thus, θ is preferred to be set as greater than 0.5. The obtained similarity matrix will be used in the nearest neighbour classifier (see Section 4.2.1).

Both the original training instances $\mathbf{X}$ and the newly created prototypes $\mathbf{P}$ are used to train the proposed hierarchical classification system. Subject to the nature of the classifier employed, it is convenient to construct one single feature matrix instead of using $\mathbf{X}$ and $\mathbf{P}$ individually. Aiming at keeping the same similarity values between each entity instance and the species prototype, we construct the following $(n+c) \times (d+b)$ feature matrix for both the training instances and prototypes:

$$\mathbf{F} = \begin{bmatrix} \mathbf{X} & \theta\mathbf{X}_1 + (1-\theta)\mathbf{X}_2 \\ \mathbf{0} & \mathbf{P} \end{bmatrix}, \qquad (2)$$

where $\mathbf{X}_1$ and $\mathbf{X}_2$ are constructed in the same way as $\hat{\mathbf{X}}_1$ and $\hat{\mathbf{X}}_2$ but for training instances. Their corresponding label vector is $\boldsymbol{l} = [\boldsymbol{y}^T, 1, 2, \ldots, c]^T$.

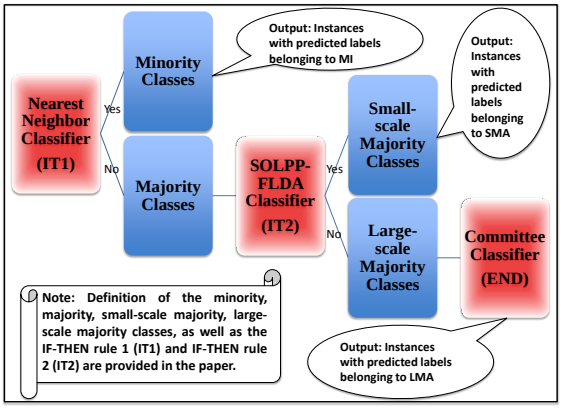


Figure 1: Structure of the proposed hierarchical classification system

4.2 Hierarchical Classification

Multi-stage or hierarchical classification (Giusti et al., 2002; Podolak, 2007; Kurzyński, 1988) is widely used in many complex multi-category classification tasks. Existing research shows such techniques can potentially achieve right trade-off between accuracy and resource allocation (Giusti et al., 2002; Podolak, 2007). Our proposed hierarchical system has a tree-like structure with three different types of classifier at nodes (see Figure 1). Different classes are organized in a hierarchical order to be classified based on the corresponding numbers of available training instances. Letting n_i denote the number of training instances available in the $i$the class excluding the created prototypes, we categorize the classes as follows:

- **Minority Classes (MI):** Classes with less training instances than the threshold: MI $= \{i : \frac{n_i}{n} < \sigma_1, i \in \{1, 2 \ldots, c\}\}$.

- **Majority Classes (MA):** Classes with more training instances than the threshold: MA $= \{i : \frac{n_i}{n} \geq \sigma_1, i \in \{1, 2 \ldots, c\}\}$.
- **Small-scale Majority Classes (SMA):** Majority Classes with less training instances than the threshold: SMA $= \{i : \frac{n_i}{n} < \sigma_2, i \in \text{MA}\}$.
- **Large-scale Majority Classes (LMA):** Majority Classes with more training instances than the threshold: LMA $= \{i : \frac{n_i}{n} \geq \sigma_2, i \in \text{MA}\}$.

Here, $0 < \sigma_1 < 1$ and $0 < \sigma_2 < 1$ are size thresholds set by users. We have MI $\cap$ MA $= \emptyset$, SMA $\cap$ LMA $= \emptyset$, and SMA $\cup$ LMA $=$ MA.

The tree-like hierarchical structure of our system is determined by MI, MA, SMA, and LMA. We propose two IF-THEN rules to control the system: Given a query instance $\hat{\boldsymbol{x}}_i$, the level 1 classifier $\mathfrak{C}_1$ is used to predict whether $\hat{\boldsymbol{x}}_i$ belongs to MA or a specific class in MI, which wer call IF-THEN rule 1 (IT1). If $\hat{\boldsymbol{x}}_i$ belongs to MA, the level 2 classifier $\mathfrak{C}_2$ is used to predict whether $\hat{\boldsymbol{x}}_i$ belongs to LMA or a specific class in SMA, called IF-THEN rule 2 (IT2). If $\hat{\boldsymbol{x}}_i$ belongs to LMA, the level 3 classifier $\mathfrak{C}_3$ finally predicts the specific class in LMA $\hat{\boldsymbol{x}}_i$ belongs to. We explain in the following sections how the classifiers $\mathfrak{C}_1$, $\mathfrak{C}_2$, and $\mathfrak{C}_3$ work in detail.

4.2.1 Nearest Neighbour Classifier

The goal of the nearest neighbour classifier, denoted by $\mathfrak{C}_1$, is to decide whether the nearest-neighbour prototype of the query instance belongs to MI. The only used training instances are our created dictionary-based prototypes $\{\boldsymbol{p}_i\}_{i=1}^{c}$ with the label vector $[1, 2, \ldots, c]^T$. The nearest-neighbour prototype of the query instance $\hat{\boldsymbol{x}}_i$ possesses the maximum similarity to $\hat{\boldsymbol{x}}_i$:

$$NN(\hat{\boldsymbol{x}}_i) = \arg \max_{j=1,2,\ldots,c} \hat{s}_{ij}, \tag{3}$$

where $\hat{s}_{ij}$ is obtained by Eq. (1). Consequently, the output of the classifier $\mathfrak{C}_1$ is given as

$$\mathfrak{C}_1(\hat{\boldsymbol{x}}_i) = \begin{cases} NN(\hat{\boldsymbol{x}}_i), & \text{If } NN(\hat{\boldsymbol{x}}_i) \in \text{MI}, \\ 0, & \text{Otherwise.} \end{cases} \tag{4}$$

The IF-THEN rule 1 can then be expressed as

$$\text{Action}^{(\text{IT1})} = \begin{cases} \text{Go to } \mathfrak{C}_2, & \text{If } \mathfrak{C}_1(\hat{\boldsymbol{x}}_i) = 0, \\ \text{Stop}, & \text{Otherwise.} \end{cases}$$

4.2.2 SOLPP-FLDA Classifier

The goal of the SOLPP-FLDA classifier, denoted by $\mathfrak{C}_2$, is to predict whether the query instance belongs to LMA or a specific class in SMA. In this classifier, the used training instances are the original training entities and the dictionary-based prototypes, both belonging to MA. The feature matrix $\mathbf{F}$ and the label vector $\boldsymbol{l}$ defined in Section 4.1 are used, but with instances from MI removed (we use $\tilde{n}$ to denote the number of remaining training instances, and the same symbol $\mathbf{F}$ for feature matrix). The used label vector $\tilde{\boldsymbol{l}}$ to train $\mathfrak{C}_2$ should be re-defined as $\tilde{l}_i = l_i$ if $l_i \in$ SMA, and 0 otherwise.

First, we propose to implement orthogonal locality preserving projection (OLPP) (Kokiopoulou and Saad, 2007) in a supervised manner, leading to SOLPP, to obtain a smaller set of more powerful features for classification. Also, we conduct SOLPP in a similarity-based feature space computed from $(d + 2b)$ original features by employing dot-product based similarity, given by $\mathbf{F}\mathbf{F}^T$. As explained later, to compute the new features from $\mathbf{F}\mathbf{F}^T$ instead of the original features $\mathbf{F}$ achieves reduced computational cost. An $\tilde{n} \times k$ projection matrix $\mathbf{V} = [v_{ij}]$ is optimized in this n-dimensional similarity-based feature space. The optimal projections are obtained by minimizing the weighted distances between the lower-dimensional embeddings so that "similar" instances are mapped together in the projected feature space. Mathematically, this leads to the following constrained optimization problem:

$$\min_{\substack{\mathbf{V} \in R^{\tilde{n} \times k}, \\ \mathbf{V}^T\mathbf{V} = \mathbf{I}_{k \times k}}} \text{tr}[\mathbf{V}^T\mathbf{F}^T\mathbf{F}(\mathbf{D} - \mathbf{W})\mathbf{F}\mathbf{F}^T\mathbf{V}], \tag{5}$$

where $\mathbf{W} = [w_{ij}]$ denotes the $n \times n$ weight matrix with w_{ij} defining the degree of "closeness" or "similarity" between the ith and jth instances, $\mathbf{D}$ is a diagonal matrix with $\{d_i = \sum_{j=1}^{\tilde{n}} w_{ij}\}_{i=1}^{\tilde{n}}$ as the diagonal elements.

Usually, the weight matrix $\mathbf{W}$ is defined by an adjacency graph constructed from the original

data, e.g. for OLPP. One common way to define the adjacency is by including the K-nearest neighbors (KNN) of a given node to its adjacency list, which is also called the KNN-graph (Kokiopoulou and Saad, 2007). There are two common ways to define the weight matrix: constant value, where $w_{ij} = 1$ if the ith and jth samples are adjacent, while $w_{ij} = 0$ otherwise, and Gaussian kernel. We will denote in the rest of the paper such a weight matrix computed only from the features as $\mathbf{W}_X$. Ideally, if the features can accurately describe all the discriminating characteristics, the samples that are close or similar enough to each other should have the same label vectors. However, when processing real dataset, what may happen is that, in the d-dimensional feature space, the data points that are close to each other may belong to different categories, while on the contrary, the data points that are in a distant to each other may belong to the same category. In the k-dimensional projected feature space obtained by OLPP, one may have the same problem. Because OLPP solves the constrained optimization problem in Eq. (5) using $\mathbf{W}_X$: if two instances are close or similar to each other in the original feature space, they will be the same close or similar to each other in the projected space. To solve this problem, we decide to modify the "closeness" or "similarity" between instances in the projected feature space by considering the label information. The following computation of a supervised weight matrix is used for our SOLPP:

$$\mathbf{W} = (1-\alpha)\mathbf{W}_X + \alpha\mathbf{L}\mathbf{L}^T, \tag{6}$$

where $0 \leq \alpha \leq 1$ is a user-defined parameter controlling the tradeoff between the label-based and feature-based neighborhood structures, and $\mathbf{L} = [l_{ij}]$ is an $\tilde{n} \times c$ binary label matrix with $l_{ij} = 1$ if the ith instance belongs to the jth class, and $l_{ij} = 0$ otherwise.

The optimal solution of Eq. (5) is the top $(k+1)$th eigenvectors of the $\tilde{n} \times \tilde{n}$ symmetric matrix $\mathbf{F}^T\mathbf{F}(\mathbf{D} - \mathbf{W})\mathbf{F}\mathbf{F}^T$, corresponding to the $k+1$ smallest eigenvalues, but with the top one eigenvector removed, denoted by $\mathbf{V}^*$. It is worth to mention that if the original feature matrix $\mathbf{F}$ is used as the input of SOLPP, one needs to compute the eigen-decomposition of the $(d+b) \times (d+b)$ symmetric matrix $\mathbf{F}^T(\mathbf{D} - \mathbf{W})\mathbf{F}$. The corresponding computation complexity increases in $O((d+b)^3)$, which is unacceptable in practical when $d + b \gg \tilde{n}$. The projected features for the training instances are computed by

$$\mathbf{Z} = \mathbf{F}\mathbf{F}^T\mathbf{V}^*. \tag{7}$$

Given a different set of m query instances with an $m \times (d+b)$ feature matrix,

$$\hat{\mathbf{F}} = [\hat{\mathbf{X}},\ \theta\hat{\mathbf{X}}_1 + (1-\theta)\hat{\mathbf{X}}_2], \tag{8}$$

their embeddings can be easily obtained by

$$\hat{\mathbf{Z}} = \hat{\mathbf{F}}\hat{\mathbf{F}}^T\mathbf{V}^*. \tag{9}$$

Then, the projected feature matrix $\mathbf{Z}$ and label vector $\tilde{\boldsymbol{l}}$ are used to train a multi-class classifier. By employing the one-against-all scheme, different binary classifiers $\{\mathfrak{C}_i^{(2)}\}_{i \in \text{SMA} \cup \{0\}}$ with label space $\{+1, -1\}$ are trained. For the ith class ($i \in \text{SMA} \cup \{0\}$), the training instances belonging to it are labeled as positive, otherwise negative. In each binary classifier $\mathfrak{C}_i^{(2)}$, a separating function

$$f_i^{(2)}(\boldsymbol{x}) = \boldsymbol{x}^T\boldsymbol{w}_i^{(2)} + b_i^{(2)} \tag{10}$$

is constructed, of which the optimal values of the weight vector $\boldsymbol{w}_i^{(2)}$ and bias $b_i^{(2)}$ are computed using Fisher's linear discriminant analysis (FLDA) (Fisher, 1936; Mu, 2008). Finally, the output of the classifier $\mathfrak{C}_2$ can be obtained by assigning the most confident class label to the query instance $\hat{\boldsymbol{x}}_i$, with the confidence value indicated by the value of separating function:

$$\mathfrak{C}_2(\hat{\boldsymbol{x}}_i) = \arg \max_{j \in \text{SMA} \cup \{0\}} f_j^{(2)}(\hat{\boldsymbol{x}}_i). \tag{11}$$

The IF-THEN rule 2 can then be expressed as

$$\text{Action}^{(\text{IT2})} = \begin{cases} \text{Go to } \mathfrak{C}_3, & \text{If } \mathfrak{C}_2(\hat{\boldsymbol{x}}_i) = 0, \\ \text{Stop}, & \text{Otherwise.} \end{cases}$$

4.2.3 Committee Classifier

The goal of the committee classifier, denoted by $\mathfrak{C}_3$, is to predict the specific class in LMA the query instance belongs to. The used training

instances are entities and dictionary-based prototypes only belonging to LMA. With the same one-against-all scheme, there are large number of positive and negative training instances to train a binary classifier for a class in LMA. To accelerate the training procedure without sacrificing the accuracy, the following scheme is designed.

Letting n_e denote the number of experts in committee, all the training instances are averagely divided into n_e+1 groups each containing similar numbers of training instances from the same class. The instances in the ith and the $(i+1)$th groups are used to train the ith expert classifier. This achieves overlapped training instances between expert classifiers. The output value of $\mathfrak{C}_i^{(3)}$ is not the class index as used in $\mathfrak{C}_2$, but the value of the separating function of the most confident class, denoted by $f_i^{(3)}$. Different from the commonly used majority voting rule, we only trust the most confident expert. Thus, the output of $\mathfrak{C}_3$ for a query instance $\hat{\boldsymbol{x}}_i$ can be obtained by

$$\mathfrak{C}_3(\hat{\boldsymbol{x}}_i) = \arg \max_{j=1,2,...,n_e} f_j^{(3)}(\hat{\boldsymbol{x}}_i). \qquad (12)$$

By using $\mathfrak{C}_3$, different expert classifiers can be trained in parallel. The total training time is equal to that of the slowest expert classifier. The more expert classifiers are used, the faster the system is, however, the less accurate the system may become due to the decrease of used training instances for each expert, especially the positive instances in the case of imbalanced classification. This is also the reason we do not apply the committee scheme to SMA classes.

5 Experiments

5.1 System Evaluation and Baseline

We evaluate the proposed method using 5-fold cross validation, with around 4,980 instances for training, and 1,245 instances for test in each trial. We compute the F_1 score for each species, and employ macro- and micro- average scheme to compute performance for all species. Three baselines for comparison include:

- **Baseline 1 (B1)** : A maximum entropy model trained with training data only.
- **Baseline 1 (B2)** : Combination of B1 and the species dictionary using rules employed in Wang et al. (2010).
- **Baseline 2 (B3)**: The "hybrid" system combining B1, the dictionary and a relation model [4] using rules (Wang et al., 2010).

Our hierarchical classification system were implemented in two ways:

- **HC**: Only the training data on its own is used to train the system.
- **HC/D**: Both the training data and the dictionary-based prototypes are used to train the system.

5.2 Results and Analysis

The proposed system was implemented with $\theta = 0.8$, $\alpha = 0.8$, $n_e = 4$, and $k = 1000$. The species 9606, 10090, 7227, and 4932 were categorized as LMA, the species 10116 as SMA, and the rest species as MI. To compute the supervised weight matrix, the percentage of the used KNN in the KNN-graph was 0.6. Parameters were not fine tuned, but set based on our empirical experience on previous classification research. As shown in Table 1: HC and B1 were trained with the same instances and features, and HC outperformed B1 in both macro and micro F1. Both HC and B1 obtained zero F_1 scores for most minority species, showing that it is nearly impossible to correctly label the query instances of minority classes, due to lack of training data. By learning from a related resource, HC/D, B2, and B3 yielded better macro performance. In particular, while HC/D and B2 learned from the same dictionary and training data, HC/D outperformed B2 by 19.1% in macro and 2.5% in micro F1. B3 aimed at improving the macro performance by employing computationally expensive syntactic parsers and also by training an extra relation classifier. With the same goal, HC/D integrated the cue word information into the ML classifier in a more general way, and yielded an 8.3% improvement over B3, as measured by macro-F1.

[4] This is an SVM model predicting relations between entities and nearby species words with positive output indicates species words bear the semantic label of entities.

Species Name	Cat.	No.	HC	HC/D	B1	B2	B3
Homo sapiens (9606)	LMA	3220	87.39	**87.48**	86.06	85.43	86.48
Mus musculus (10090)	LMA	1709	79.99	79.98	79.59	80.00	**80.41**
Drosophila melanogaster (7227)	LMA	641	86.62	86.35	**87.96**	87.02	87.37
Saccharomyces cerevisiae (4932)	LMA	499	**90.24**	**90.24**	83.35	81.64	84.64
Rattus norvegicus (10116)	SMA	50	55.07	**69.23**	48.42	64.41	59.41
Escherichia coli K-12 (83333)	MI	18	0.00	0.00	0.00	0.00	0.00
Xenopus tropicalis (8364)	MI	8	0.00	40.00	0.00	**41.67**	36.36
Caenorhabditis elegans (6239)	MI	7	0.00	22.22	0.00	**28.57**	**22.22**
Oryctolagus cuniculus (9986)	MI	3	0.00	0.00	0.00	**20.00**	0.00
Bos taurus (9913)	MI	3	0.00	50.00	0.00	0.00	**100.00**
Arabidopsis thaliana (3702)	MI	2	0.00	0.00	0.00	0.00	**66.67**
Arthropoda (6656)	MI	1	0.00	**100.00**	0.00	50.00	0.00
Martes zibellina (36722)	MI	1	0.00	**50.00**	0.00	28.57	0.00
Micro-average	N/A	N/A	85.03	**85.13**	83.59	83.04	83.80
Macro-average	N/A	N/A	30.72	**51.96**	29.42	43.64	47.97

Table 1: Performance is compared in F1 (%), where "No." denotes the number of training instances and "Cat." denotes the category of species class as defined in Section 4.2.

6 Conclusions and Future Work

Disambiguating bio-entities presents a challenge for traditional supervised learning methods, due to the high number of semantic classes and lack of training instances for some classes. We have proposed a hierarchical framework for imbalanced learning, and evaluated it on the species disambiguation task. Our method automatically builds training instances for the minority or missing classes from a cue word dictionary, under the assumption that cue words in the surrounding context of an entity strongly indicate its semantic category. Compared with previous work (Wang et al., 2010; Hatzivassiloglou et al., 2001), our method provides a more general way to integrate the cue word information into a ML framework without using deep linguistic information.

Although the species disambiguation task is specific to bio-text, the difficulties caused by imbalanced frequency of different senses are common in real application of sense disambiguation. The proposed technique can also be applied to other domains, providing the availability of a cue word dictionary that encodes semantic information regarding the target semantic classes. Building such a dictionary from scratch can be challenging, but may be easier compared to manual annotation. In addition, such dictionaries may already exist in specialised domains.

Acknowledgment

The authors would like to thank the biologists who annotated the species corpus, and National Centre for Text Mining. Funding: Pfizer Ltd.; Joint Information Systems Committee (to UK National Centre for Text Mining)

References

Agirre, E. and D. Martinez. 2004. Unsupervised WSD based on automatically retrieved examples: The importance of bias. In *Proceedings of EMNLP*.

Bunescu, R. and M. Paşca. 2006. Using encyclopedic knowledge for named entity disambiguation. In *Proceedings of EACL*.

Fisher, R. A. 1936. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2):179–188.

Giusti, N., F. Masulli, and A. Sperduti. 2002. Theoretical and experimental analysis of a two-stage system for classification. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(7):893–904.

Haibo, H. and E. A. Garcia. 2009. Learning from imbalanced data. *IEEE Trans. on Knowledge and Data Engineering*, 21(9):1263–1284.

Hakenberg, J., C. Plake, R. Leaman, M. Schroeder, and G. Gonzalez. 2008. Inter-species normalization of gene mentions with GNAT. *Bioinformatics*, 24(16).

Hatzivassiloglou, V., PA Duboué, and A. Rzhetsky. 2001. Disambiguating proteins, genes, and RNA in text: a machine learning approach. *Bioinformatics*, 17(Suppl 1).

Kokiopoulou, E. and Y. Saad. 2007. Orthogonal neighborhood preserving projections: A projection-based dimensionality reduction technique. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 29(12):2143–2156.

Krallinger, M., A. Morgan, L. Smith, F. Leitner, L. Tanabe, J. Wilbur, L. Hirschman, and A. Valencia. 2008. Evaluation of text-mining systems for biology: overview of the second biocreative community challenge. *Genome Biology*, 9(Suppl 2).

Kurzyński, M. W. 1988. On the multistage bayes classifier. *Pattern Recognition*, 21(4):355–365.

Mu, T. 2008. *Design of machine learning algorithms with applications to breast cancer detection*. Ph.D. thesis, University of Liverpool.

Okazaki, N., S. Ananiadou, and J. Tsujii. 2010. Building a high quality sense inventory for improved abbreviation disambiguation. *Bioinformatics*, doi:10.1093/bioinformatics/btq129.

Pan, S. J. and Q. Yang. 2009. A survey on transfer learning. *IEEE Trans. on Knowledge and Data Engineering*.

Podolak, I. T. 2007. Hierarchical rules for a hierarchical classifier. *Lecture Notes in Computer Science*, 4431:749–757.

Provost, F. 2000. Machine learning from imbalanced data sets 101. In *Proc. of Learning from Imbalanced Data Sets: Papers from the Am. Assoc. for Artificial Intelligence Workshop*. (Technical Report WS-00-05).

Wain, H., E. Bruford, R. Lovering, M. Lush, M. Wright, and S. Povey. 2002. Guidelines for human gene nomenclature. *Genomics*, 79(4):464–470.

Wang, X., J. Tsujii, and S. Ananiadou. 2010. Disambiguating the species of biomedical named entities using natural language parsers. *Bioinformatics*, 26(5):661667.

A Vector Space Model for Subjectivity Classification in Urdu aided by Co-Training

Smruthi Mukund
CEDAR
University at Buffalo
smukund@buffalo.edu

Rohini K. Srihari
CEDAR
University at Buffalo
rohini@cedar.buffalo.edu

Abstract

The goal of this work is to produce a classifier that can distinguish subjective sentences from objective sentences for the Urdu language. The amount of labeled data required for training automatic classifiers can be highly imbalanced especially in the multilingual paradigm as generating annotations is an expensive task. In this work, we propose a co-training approach for subjectivity analysis in the Urdu language that augments the positive set (subjective set) and generates a negative set (objective set) devoid of all samples close to the positive ones. Using the data set thus generated for training, we conduct experiments based on SVM and VSM algorithms, and show that our modified VSM based approach works remarkably well as a sentence level subjectivity classifier.

1 Introduction

Subjectivity tagging involves distinguishing sentences that express opinions from sentences that present factual information (Banfield 1982; Wiebe, 1994). A wide variety of affective nuances can be used while delivering a message pertaining to an event. Although the factual content remains the same, lexical selections and grammatical choices can considerably influence the affective nature of the text. Recognizing sentences that exhibit affective behavior will require, at the least, recognizing the structure of the sentence and the emotion bearing words.

To date, much of the research in this area is focused on English. A variety of reliable resources that facilitate effective sentiment analysis and opinion mining, such as polarity lexicons (Senti-WordNet [1]) and contextual valence shifters (Kennedy and Inkpen, 2005) are available for English. The MPQA corpus of 10,000 sentences (Wiebe *et al.,* 2005) provides detailed annotations for sources of opinions, targets, speech events and fragments that indicate attitudes for the English newswire data. The IMDB corpus contains 10,000 sentences categorized as subjective and objective in the movie review domain. Clearly, English is well supported with resources. There are other widely spoken resource poor languages that are not as privileged. When we consider social media, limiting our analysis to a language like English, however universal, will lead to loss of information. With the advent of virtual keyboards and extended Unicode support, the internet is rapidly getting flooded by users who use their native language in textual communication. There is a pressing need to perform non-topical text analysis in the multilingual paradigm.

Subjectivity analysis is a precursor to numerous applications performing non-topical text analysis like sentiment analysis, emotion detection, and opinion extraction (Liu *et al.*, 2005; Ku *et al.,* 2006; Titov and McDonald, 2008). Creating the state-of-the-art subjectivity classifier using machine learning techniques require access to large amounts of annotated data. For less commonly taught languages like

[1] http://swn.isti.cnr.it/download_1.0/

Coling 2008: Poster Volume, pages 860–868,
Beijing, August 2010

Urdu, Hindi, Bengali, Spanish and Romanian, the resources required to automate subjectivity analysis are either very sparse or unavailable. Generating annotated corpus for subjectivity detection is laborious and time consuming.

However, several innovative techniques have been proposed by researchers in the past to generate annotated data and lexical resources for subjectivity analysis in resource poor languages. Mihalcea *et al.,* (2007) and Banea *et al.,* (2008) used machine translation technique to leverage English resources for analysis in Romanian and Spanish languages. Wan (2009) proposed a co-training technique that leveraged an available English corpus for Chinese sentiment classification. Wan (2008) focused on improving Chinese sentiment analysis by using both Chinese and English lexicons.

Unfortunately, not much work has been done in the area of subjectivity analysis for the Urdu language. This language lacks annotated resources required to generate even the basic NLP tools (POS tagger, NE tagger etc.) needed for text analysis. In order to facilitate subjectivity analysis in Urdu language, we annotated a small set of Urdu newswire articles for emotions (§2). The sentence level annotations provided in this dataset follow the annotation guidelines proposed by Wiebe *et al.,* (2003). Although tremendous effort was put into generating this corpus, the data set is not very comprehensive and contains only about 500 sentences marked subjective. This is definitely insufficient to train a suitable subjectivity classifier.

1.1 Issue with unbalanced data set

A subjectivity classifier is a binary classifier. A traditional binary classifier is trained using universal representative sets for positive and negative categories. But in subjectivity analysis, especially for languages like Urdu that have no annotated data, generating universal representative sets is extremely difficult and almost an impossible task. Assimilating the negative set is especially a delicate task as the set should be carefully pruned of all the positive samples. Also, detecting subjectivity in a sentence is highly personalized. Annotators are sometimes prejudiced while marking samples. This bias, however small, produces errors with some true positive samples being unintentionally missed and categorized as negative. Traditionally, research in machine learning has assumed the class distribution in the training data to be reasonably balanced. However, when the training data is highly imbalanced, i.e., the number of positive examples is very small, the performance of text classification algorithms such as linear support vector machine (SVM) (Brank and Grobelnik, 2003), naïve Bayes and decision trees (Kubat and Matwin, 1997) are adversely affected.

In order to achieve a balanced training set, Japkowicz (2000) duplicates positive examples (oversampling) and discards negative ones (downsizing). Kubat and Matwin (1997) discard all samples that are close to the positive set to avoid misclassification. Chan and Stalfo (1998) have trained several classifiers on different balanced data subsets, each constructed to include all positive training samples and a set of negative samples of comparable size. The predictions are combined through stacking.

For the task of subjectivity analysis, especially in the multilingual paradigm where the data set is highly unbalanced, using one of the techniques proposed above will yield benefit. To the best of our knowledge, co-training technique has not been applied before for the subjectivity detection task, in particular, for the Urdu language.

1.2 Contribution

Our first contribution is inspired by the work of Luo *et al.,* (2008). We propose a similar co-training technique that helps to create a likely negative set (objective sentences) and a filtered positive set (subjective sentences) simultaneously from the unlabeled set. We use two learning models trained using the linear SVM algorithm iteratively. In every iteration of co-training, the likely positive samples are filtered. The iterative process terminates when no more positive samples are found. The final negative set is the likely negative set, considered as the universal representative set for the non-subjective category. The likely positive sample set is appended to the already existing positive set (annotated set). The SVM models are trained using part of speech, unigrams and emotion bearing words, as features.

The second contribution of this work includes training a state-of-the-art Vector Space Model

(VSM) for Urdu newswire data using the data sets generated by the co-training method. Experiments that use the SVM classifier are also performed. The results show that the performance of the proposed VSM based approach helps to achieve state-of-the-art sentence level subjectivity classifier. The F-Measure of the VSM subjectivity classifier is 82.72% with 78.7% F-measure for the subjective class and 86.7% F-Measure for the objective class.

2 Data Set

The data set used to generate a subjectivity classifier for Urdu newswire articles is obtained from BBC Urdu[2]. The annotating efforts are directed towards achieving the final goal- *emotion detection* in Urdu newswire data and the annotation guidelines are based on the MPQA standards set for English.

The repository of articles provided by BBC is huge and needs to be filtered intelligently. Two levels of filters are applied. – *date* and *keyword search*. The *date* filter is applied to retrieve articles of three years, starting year 2003. The *keyword* based filter consists of a set of seed words that are commonly used to express emotions in Urdu -*ghussa (~anger), pyar (~love)* etc. Clearly, this list will not cover all possible linguistic expressions that express emotion and opinion. But it is definitely a representative of a wide range of phenomena that naturally occurs in text expressing emotions.

The data retrieved is parsed using an in-house HTML parser to produce clean data. To date, we have 500 articles, consisting of 700 sentences annotated for emotions. There are nearly 6000 sentences that do not contain any emotions making it highly unbalanced. This data set is divided into testing and training sets with 30% and 70% of the data respectively. Co-training is performed only on the 70% training set that consists of 470 subjective sentences and about 4000 objective sentences. The purpose of co-training here is to remove samples that are close to subjective from the objective set and create a likely negative set. The samples removed are the likely positive set. This set of 4000 objective sentences can be considered as the un-annotated set.

[2] http://www.bbc.co.uk/urdu/

3 Co-Training

Identifying sentences that express emotions in Urdu newswire data is not trivial. Subjective sentences do not always contain individual expressions that indicate subjectivity. Analysis is highly dependent on the contextual information. Wiebe *et al.,* (2001) reported that nearly 44% of sentences in the MPQA corpus (English newswire data) are subjective. In newswire data, though most facts are reported objectively, there are cases when the tone of the sentence is very intense indicating the existence of emotion. Consider Example 1.

Example 1:
Political news headline

بھارت کا پاکستان کے ساتھ **جامع مذاکرات** سے انکار ،**بھارتی لیکچر سننے کے خواہاں نہیں**

[bhart ka pakstan kE sath jame mZakrat sE ankar, bharty lykcr snnE kE KwahaN nhyN]

*[India refuses to have a **dialog** with Pakistan, Indians **are not willing to listen to the lecture**]*

Common Urdu

انڈیا نے پاکستان سے **بات چیت** کارنے سے انکار کر دیا ہے

*[India refuses to **talk** to Pakistan]*

Clearly, the news headline is extremely intense and strongly expresses the opinion of India on Pakistan. However, the statement in common Urdu is not as affective.

Example 2:

انصاری نے کہا، میری رائے میں عامر سہیل ایک بد دماغ اور ضدی شخص ہیں

[anSary nE kha "myry ray^E myN eamr shyl ayk bd dmaG awr Zdy XKS hyN"]

[Ansari said, "according to me Aamir Sohail is one crazy and stubborn man"]

Statements in quotes that express emotions are subjective as shown in example 2.

Consider example 3. Here, identifying the words that indicate subjectivity is not straight forward. The phrase, *"found it very difficult to hide his smile"* is indicative of the emotion experienced by *"Habib Miya"*.

Example 3:

رقم کی اس وصولی پر یہ حبیب میاں کے لئے **بہت مشکل** تھا کہ وہ اپنی مسکراہٹ چھپا سکیں

[rqm ky as wSwly pr yh Hbyb myaN kE ly^E bht mXkl t\|ha kh wh apny mskrahT c\|hpa skyN]

[At this event of money collection, Habib Miyan found it very difficult to hide his smile.]

There are also several false positives that make subjective detection hard task. Example 4 is an objective sentence despite the usage of word *"pyar" ~ love*, an emotion bearing word.

Example 4:

انضمام کا نیا پیار کا نام انزی پڑا ہے

[n|Zmam ka nya pyar ka nam anzy pRa hE]

[The new nickname for Inzaman is Inzi]

Expressive elements in Urdu sentences were marked with an inter-annotator agreement of 0.8 kappa score. Though high, there still exists a bias that can influence classification especially when the number of sentences in the positive set is relatively less. In order to obtain a reliable positive and negative set for training a learning algorithm, we adopt a semi-supervised learning technique of co-training. *Co-training* (Blum and Mitchell, 1998) is similar to self-training in that it increases the amount of labeled data by automatically annotating unlabeled data. The intuition here is that if the conditional independence assumption holds, then on an average each selected document will be as informative as a random document, and the learning will progress. Co-training differs from self-training as it uses multiple learners to do the annotation. Each learner offers its own perspective that when combined gives more information. This technique is especially effective when the feature space of a particular type of problem can be divided into distinct groups and each group contains sufficient information to perform the annotation. In other words, co-training algorithm involves training two different learning algorithms on two different feature spaces. The learning of one becomes conditionally independent of the other and the prediction made by each classifier is used on the unlabeled data set to augment the training data of the other.

A traditional co-training classifier is trained and later applied on the same unlabeled data set. Theoretically such classifiers are not likely to assign confident labels. In this work, the proposed co-training method differs from the traditional co-training method in that the two classifiers are based not on two different feature spaces but on two different training data sets with the same feature space.

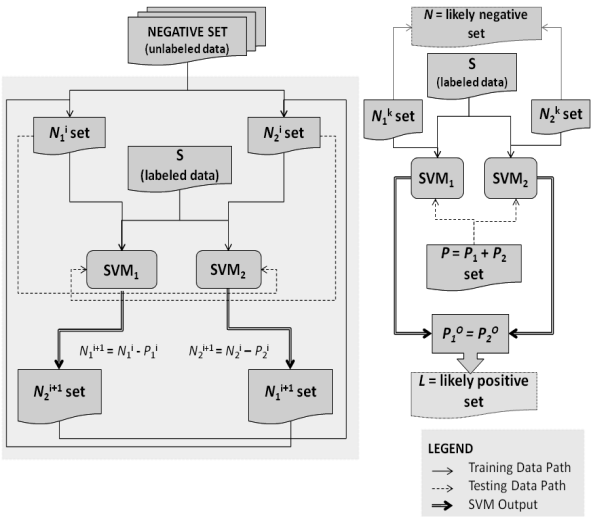


Figure 1: Co-Training model

Figure 1 explains the overall working of the model. The negative set (which can also be the unlabeled set) is split into two equal parts N_1 and N_2. S represents the positive annotated set. Two linear SVM classifiers are trained iteratively to purify the negative data set. SVM_1 is trained using $S+N_1^i$ and SVM_2 is trained using $S+N_2^i$ data sets. In every iteration *i*, N_1^i data set is evaluated using SVM_2 model and N_2^i data set is evaluated using SVM_1 model. The samples that are classified as positive in a given iteration *i* are binned into sets P_1^i and P_2^i respectively. These samples are removed from N_1^i and N_2^i data sets to create new N_1^{i+1} and N_2^{i+1} sets that are used for training in the next iteration *i+1*. The iterations continue until no positive samples are marked by both SVM_1 and SVM_2 models. The final set of likely negatives is $\mathbf{N} = N_1^k + N_2^k$ sets, where N_1^k and N_2^k are sets created in the last *k* iteration of the algorithm. In order to obtain the likely positive set, the final $P_1 = \{P_1^1 + P_1^2 + \ldots. + P_1^k\}$ and $P_2 = \{P_2^1 + P_2^2 + \ldots. + P_2^k\}$ sets are combined and tested using the SVMs modeled in the last *k* iteration of the co-training algorithm. Similar to the traditional co-training method the samples that are marked positive by both classifiers ($P_1^o = P_2^o$) are considered to be the likely positive set **L**.

Several features are used to train the SVM learning models used for co-training. The best performance is obtained when word unigrams, parts of speech and likely emotion words are used as features.

This technique of co-training provides us with a relatively huge set of likely positive samples

(close to 400 sentences). Sentences in this set were examined by the annotators and nearly 60% of the sentences were subjective or near subjective in nature (Example 5 and 6).

Labels	R %	P %	IF %	AF %
Unigram				**52.63**
1	**18.64**	74.57	**29.83**	
-1	95.4	62.35	75.44	
Unigram+Bigram				50.25
1	**14.40**	85	**24.63**	
-1	98.19	61.82	75.87	

Table 1: Performance of the model using un-balanced data set[3]

Labels	R %	P %	IF %	AF %
Annotated positive + likely positive + likely negative				**62.95**
1	**39**	70	**50.09**	
-1	87.28	67.34	79.9	
Annotated positive + likely negative				55.42
1	30	61.2	**40.26**	
-1	86.1	64.23	73.57	

Table 2 – Performance of the model after co-training method

Table 1 shows the performance of the SVM model using the unbalanced data set for training. Table 2 shows the performance of the same model using data generated after co-training.

Example 5:

پوتن نے کہا کہ لوگ دوسروں کی آنکھ میں تنکا دیکھ لیتے ہیں لیکن اپنی آنکھ میں پڑا شہتیر انہیں نظر نہیں آتا ۔

[pwtn nE kha kh lwg dwsrwN ky Ank|h myN tnka dyk|h lytE hyN lykn apny Ank|h myN pRa Xhtyr anhyN n|zr nhyN Ata .]

[Potan said people who see dust in others eyes never realize that it is their eyes that are filled with dirt.]

The above example is a metaphor indicating extreme anger.

Example 6:

عطاء الرحمن شیخ کا کہنا ہے کہ بارہ اگست کو انہیں ان کے بیٹوں کے سامنے مکمل طور پر برہنہ کر کے پریڈ کرائی گئی

[e|ta& alrHmn XyK ka khna hE kh barh agst kw anhyN an kE byTwN kE samnE mkml |twr pr brhnh kr kE pryD kray^y gy^y]

[etlaalrahman said that on 12th Aug they made him parade naked in front of his children.]

Example 6 indicates extreme sad emotion. Such examples were found in the likely positive set.

4 Features

Features that are commonly used to train a subjectivity classifier for English are word unigrams, emotion keywords, part of speech information and noun patterns (Pang *et al.,* 2002). Due to difference in syntactic structure, vocabulary and style, features that work for English may not work for Urdu. Also, Urdu is handicapped by the lack of resources required to perform basic NLP analysis. However, it is worth exploring the English feature set as subjectivity is more a semantic phenomenon. Efforts to generate likely emotion word lexicons and subjectivity patterns for the Urdu language are underway. The sections that follow summarize the experimented features.

4.1 Word Unigrams

Unigram word features are very informative. Three different approaches are tried for selecting the unigrams. The first method involves selecting only those words that occur more than twice in the dataset. This eliminates proper nouns (low frequency named entities do not generally contribute towards subjectivity detection) and spelling errors (Pang *et al.,* 2002). In the second method, only words that are adjectives and verbs along with the surrounding case markers are accounted for as features. This has the advantage of drastically reducing the feature set. The third method involves including the nouns as well to the feature set. A simple list of stop words (common Urdu words – pronouns such as *'us', 'is', 'aap', 'un'*, salutations like *'shabba khair', 'aadab'* and honorifics along with punctuations and special symbols) are eliminated. The features are represented as Boolean features for the SVM model. The value is 1 if the feature word appears in the sentence to be classified and 0 otherwise. The best performance is obtained for the first method that considers all words with frequency greater than 2. This conforms to what is shown by Pang *et al.,* (2002) for classification of English movie reviews.

4.2 Part of Speech (POS) Information

The work done by Mukund and Srihari (2009) provides suitable POS and NE tagger for Urdu.

[3] Convention used across tables - Label 1: subjective sentences Label -1: objective sentences R: Recall P: Precision IF: Individual F-Measure AF: Average F-Measure.

This POS tagger is used to generate parts of speech tags on the acquired data set (§3). The POS tags associated with adjectives, verbs, common nouns and auxiliary words are considered and used as Boolean features for the SVM model. The proper noun words are normalized to one common word *"nnp"* and are assigned the common noun tag. For the English language, when building a subjectivity classifier for review classification, the use of POS information did not benefit the system (Kennedy and Inkpen, 2006). However, for Urdu, the performance of the co-training model with POS information showed 1.2% improvement (table 3).

4.3 Likely Emotion Lexicon

In order to facilitate simple keyword based detection of subjectivity, access to a lexicon consisting of likely emotion words is needed. Unfortunately, no such lexicon is available off the shelf for Urdu. In this work, an Urdu specific emotion list is generated that contains translations from the English emotion list released by SemEval (2007) '*WordNet affect Emotion List*'. Words for each emotion category - sadness (sad), fear, joy (happy), surprise, anger and disgust are obtained for Urdu by using an Urdu-English dictionary. The list is pruned manually and corrected to remove errors. Simple keyword lookup on the Urdu annotated corpus has an emotion detection rate of 29.27%. This shows that although the contribution of the emotion lexicon for subjectivity classification is not significant, it contains information which when used along with other features aid subjectivity detection.

4.4 Patterns

Extracting syntactic patterns contribute towards the affective orientation of a sentence (Riloff *et al.,* 2003). The Apriori algorithm (Agarwal and Srikant, 1994) for learning association rules is used here to mine the syntactic word patterns commonly used in the positive and negative data set. The length of the candidate item set k = 4. Starting from a small set of seed words (likely emotion words) and the associated POS tags, POS sequential patterns like "adverb verb verbtransitive sentencemarker", "noun noun casemarker verbtransitive", etc., that are most commonly found in subjectivity set are extracted. 23 patterns that strongly indicate subjectivity were found by this method and included as features to train the SVM learning algorithm.

4.5 Confidence Words

The confidence word list positively aids the VSM classifier (§5). The words in the likely emotion list are not the only ones that contribute towards the emotion orientation of a sentence and also, not all of these words contribute effectively. There are several stop words (eliminated while accounting for unigrams) (esp. case markers) that contribute significantly for categorization. In order to identify all the keywords that actually contribute to subjectivity categorization, a technique proposed by Soucy and Mineau (2004) is used.

The confidence weight of a given word w, based on the number of documents it is associated with under each category, is measured using the Wilson Proportion Estimate (Wilson, 1927). In order to compute the confidence of w for a specific category, the number of positive and negative documents associated with w has to be determined. A document is positive if it belongs to that category and negative otherwise. Thus, two kinds of word confidence metrics are computed, $C_{POS:w}$ and $C_{NEG:w}$ as given below.

$$C_{POS:w} = \frac{\left(\hat{p}_{POS:w} + \frac{z^2_{\alpha/2}}{2n} \pm z_{\alpha/2} \sqrt{[\hat{p}_{POS:w}(1-\hat{p}_{POS:w}) + z^2_{\alpha/2}/4n]/n} \right)}{(1 + z^2_{\alpha/2}/n)} \quad \text{..... (Eq. 1)}$$

$$C_{NEG:w} = \frac{\left(\hat{p}_{NEG:w} + \frac{z^2_{\alpha/2}}{2n} \pm z_{\alpha/2} \sqrt{[\hat{p}_{NEG:w}(1-\hat{p}_{NEG:w}) + z^2_{\alpha/2}/4n]/n} \right)}{(1 + z^2_{\alpha/2}/n)} \quad \text{..... (Eq. 2)}$$

where n is the total number of positive and negative documents, $\hat{p}_{POS:w}$ is the ratio of the number of positive documents which contain w to the total number of documents, and $\hat{p}_{NEG:w}$ is the ratio of the number of negative documents which contain w to the total number of documents. The normal distribution is used when $n > 30$.

Note that equations 1 and 2 give a range of values for $C_{POS:w}$ and $C_{NEG:w}$. If the lower bound of $C_{POS:w}$ is greater than the upper bound of $C_{NEG:w}$, we say that w is likely to be a word in that category. Now, we compute the strength of a word S_w in a particular category as

$$S_w = \begin{cases} \log(2 \cdot mPRF) & ; \text{if } lb(C_{POS:w}) > ub(C_{NEG:w}) \\ 0 & ; \text{otherwise} \end{cases}$$

..... **(Eq. 3)**

where *mPRF* is given by

$$mPRF = \frac{lb(C_{POS:w})}{lb(C_{POS:w}) + ub(C_{NEG:w})}$$

..... **(Eq. 4)**

and lb(…) and ub(…) are the lower and upper bounds of their arguments, respectively.

Equations 1 through 4 generated a very good set of keywords that are used as category word features in the SVM learning model. For VSM, the strength value is used as a boost factor along with the *tf-idf* weight when calculating the similarity score (table 3).

5 Final Subjectivity Classifier

Wiebe *et al.,* (2005) and Pang *et al.,* (2002) have shown that an SVM based approach works well for subjectivity classification. Riloff *et al.,* (2003) have conducted experiments that use Bag-Of-Words (BoW) as features to generate a Naïve Bayes subjectivity classifier for the MPQA corpus in English. This method has an accuracy of 73.3%. Su and Markert (2008) use BoW features termed as lexical features on the IMDB corpus to generate an accuracy of 60.5%. Das and Bandyopadhyay (2009) use a CRF based approach to generate a subjectivity classifier for Bengali data with a precision of 72.16% for news and 74.6% for blogs domain. The same approach has a precision of 76.08% and 79.9% on the two domains respectively. Impressive results for emotion detection are obtained by Danisman and Alpkocak, (2007) who use a VSM based approach. They show that their approach works much better than a traditional SVM based approach commonly used for emotion detection.

In this work, we conduct subjectivity classification experiments using two different learning algorithms – linear SVM and VSM. The best performance is obtained using the VSM model as shown in table 4. All experiments are conducted on the data set obtained after applying the co-training technique.

5.1 VSM algorithm

The final subjectivity classifier is based on the VSM approach. Inspired by the work done in "*Feeler*" (Danisman and Alpkocak, 2007), a similar technique is used to train the final subjectivity classifier for Urdu. The algorithm is explained in table 3. The similarity metric is modified to include the confidence score for each word (pt.5). In VSM, documents and queries are represented as vectors, and the cosine angle between them indicates the similarity.

1.	$d_i = <w_{1i}, w_{2i}, ... w_{ni}>$ where w_{ki} is the weight of the k^{th} term in document i , d_i is the document vector. w_{ki} is computed using *tf-idf* weighting scheme.
2.	$M_j=\{d_1,d_2,\ldots,d_c\}$ where M_j is each class (subjective and objective)
3.	Model vector for an arbitrary class E_j is created by taking the mean of d_j vectors $$E_j = \frac{1}{M_j}\sum_{d_i \in M_j}^{|M_j|} d_i$$ where $\|M_j\|$ represents number of documents in M_j.
4.	The whole system is represented with a set of model vectors, $D=\{E_1,E_2,...,E_s\}$ where s represents the number of distinct classes to be recognized.
5.	The normalized similarity between a given query text Q, and a class, E_j, is defined as follows: $$sim(Q,E_j) = \sum_{k=1}^{n}(w_{kq} + conf) * E_{kj}$$ *conf* is the confidence factor applied for lexical terms found in the word list.
6.	classification result is, $VSM(Q) = \arg\max(sim(Q,E_j))$

Table 3: VSM Algorithm for subjectivity Classification

Labels	R %	P %	IF %	AF %
Before Co-Training (all data)				**62.95**
1	65.85	70.85	**67.4**	
-1	85.58	83.33	84.44	
After Co-Training (pruned data)				**86.73**
1	**72.88**	85.57	**78.72**	
-1	91.29	82.60	86.73	

Table 4: VSM approach, using all training data and using pruned training data (L+N+true)

The confidence metric (strength) for each term is calculated using the Wilson proportion estimate (§4.4) and added to the term score as the boost factor. Q is the test set. Model vectors are obtained using the data set that consists of true set (annotated positive samples), likely positive set L and likely negative set N. Sets L and N are obtained from the co-training method. The results are shown in table 4.

The power of SVM cannot be ignored. Pang *et al.,* (2002) use SVM to generate a subjectivity (polarity) classifier for English. Our second set of experiments is conducted to measure the performance of a linear SVM classifier for subjectivity analysis on the Urdu newswire data. The data set used for training is the pruned data set

obtained after applying the co-training technique. The features used and the performance of the model with each feature is documented in table 6.

Labels	R %	P %	IF %	AF %
Unigrams + POS				64.2
1	40.67	71.1	**51.75**	
-1	88.29	67.74	76.67	
Unigrams + POS + Patterns				65.68
1	43.22	72.34	**54.11**	
-1	88.29	68.69	77.26	
Unigrams + POS + Patterns + emotion words				**67.31**
1	48.31	70.81	**57.43**	
-1	85.88	70.09	77.19	

Table 6: SVM classifier on Urdu newswire data

In order to provide a better understanding of the power of the VSM technique, we applied this model on the IMDB data set. The training data consists of 4000 positive (subjective) and 4000 negative (objective) samples. Since the data set is already balanced, we skip the co-training method. Our aim here is to test the working of VSM classifier. The test set consists of 1000 positive and 1000 negative samples. The classification result on this data set is shown in table 5. The results are comparable to the state-of-the-art performance of English subjectivity classifier that uses SVM (Wiebe *et al.,* 2005).

Labels	R %	P %	IF %	AF %
Balanced training				**78.01**
1	64	90.57	**75**	
-1	93.18	71.68	81.03	

Table 5: VSM approach on IMDB data set

6 Analysis of results

In this work, experiments were conducted using two different classification approaches; 1. VSM based 2. SVM based. Results in table 4 indicate that the VSM technique when combined with the modified boost factor (confidence measure) can be a very powerful technique for sentence level classification tasks. When model vectors were constructed using the entire training set (highly unbalanced), the performance was at 62% F-Measure with the subjectivity detection rate of 70.85%. Post co-training, using the modified model vectors obtained from the pruned data set generated better scores. The increase in the recall of negative class and the increase in the overall F-Measure can be attributed to (i) increase in the positive samples (~likely positive set), and (ii) cleaner negative set (no near positive samples).

The results in table 6 for the SVM classifier also indicate the benefits of co-training. The subjectivity classification performance show positive improvement. Although the performance of the SVM model is not as good as the VSM model, addition of each feature shows an improvement in the subjectivity recognition rate. This performance indicates that the feature sets explored definitely contain positive information necessary for accurate detection.

The poor performance of SVM (over VSM) can be attributed to 1. lack of balanced data for training a traditional SVM model and, 2. small number of positive samples. In VSM the problem of unbalanced data set in a way is overcome by using the confidence score at the time of calculating similarity. If these factors are compensated, the performance of the SVM model will significantly improve.

7 Conclusion

This research provides interesting insights in modeling a subjectivity classifier for Urdu newswire data. We show that despite Urdu being a resource poor language, techniques like co-training and statistical techniques based on *tf-idf* and word unigrams coupled with confidence measures help model the state-of-the-art subjectivity classifier. We demonstrate the power of the co-training technique in generating likely negative and positive sets. The number of near subjective samples in the likely positive set suggests that this method can be used as an adaptive learning technique to enable the annotators produce more samples. For a task like emotion detection, that requires fine grained analysis, sentences need to be analyzed at the semantic level and this goes beyond simple keyword based approach. Our efforts are now concentrated in this direction.

References

Agrawal R, Srikant R. 1994. Fast Algorithms for Mining Association Rules. *In Proc. Of the Intl. Conf on Very Large databases.* Santiago, Chile. Sept. Pp. 478-499.

Banea, C., Mihalcea, R., Wiebe, J., and Hassan, S. 2008. Multilingual subjectivity analysis using machine translation. In *Proceedings of EMNLP-2008.*

Banfield, A. 1982. *Unspeakable Sentences*. Routledge and Kegan Paul, Boston.

Blum, A. and Mitchell, T. 1998. Combining labeled and unlabeled data with co-training. *Proceedings of the eleventh annual conference on Computational learning theory, ACM. p. 100.*

Brank, J., Grobelnik, M., Milic-Frayling, N., and Mladenic, D. 2003. Training text classifiers with SVM on very few positive examples. *Technical Report* MSR-TR-2003-34, Microsoft Corp.

Chan, Philip K. and Stolfo J. Salvatore. 1998. Toward Scalable Learning with Non-Uniform Class and Cost Distributions: A Case Study in Credit Card Fraud Detection. *Proc. 4th Int. Conf. on Knowledge Discovery and Data Mining* (KDD-98), August 27–31, 1998, New York City, New York, USA, pp. 164–168. AAAI Press.

Danisman, T., and Alpkocak, A. 2008. Feeler: Emotion Classification of Text Using Vector Space Model. *AISB 2008 Convention Communication, Interaction and Social Intelligence, p. 53.*

Das, A., and Bandyopadhyay, S. 2009. Subjectivity Detection in English and Bengali: A CRF-based Approach. *Seventh International Conference on Natural Language Processing* (ICON 2009), December. Hyderabad, India.

Japkowicz Nathalie. 2000. Learning from Imbalanced Data Sets: A Comparison of Various Strategies. *In Nathalie Japkowicz (ed.), Learning from Imbalanced Data Sets*: Papers from the AAAI Workshop (Austin, Texas, Monday, July 31, 2000), AAAI Press, Technical Report WS-00-05, pp. 10–15.

Kennedy, A, & Inkpen, D. 2005. *Sentiment classification of movie and product reviews using contextual valence shifters*. In Workshop on the analysis of informal and formal information exchange during negotiations (FINEXIN 2005)

Ku, L. W., Liang, Y. T., and Chen, H. H. 2006. Opinion extraction, summarization and tracking in news and blog corpora. In *Proceedings of AAAI-2006*.

Kubat, Miroslav and Matwin Stan. 1997. Addressing the curse of imbalanced training sets: one-sided selection. *Proc. 14th ICML*, Nashville, Tennessee, USA, July 8–12, 1997, pp. 179–186.

Liu, B., Hu, M., and Cheng, J. 2005. Opinion observer: Analyzing and comparing opinions on the web. *In Proceedings of WWW-2005*.

Luo, N., Yuan, F., and Zuo, W. 2008. Using CoTraining and Semantic Feature Extraction for Positive and Unlabeled Text Classification. *International Seminar on Future Information Technology and Management Engineering*.

Mihalcea, R., Banea, C., and Wiebe, J. 2007. Learning multilingual subjective language via cross-lingual projections. In *Proceedings of ACL-2007*.

Mukund, S., and Srihari, R.K., 2009. NE Tagging for Urdu based on Bootstrap POS Learning. *Third International Workshop on Cross Lingual Information Access*: Addressing the Information Need of Multilingual Societies (CLIAWS3), NAACL - 2009, Boulder, CO.

Pang, B., Lee, L., and Vaithyanathan, S. 2002. Thumbs up? Sentiment classification using machine learning techniques. In *Proceedings of the Conference on EMNLP*, pages 79–86.

Riloff, E., Wiebe, J., and Wilson, T. 2003. Learning subjective nouns using extraction pattern bootstrapping. *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003* - Volume 4, Edmonton, Canada: Association for Computational Linguistics, pp. 25-32.

Soucy, P., and Mineau, G. W. 2005. Beyond tfidf weighting for text categorization in the vector space model. *International Joint Conference on Artificial Intelligence*, Citeseer, p. 1130.

Su, F., and Markert. K. 2008. From words to senses: a case study of subjectivity recognition. *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1, ACL*, pp. 825-832.

Titov, I., and McDonald, R. 2008. A joint model of text and aspect ratings for sentiment summarization. *In Proceedings of ACL-08:HLT*.

Wan, X. 2008. Using bilingual knowledge and ensemble techniques for unsupervised Chinese sentiment analysis. In *Proceedings of EMNLP-2008*.

Wan, X. 2009. Co-Training for Cross-Lingual Sentiment Classification. *In Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, Association for Computational Linguistics, pp. 235-243.

Wiebe, J. 1994. *Tracking point of view in narrative*. Computational Linguistics, 20(2):233-287.

Weibe, J., Bruce, R., and O'Hara, T. 1999. Development and use of a gold standard data set for subjectivity classifications. *In Proc. 37th Annual Meeting of the Assoc. for Computational Linguistics (ACL-99).*

Wiebe, J., and Riloff, E. 2005. Creating Subjective and Objective Sentence Classifiers from Unannotated Texts. *Proceedings of the 6th International Conference on Intelligent Text Processing and Computational Linguistics*.

Wiebe, J., Wilson, T., and Cardie, C. 2005. Annotating expressions of opinions and emotions in language. *Language Resources and Evaluation*, volume 39, issue 2-3, pp. 165-210.

Wilson, B. Edward. 1927. *Probable Inference, the Law of Succession, and Statistical Inference*. Journal of the American Statistical Association, Vol. 22, No. 158 (Jun., 1927), pp. 209-212.

Support or Oppose? Classifying Positions in Online Debates from Reply Activities and Opinion Expressions

Akiko Murakami[1,2] **Rudy Raymond**[1]
[1] IBM Research - Tokyo
[2] Graduate School of Interdisciplinary Information Studies, The University of Tokyo
{akikom,raymond}@jp.ibm.com

Abstract

We propose a method for the task of identifying the *general* positions of users in online debates, i.e., support or oppose the main topic of an online debate, by exploiting *local* information in their remarks within the debate. An online debate is a forum where each user post an opinion on a particular topic while other users state their positions by posting their remarks within the debate. The supporting or opposing remarks are made by directly replying to the opinion, or indirectly to other remarks (to express local agreement or disagreement), which makes the task of identifying users' general positions difficult. A prior study has shown that a link-based method, which completely ignores the content of the remarks, can achieve higher accuracy for the identification task than methods based solely on the contents of the remarks. In this paper, we show that utilizing the textual content of the remarks into the link-based method can yield higher accuracy in the identification task.

1 Introduction

Social computing tools, such as a SNS (Social Network Service) or an online discussion board have become very powerful communication tools for discussing topics with people around the world. Many companies use these kinds of social computing tools to understand their customers' requirements and their marketing activities. Social computing tools are very useful not only for aggregating customers' opinions outside the companies, but also for aggregating their employees' ideas. For example, IBM has held Jam[1] sessions, which are short-term online discussions to aggregate ideas from employees and customers. The results of Jam sessions help management decisions, for instance the technology areas to invest.

Not just enterprises, but some nations are trying to aggregate their citizens' ideas in the Internet and provide systems for discussions at the people-to-people levels as part of the movement for open government. The United States government has the *Idea Factory*[2] website for collecting ideas to enhance activities of Department of Homeland Security (DHS) and the *Open For Questions*[3] to collect requests for the US government.

The motivation for creating these kinds of online discussions is not limited to collecting ideas but also to help understand the trends of opinions about the ideas or topics. This means that getting a quick overview of opinions about ideas is a key point for the success of online discussions.

In this paper we propose a method to show quick overview of participants' positions, "Support" or "Oppose" for the main idea or topic in online debates. It is difficult to identify each person's position for a topic directly, since most of opinionative expressions are made not for main topic but for adjacent remarks. This causes a difficulty in building answer sets for classifier. The following example shows opinion expressions for a main topic focused on an adjacent remark in a

[1] https://www.collaborationjam.com
[2] http://www.whitehouse.gov/open/innovations/IdeaFactory
[3] http://www.whitehouse.gov/OpenForQuestions/

Coling 2008: Poster Volume, pages 869–875,
Beijing, August 2010

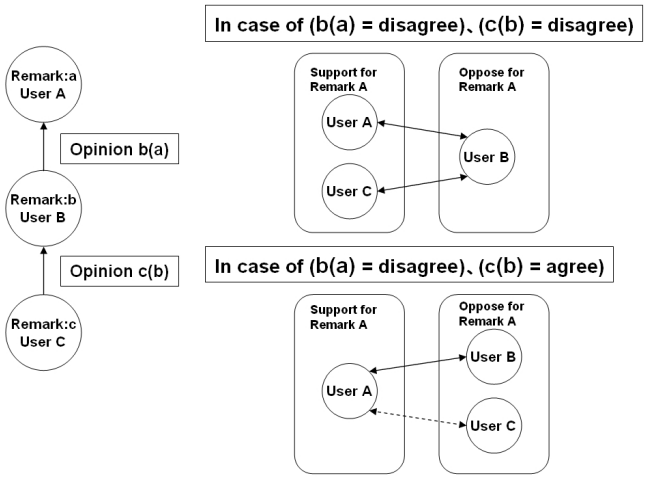


Figure 1: Identifying users' positions from their opinions about previous remarks

debate. In this example, The main topic is "Travel and F2F (face-to-face) meeting is fundamental to business".

Remark A Travel isn't necessary because besides the high cost of travels around the world, today we have a lot of communication tools, for instance web conference, video chat that can easily contribute to join leaders around the world in a cheaper way.

Remark B I disagree. Without travel and F2F meetings global integration does not work as well or as quickly. It doesn't mean that everybody has to travel all the time, but at least some meetings are key to success.

The author of Remark A mentions that travel is not necessary to business. This opinion opposed to the main topic, so that the position for the main topic is "Oppose". In contrast, the opinion expression in Remark B is not an opinion about the main topic, but relates to the previous opinion in Remark A. This opinion expression indicates that the author of Remark B disagrees with the opinion of Remark A, and indirectly implies agreement with the main topic. Thus, although it is hard to infer the global position of Remark B from only the surface expressions, it is straightforward to infer that an opinion in Remark B about Remark A is negative (i.e., Remark B expresses disagreement with Remark A).

In this paper, positions with regards to the main topic (global positions) are classified into two classes: support and oppose, while opinions about the previous remarks (local positions) are classified into three: *agree*, *disagree*, and *neutral*. For example, let us consider the case in Fig. 1, where Remark "a" is the main topic, and Remark "b" is the reply to Remark "a" and Remark "c" is the reply to Remark "b". Here, let $b(a)$ be the local position, that is, opinion (agree/disagree/neutral) in Remark "b" on the topic in Remark "a". For example, if $b(a)$ and $c(b)$ are *disagree*, one can determine that the authors of the corresponding remarks are in the opposition. That is, the author of Remark "c" agrees with Author A (the author of Remark "a"), that is, the main topic, while Author B is against the others. On the contrary, if $b(a)$ is *disagree* and $c(b)$ is *agree*, then Author C agrees with Author B and therefore it implies that Author C is against Author A. In this case, only Author A supports the main topic while Author B and Author C oppose to the main topic.

To infer supporting or opposing positions with regards to the main topic, two steps are used. First, the degree of disagreement between any two users is computed from the link structure and the text of

each pair of their adjacent replies. This is used as the link weight between nodes (which correspond to users in a debate) in the network. Second, the bipartition of the users in the weighted network is computed by finding a bipartition that induces the *maximum cut* of the network, a partition of nodes into two disjoint sets that maximizes the sum of the weights of the links connecting nodes in different sets. Since the weight of the links is higher (more positive) when the degree of disagreement is higher, the bipartition is expected to express two groups of opposing positions.

In order to evaluate the performance of our method, we conducted some experiments to identify the supporting and opposing positions of participants in online debates. The experimental results indicate that our method leads to higher precision than the baseline method, which is described in (Agrawal et al., 2003).

The rest of this paper is organized as follows. First we describe related work in Section 2, and in Section 3 we propose our method for identifying participants' positions from their reply activities and text contents. In Section 4 we explain the data sets used for the evaluations and show the experimental results of an opinion classifier for adjacent remarks and a support/oppose classifier for the participants in online debates. We conclude this paper and describe future work in Section 5.

2 Related Work

There are some research papers published on analysis of online discussions. Some researches reported on how to analyze and navigate IBM Jam sessions. Millen et al. pointed out the importance of supporting the participants in discussions and demonstrated the effectiveness of their methods in one of these jams (Millen and Fontaine, 2003). Dave et al. described ways for jam participants to navigate using visualization techniques (Dave et al., 2004). One of the authors previously also proposed a method to mine discussion records using XML annotations (Murakami et al., 2001) and a method to find important remarks in a discussion thread based on the reply-to structure and participants' opinions (Murakami et al., 2007).

Classifying agree/disagree opinions in conversational debates using Bayesian networks was presented in (Galley et al., 2004). Agrawal et al. described an observation that reply activities show disagreement with previous authors, and showed a method to classify the supporting/opposing position of users based on this observation in (Agrawal et al., 2003). Thomas et al. (Thomas et al., 2006) introduced some constraints that a single speaker retains the same position for the classification of participants' positions from floor-debate transcripts.

3 Proposed Method

3.1 Calculating the *Reaction Coefficient* between participants

We call the degree of divergence in the opinions between participants a *reaction coefficient*. This reaction coefficient is defined as a function of the participants i, j, represented as $r(i, j)$. To calculate reaction coefficients, we extracted pairs of a remark and its reply remark, and assigned "local position flags" to the pairs. There are three local position flags, "agree", "disagree", and "neutral". The reaction coefficient $r(i, j)$ between participants i and j is defined as:

$$r(i,j) = \alpha N_{\mathrm{disagree}}(i,j) + \beta N_{\mathrm{neutral}}(i,j) + \gamma N_{\mathrm{agree}}(i,j), \quad (1)$$

where $N_{\mathrm{opinion}}(i, j)$ is the number of remark pairs with $\mathrm{opinion}$ as the corresponding local position flag between participants i and j.

Typically we assign a positive value to α, a slightly positive value to β, and a negative value to γ. This means that $r(i, j)$ is positive when there are only neutral remarks between user i and j. This is based on the hypothesis in (Agrawal et al., 2003) that replies usually indicate disagreement with previous remarks. There is no directionality in reaction coefficients so that $r(i, j) = r(j, i)$.

3.2 Classification of Participants' Positions based on the Max Cut Problem

Let the graph corresponding to the activity network of the participants in an online debate be $G(V, E)$, where V is the set of nodes that corresponds to participants and E is the set of edges each of which links participants that exchanged remarks. For any $i, j \in V$, let $r(i, j)$ be the weight of the link between i and j. A partition of the

Table 1: Ideas and Number of Comments and Participants for the Ideas

Idea ID	Title	# of Comments	# of Participant	# of Remarks per Participant
1	Making "IT" Education as a Compulsory Subject in Schools	75	45	1.7
2	Making Personal-Computer Makers to Supplying Service Parts	130	21	6.2
3	Adoption of "Basic Income"	118	57	2.1
4	Votes in elections using Closed Networks	108	40	2.7
5	Computerized Books in Libraries	50	12	4.2

participants into supporting and opposing parties, S_{sup} and S_{opp} respectively, is computed by solving the *max cut* problem on $G(V, E)$ defined as follows.

[Max cut problem] Given $G(V, E)$ as above, find a bipartition of V into S_{sup} and $S_{\text{opp}} = V \setminus S_{\text{sup}}$ so that $\sum_{i \in S_{\text{sup}}, j \in S_{\text{opp}}} r(i, j)$ is maximized.

The max cut problem is known to be NP-hard, and thus in general is difficult to solve. However, good approximation algorithms based on Linear Programming and Semidefinite Programming have been developed recently, and combined with branch-and-bound techniques a good exact max-cut solver called *BiqMac* exists (Rendl et al., 2010). We used BiqMac for solving the max cut problem *exactly* on the activity network. Although a faster approximate max cut solver is used in (Agrawal et al., 2003), it is based on the limiting assumption that the size of S_{opp} is approximately the same as S_{opp}. This cannot be assumed for the networks in this paper.

4 Experiments

4.1 Corpus

The Ministry of Economy, Trade and Industry in Japan (METI) was accepting public opinions on e-government programs via the "e-METI Idea Box[4]" from February 23 to March 15 2010. Participants could show their positions for the ideas since the site accepted comments on the main idea and other comments, so this discussion can be regarded as a kind of debate. We used this data to evaluate our proposed method. The ideas and comments were written in Japanese and the data is available at the METI website.

For the 936 ideas that were posted to the Idea Box, we examined 17 ideas with more than 40 comments. Finally we selected five ideas for the evaluation. The numbers of remarks (a main idea and comments), participants, and remarks per participants are shown in Table 1.

We extracted the reply-to structure information in textual contents. The Idea Box system had a capability to adding a comment to a main topic or the other comment, and the system inserted an identifier in comment's text. Each identifier started with "#" and the IDs of the previous comments followed the identifier, such as "`#003`" (with `#001` referring to the main topic in the thread). An idea or comments may have several comments as replies, so this reply-to structure in a debate is a tree structure whose root node is the main topic. A typical reply-to tree structure is shown in Fig 2.

4.2 Agree/Disagree Classification

To calculate the reaction coefficients, we need to extract the reply-to pairs and classify these pairs into the agree/disagree/neutral classes. To classify these remark pairs we use opinionative and sentiment expressions. If a reply remark contains an expression of "I agree with you" then it should be classified into the agree class. Another example of expressions of the agree class would be "That's a good idea!".

To extract expressions of opinion, we created a simple pattern dictionaries that contains

[4]http://www.meti.go.jp/policy/it_policy/open-meti/

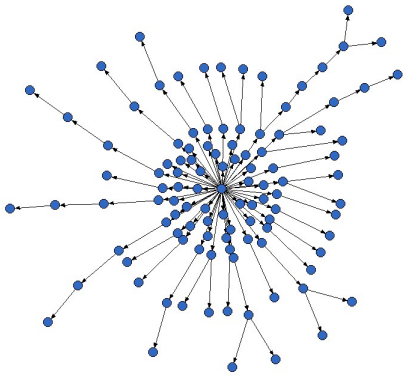


Figure 2: Reply-to Structure of a Debate

agree/disagree expressions. For instance, "I disagree with your idea" and "I don't agree with you" are in the disagree pattern dictionary. At the same time we use a sentiment analysis tool to extract sentiment expressions. The tool we used for sentiment expression extractions is the same as described in (Kanayama et al., 2004), which use machine translation techniques to identify sentiment expressions in text. The tool returns sentiment expressions with a sentiment label, favorable or unfavorable.

After identifying opinionative and sentiment expressions in the remarks, scores for the opinion classification are calculated. The score of each reply-to pair is the number of agreeing and favorable expressions minus the number of disagreeing and unfavorable expressions in the reply remark. When the score is positive, the opinion of the pair is identified as agree, and if the score is negative then the opinion of the pair identified as disagree. If the score equals zero, then the opinion is identified as neutral.

To evaluate this opinion classifier, we did an experiment with the METI data, which was manually assigned agree/disagree/neutral flags. The answers for these evaluation were created by us for three of the idea threads (Idea IDs #1,#2 and #3). Since most remarks do not have agree or disagree expressions, most reply-to pairs are classified into the neutral class. This means that calculating precision and recall for the neutral class are not important. For the evaluation of the classifier we calculated precisions and recalls only for agree and disagree classes. The results are shown in Table 2.

Table 2: Accuracy of opinion classification for reply-to pairs

Idea ID	Precision	Recall
1	0.63	0.25
2	0.62	0.14
3	0.44	0.38
Ave.	0.56	0.26

4.3 Support/Oppose Classification

Using the numbers of agree/disagree/neutral reply-to pairs, we can calculate the reaction coefficients for each pair of participants. After calculating the reaction coefficients for all of the participants' pairs, we can classify each participant into support or oppose sets using the max cut technique. In this subsection, we explain how to evaluate our proposed method and show experimental results.

4.3.1 Answer Sets for Global Position Classification

To evaluate our method we created answer sets for a global position classifier, consisting of participant sets with the position labels *Support* or *Oppose*. We identified the positions of the participants' remarks with contexts, but we assigned the "Unclear" label for some participants since their remarks did not contain enough information to classify their global positions. For showing the validity of the answer sets, two annotators annotated three ideas and calculated a κ value. The κ value is 0.69 so that this answer set is appropriate as an evaluation set. The use of the answer set annotated by a single annotator for the evaluation of support/oppose classification is justified since the agreement rate (the κ value) is enough for the evaluation.

4.3.2 Evaluation Index for Position Classification

For evaluation we defined the estimation index $accuracy$ since the number of participants in the Support position is not always the same as the

number of participants in the Oppose position. If the answers are grossly one-sided, the general accuracy does not work well, since the system can lead to a high score when it classifies all of the participants into the larger side. To minimize this potential bias, we defined an estimation index $accuracy$ using the average of the accuracies for the Support/Oppose sets. The estimation index $accuracy$ is defined as:

$$\text{accuracy} = \frac{1}{2}\left(\frac{|A_{\text{sup}} \cap S_{\text{sup}}|}{|A_{\text{sup}}|} + \frac{|A_{\text{opp}} \cap S_{\text{opp}}|}{|A_{\text{opp}}|}\right), \quad (2)$$

where A_{sup} and A_{opp} are the Support and Oppose participant sets in the answer set and S_{sup} and S_{opp} are the Support and Oppose participant sets generated by the system, respectively. For $accuracy$, we ignore "Unclear" users since the system is a two-class classifier.

4.3.3 Experimental Results

In the experiments we use the reaction coefficients $r(i, j)$ calculated based on the results of the agree/disagree/neutral Classifier, and classify participants into Support/Oppose position sets using BiqMac. Since we assumed that the main topic of the debate is the first remark of the debate thread, we assume that the set which includes the author of the first remark as the "Support" set and the other set as the "Oppose" set[5].

We conducted experiments for $(\alpha, \beta, \gamma) = (1, 0, 0), (1, 0.5, 0), (1, 0.5, -1)$ in Eq. (1) to examine the dependency of the accuracy on the coefficients $r(i, j)$. We also conducted an experiment for $(1, 1, 1)$, which is regarded as a baseline method described in (Agrawal et al., 2003), since all of the reply actions represent "disagree" opinions for the previous remarks with these parameter. The experimental results are shown in Table 3.

The ideas other than ID 1 show better accuracy than the baseline and their accuracies tend to increase in the order of $(1, 0, 0)$, $(1, 0.5, 0)$, $(1, 0.5, -1)$. This result shows that the effectiveness of distinguishing between "disagree" and "agree" replies. This distinction makes it possible to introduce the constraint in which the user pairs with "disagree" and "neutral" should be classified into opposing positions and user pairs with "agree" should be classified into same position in the Support/Oppose user sets.

[5]For this reason, the values of the accuracies can be lower than 0.5.

Table 3: Accuracy of Support/Oppose position classification

ID	Baseline	(1,0,0)	(1,0.5,0)	(1,0.5,-1)
1	47.86	**66.67**	54.52	54.05
2	66.43	76.43	76.43	**89.29**
3	46.47	48.88	42.63	**55.45**
4	53.19	51.52	55.36	**77.60**
5	66.67	58.33	66.67	**75.00**

At the same time, ID 1 shows lower accuracy for $(1, 0.5, 0), (1, 0.5, -1)$ even though the accuracy of agree/disagree classifier is good. In idea ID 1, the number of remarks per participant is the lowest in data sets, so the errors of the Agreement/Disagreement classifier strongly affect the results of the Support/Oppose classifier.

5 Conclusion and Future Work

We have shown how to classify users in an online debate based on their general positions with regards to the main topic by the textual contents of their remarks and the link structure of their replies. The previous work used the assumption that the replies are usually disagreements and based on this assumption used a link-based method to classify the participants. However, in an online debate the replies are also used for clarifying previous remarks and quite often for supporting the previous ones. Our proposed method uses not only the link structure of the replies, but also the textual contents of the local agreement/disagreement positions between the remarks to boost the accuracy of the task of classifying users into the supporting and opposing parties.

The proposed method is based on the observation that it is easier to use the textual contents for classifying the local positions of a user's replies with regards to the previous remarks, than to use them (e.g., by aggregating them) for classifying his/her global position with regards to the main topic of the debate. In our experiments, we used a rule-based classifier to classify the replies into

agree, disagree, and neutral (with regards to the previous replies) and used these classifier's result to determine the weight of the corresponding links in the link structure of the reply network. The max cut algorithm is then applied to the network, which results in a classification of the users into supporting or opposing parties (with regards to the main topic of the debate). The experiments indicate that the accuracies of the link-based method of (Agrawal et al., 2003) can be significantly increased by considering the textual contents of the replies.

There are several directions to extend our method. When an expression of opinion appears in a reply, we have to locate the target of the opinion. In the current method the target is determined by the ID of the remark pointed by the reply. When the ID is not available, we assume that the reply is with regards to the main topic. However, we also observed that even though a reply was directed to a particular remark, it often also contained opinions about the main topic. Identifying such replies can be used to yield higher accuracy in the classification task.

Much work remains for ultimate understanding of the participants' opinions in debate corpus. Understanding the reasons for the position for the main topic is one of the ways to understand their opinions and it may help to decide the next steps for companies or governments which held the debate sessions. An integrated system that includes a discussion system and an analysis system showing the ratio of positions and the reasons would support such purposes.

Acknowledgments

The authors would like to acknowledge Kenji Hiramoto and Manabu Morita, who are responsible for the IdeaBox, for their helpful comments and conversation.

References

Agrawal, Rakesh, Sridhar Rajagopalan, Ramakrishnan Srikant, and Yirong Xu. 2003. Mining newsgroups using networks arising from social behavior. In *WWW '03: Proceedings of the 12th international conference on World Wide Web*, pages 529–535, New York, NY, USA. ACM.

Dave, Kushal, Martin Wattenberg, and Michael Muller. 2004. Flash forums and forumreader: navigating a new kind of large-scale online discussion. In *CSCW '04: Proceedings of the 2004 ACM conference on Computer supported cooperative work*, pages 232–241, New York, NY, USA. ACM.

Galley, Michel, Kathleen McKeown, Julia Hirschberg, and Elizabeth Shriberg. 2004. Identifying agreement and disagreement in conversational speech: use of bayesian networks to model pragmatic dependencies. In *ACL '04: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, pages 669–676, Morristown, NJ, USA. Association for Computational Linguistics.

Kanayama, Hiroshi, Tetsuya Nasukawa, and Hideo Watanabe. 2004. Deeper sentiment analysis using machine translation technology. In *COLING '04: Proceedings of the 20th international conference on Computational Linguistics*, page 494, Morristown, NJ, USA. Association for Computational Linguistics.

Millen, David R. and Michael A. Fontaine. 2003. Multi-team facilitation of very large-scale distributed meetings. In *ECSCW'03: Proceedings of the eighth conference on European Conference on Computer Supported Cooperative Work*, pages 259–275, Norwell, MA, USA. Kluwer Academic Publishers.

Murakami, Akiko, Katashi Nagao, and Koichi Takeda. 2001. Discussion Mining: Knowledge discovery from online discussion records. In *NLPRS Workshop XML and NLP, 2001*.

Murakami, Akiko, Tetsuya Nasukawa, Fusashi Nakamura, Hironori Takeuchi, Risa Nishiyama, Pnina Veisberg, and Hideo Watanabe. 2007. InnovationJam: Analysis of online discussion records using text mining technology. In *International Workshop on Intercultual Collaboration 2007 (IWIC2007)*.

Rendl, Franz, Giovanni Rinaldi, and Angelika Wiegele. 2010. Solving Max-Cut to optimality by intersecting semidefinite and polyhedral relaxations. *Math. Programming*, 121(2):307.

Thomas, Matt, Bo Pang, and Lillian Lee. 2006. Get out the vote: Determining support or opposition from congressional floor-debate transcripts. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 327–335, Sydney, Australia, July. Association for Computational Linguistics.

Semantic Classification of Automatically Acquired Nouns using Lexico-Syntactic Clues

Yugo Murawaki
Graduate School of Informatics
Kyoto University
murawaki@nlp.kuee.kyoto-u.ac.jp

Sadao Kurohashi
Graduate School of Informatics
Kyoto University
kuro@i.kyoto-u.ac.jp

Abstract

In this paper, we present a two-stage approach to acquire Japanese unknown morphemes from text with full POS tags assigned to them. We first acquire unknown morphemes only making a morphology-level distinction, and then apply semantic classification to acquired nouns. One advantage of this approach is that, at the second stage, we can exploit syntactic clues in addition to morphological ones because as a result of the first stage acquisition, we can rely on automatic parsing. Japanese semantic classification poses an interesting challenge: proper nouns need to be distinguished from common nouns. It is because Japanese has no orthographic distinction between common and proper nouns and no apparent morphosyntactic distinction between them. We explore lexico-syntactic clues that are extracted from automatically parsed text and investigate their effects.

1 Introduction

A dictionary plays an important role in Japanese morphological analysis, or the joint task of segmentation and part-of-speech (POS) tagging (Kurohashi et al., 1994; Asahara and Matsumoto, 2000; Kudo et al., 2004). Like Chinese and Thai, Japanese does not delimit words by white-space. This makes the first step of natural language processing more ambiguous than simple POS tagging. Accordingly, morphemes in a pre-defined dictionary compactly represent our knowledge about both segmentation and POS.

One obvious problem with the dictionary-based approach is caused by unknown morphemes, or morphemes not defined in the dictionary. Even though, historically, extensive human resources were used to build high-coverage dictionaries (Yokoi, 1995), texts other than newspaper articles, in particular web pages, contain a large number of unknown morphemes. These unknown morphemes often cause segmentation errors. For example, morphological analyzer JUMAN 6.0[1] wrongly segments the phrase "さっぽろ駅" (*saQporo eki*, "Sapporo Station"), where "さっぽろ" (*saQporo*) is an unknown morpheme, as follows:

"さ" (*sa*, noun-common, "difference"),
"っ" (*Q*, UNK), "ぽ" (*po*, UNK),
"ろ" (*ro*, noun-common, "sumac") and
"駅" (*eki*, noun-common, "station"),

where UNK refers to unknown morphemes automatically identified by the analyzer. Such an erroneous sequence has disastrous effects on applications of morphological analysis. For example, it can hardly be identified as a LOCATION in named entity recognition.

One solution to the unknown morpheme problem is unknown morpheme acquisition (Mori and Nagao, 1996; Murawaki and Kurohashi, 2008). It is the task of automatically augmenting the dictionary by acquiring unknown morphemes from text. In the above example, the goal is to acquire the morpheme "さっぽろ" (*saQporo*) with the POS tag "noun-location name." However, unknown morpheme acquisition usually adopts a coarser POS tagset that only represents the morphology level distinction among noun, verb and adjective. This means that "さっぽろ" (*saQporo*) is acquired as just a noun and that the semantic label "location name" remains to be assigned. The reason only the morphology level distinction is made is

[1] http://nlp.kuee.kyoto-u.ac.jp/nl-resource/juman-e.html

Coling 2008: Poster Volume, pages 876–884,
Beijing, August 2010

that the semantic level distinction cannot easily be captured with morphological clues that are exploited in unknown morpheme acquisition.

In this paper, we investigate the remaining problem and introduce the new task of semantic classification that is to be applied to automatically acquired nouns. In this task, we can exploit syntactic clues in addition to morphological ones because, as a result of acquisition, we can now rely on automatic parsing. For example, since text containing "さっぽろ" (*saQporo*, noun-*unclassified*) is correctly segmented, we can extract not only the phrase "*saQporo* station," but the tree fragment "ϕ go to *saQporo*," and we can determine its semantic label.

Japanese semantic classification poses an interesting challenge: proper nouns need to be distinguished from common nouns. Like Chinese and Thai, Japanese has no orthographic distinction between common and proper nouns as there is no such thing as capitalization. In addition, there seems no morphosyntactic (i.e. grammatical) distinction between them.

In this paper, we explore lexico-syntactic clues that can be extracted from automatically parsed text. We train a classification model on manually registered nouns and apply it to automatically acquired nouns. We then investigate the effects of lexico-syntactic clues.

2 Semantic Classification Task

2.1 Two-Stage Approach to Unknown Morpheme Acquisition

Our goal is to identify unknown morphemes in unsegmented text and assign POS tags to them. In this section, we omit the details of boundary identification (segmentation) and review the Japanese POS tagset to see why we propose a two-stage approach to assign full POS tags.

The Japanese POS tagset derives from traditional grammar. It is a mixture of several linguistic levels: morphology, syntax and semantics. In other words, information encoded in a POS tag is more than how the morpheme behaves in a sequence of morphemes. In fact, POS tags given to pre-defined morphemes are useful for applications of morphological analysis, such as dependency parsing (Kudo and Matsumoto, 2002), named entity recognition (Asahara and Matsumoto, 2003; Sasano and Kurohashi, 2008) and anaphora resolution (Iida et al., 2009; Sasano and Kurohashi, 2009). In these applications, POS tags are incorporated as features for models.

On the other hand, the mixed nature of the POS tagset poses a challenge to unknown morpheme acquisition. Previous approaches (Mori and Nagao, 1996; Murawaki and Kurohashi, 2008) directly or indirectly reply on morphology, or our knowledge on how a morpheme behaves in a sequence of morphemes. This means that semantic level distinction is difficult to make in these approaches, and in fact, is left unresolved. To be specific, nouns are only distinguished from verbs and adjectives but they have subcategories in the original tagset. These are what we try to classify acquired nouns into in this paper.

2.2 Semantic Labels

The Japanese noun subcategories may require an explanation since they are different from the English ones (Marcus et al., 1993) in many respects. Singular and mass nouns are not distinguished from plural nouns because Japanese has no grammatical distinction between them. More importantly for this paper, proper nouns have subcategories such as person name, location name and organization name in addition to the distinction from common nouns. These subcategories provide important information to named entity recognition among other applications. For proper nouns, we adopt these subcategories as semantic labels in our task.

In contrast to proper nouns, common nouns have only one subcategory "common." However, we consider that subcategories of common nouns similar to those of proper nouns are useful for, for example, anaphora resolution (Sasano and Kurohashi, 2009). We adopt the "categories" of morphological analyzer JUMAN, with which common nouns in its dictionary are annotated. There are 22 "categories" including `PERSON`, `ORGANIZATION` and `CONCEPT`. We collapse these "categories" into coarser semantic labels that roughly correspond to those for proper nouns. To sum up, we define 9 semantic labels as shown

Table 1: List of semantic labels.

labels	P/C	sources[1]	manually registered nouns	automatically acquired nouns
PSN-P	proper	subPOS:person name	松井 (*matsui*, a surname) ジョージ (*jôji*, "George")	佐祐理 (*sayuri*, a given name) キョン (*kyoN*, a nickname)
LOC-P		subPOS:place name	京都 (*kyouto*, "Kyoto") ドイツ (*doitsu*, "Germany")	アキバ (*akiba*, "Akihabara") ワイキキ (*waikiki*, "Waikiki")
ORG-P		subPOS:organization name	日銀 (*nichigin*, a bank) NHK (a broadcaster)	マツダ (*matsuda*, "Mazda") ヤフー (*yahû*, "Yahoo")
OTH-P		subPOS:proper noun	平成 (*heisei*, an era name) スラブ (*surabu*, "Slav")	ジプシー (*jipushî*, "Gypsy")
PSN-C	common	category:PERSON	先生 (*seNsei*, "teacher") スタッフ (*sutaQfu*, "staff")	メル友 (*merutomo*, "keypal") ニート (*nîto*, "NEET")
LOC-C		category:PLACE-*[2]	職場 (*shokuba*, "office") カフェ (*kafe*, "cafe")	囲炉裏 (*irori*, "hearth") 圃場 (*hojou*, "farm field")
ORG-C		category:ORGANIZATION	政府 (*seifu*, "government") チーム (*chîmu*, "team")	メーカ (*mêka*, "manufacturer") 弊所 (*heisho*, "our office")
ANI-C		category:ANIMAL and category:ANIMAL-PART	犬 (*inu*, "dog") 顔 (*kao*, "face")	チワワ (*chiwawa*, "Chihuahua") マンタ (*maNta*, "manta")
OTH-C		other categories	主張 (*shuchou*, "argument") 枕 (*makura*, "pillow")	甚平 (*jiNbei*, a kind of clothing) 着メロ (*chakumero*, "ringtone")

[1] A subPOS refers to a subcategory of noun. For example, PSN-P corresponds to the POS tag "noun-person name".
[2] category:PLACE-INSTITUTION, category:PLACE-INSTITUION PART and others.

in Table 1.

2.3 Related Tasks

A line of research is dedicated to identify unknown morphemes with varying degrees of identification. Asahara and Matsumoto (2004) only focus on boundary identification (segmentation) of unknown morphemes. Mori and Nagao (1996), Nagata (1999) and Murawaki and Kurohashi (2008) assign POS tags at the morphology level. Uchimoto et al. (2001) assign full POS tags but unsurprisingly the accuracy is low. Nakagawa and Matsumoto (2006) also assign full POS tags. They address the fact that local information used in previous studies is inherently insufficient and present a method that uses global information, in other words, takes into consideration all occurrences of each unknown word in a document. They report an improvement in tagging proper nouns in Japanese.

A related task is named entity recognition (NER). It can handle a named entity longer than a single morpheme and is usually formalized as a chunking problem. Since Japanese does not delimit words by white-space, the unit of chunking can be a character (Asahara and Matsumoto, 2003; Kazama and Torisawa, 2008) or a morpheme (Sasano and Kurohashi, 2008). In either case, NER models encode the output of morphological analysis and therefore are affected by its errors. In fact, Saito et al. (2007) report that a majority of unknown named entities (those never appear in a training corpus) contain unknown morphemes as their constituents and that NER models perform poorly on them. A straightforward solution to this problem would be to acquire unknown morphemes and to assign semantic labels to them.

Another related task is supersense tagging (Ciaramita and Johnson, 2003; Curran, 2005; Ciaramita and Altun, 2006). A supersense corresponds to one of the 26 broad categories defined by WordNet (Fellbaum, 1998). Each noun synset is associated with a supersense. For example, "chair" has supersenses PERSON, ARTIFACT and ACT because it belongs to several synsets.

Since supersense tagging is studied in English, it differs from our task in several respects. In English, the distinction between common and proper nouns is clear. In fact, the tagging models can use POS features even for unknown nouns. In addition, the syntactic behavior of English nouns is different from that of Japanese nouns (Gil, 1987). Definiteness is not marked in Japanese as it lacks determiners (e.g. "the" and "a"), and Japanese has no obligatory plural marking. On the other hand, Japanese obligatorily uses numeral classifiers to indicate the count of nouns, as in

(1) *saN satsu no hoN*
three CL GEN book
three volumes of books, or three books,

where "*satsu*" is a numeral classifier for books. A number together with its numeral classifier forms a numeral quantifier. Numeral quantifiers would be informative about the semantic categories of nouns. Note that Japanese shares the above features with Chinese and Thai. Our findings in this paper may hold for these languages.

3 Proposed Method

3.1 Lexico-Syntactic Clues

In the task of semantic classification, we can exploit syntactic clues in addition to morphological ones. As a result of unknown morpheme acquisition, text containing acquired morphemes, or former unknown morphemes, is correctly segmented. Now we can treat automatic parsing as (at least partly) reliable with regard to acquired morphemes.

For noun X, we use the following sets of features for classification.

call: noun phrase Y that appears in a pattern like "Y called X" and "Y such as X," e.g. "`call`:*kuni*" from

X to iu kuni
X QT call country
a country called X.

cf: predicate with a case marker with which it takes X as an argument, e.g. "`cf`:*tooru*:*wo*" from

X wo tooru
X ACC pass
ϕ pass through X.

demo: demonstrative that modifies X, e.g. "`demo`:*kono*" from "*kono X*" (this X) and "`demo`:*doNna*" from "*doNna X*" (what kind of X).

ncf1: noun phrase which X modifies with the genitive case marker "*no*," e.g. "`ncf1`:*heya*" from

X no heya
X GEN room
X's room.

ncf2: noun phrase that modifies X with the genitive case marker "*no*," e.g. "`ncf2`:*subete*" from

subete no X
all GEN X
all X.

suf: suffix or suffix-like noun that follows X, e.g. "`suf`:*saN*" from "X *saN*" (Mr./Ms. X) and "`suf`:*eki*" from "X *eki*" (X station).

Using automatically parsed text to extract syntactic features has an advantage. Since no manual annotation is necessary, we can utilize a huge raw corpus. On the other hand, parsing errors are inevitable. However, we can circumvent this problem by using the constraints of Japanese dependency structures: head-final and projective. The simplest example is the second last element of a sentence, which always depends on the last element. With these constraints, we can focus on syntactically unambiguous dependency pairs and extract syntactic features accurately. We follow Kawahara and Kurohashi (2001) to extract a pair of an argument noun and a predicate (**cf**), and Sasano et al. (2004) to extract a pair of nouns connected with the genitive case marker "*no*" (**ncf1** and **ncf2**).

Noun X can be part of a compound noun. We leave it for named entity recognition. Except for **suf**, we extract features only when X alone forms a word. Similarly, we extract **suf** features only when X and a suffix alone form a noun phrase.

For **call**, **ncf1**, and **ncf2**, we generalize numerals within noun phrases. For "*hoN*" (book) in example 1, we extract the feature "`ncf2`:<NUM>*satsu*."

3.2 Instances for Classification

Now that features are extracted for each noun, the question is how to combine them together to make an instance for classification. One factor we need to consider is polysemy: a noun can be a person name in one context and a location name in another. If we combine features extracted from the whole corpus, they may represent several semantic labels.

Modeling a mixture of semantic labels might be a solution, but we do not take this approach on the grounds that each occurrence of a noun corresponds to a single semantic label.

In our strategy, we perform classification multiple times for each noun and aggregate the results at the end. The features for each classification are extracted from a relatively small subset of a corpus where the noun is supposedly consistent in

terms of semantic labels. In the field of named entity recognition, it is known that label consistency holds strongly at the level of a document and less strongly across different documents (Krishnan and Manning, 2006). Thus we start with a document and gradually cluster related documents until a sufficient number of features are obtained. For the specific procedures we took in the experiments, see Section 4.1.

3.3 Training Data

Following unknown morpheme acquisition (Murawaki and Kurohashi, 2008), we create training data using manually registered nouns, for which we can obtain correct semantic labels. We perform the same procedure as above to make instances of registered nouns.

Some registered nouns are tagged with more than one semantic label, which we call "explicit polysemy." We drop them from the training data. The remaining problem is "implicit polysemy." Nouns are sometimes used with an uncovered sense. In preliminary experiments, we found that a typical case of implicit polysemy was that a proper noun derived from a basic noun. To alleviate this problem, we use an NE tagger for filtering. We run an NE tagger over a small portion of the corpus and extract common nouns that are frequently tagged as named entities. Then we remove these nouns from the training data.

We also drop nouns that appear extremely frequently such as "人" (*hito*, "person"), "事" (*koto*, "thing") and "私" (*watashi*, "I"[2]). Since acquired nouns to be classified are typically low frequency morphemes, they would not behave similarly to these basic nouns.

3.4 Classifier

To assign a semantic label to each instance, we use a multiclass discriminative classifier. The input it takes is an instance that is represented by a feature vector $x \in \mathbb{R}^d$. The output is one semantic label $y \in Y$, where Y is the set of semantic labels.

We use a linear classifier. It has a weight vector $w_y \in \mathbb{R}^d$ for each y and outputs y that maximizes the inner product of w_y and x.

$$y = \operatorname*{argmax}_y \langle w_y, x \rangle.$$

Several methods have been proposed to estimate weight vector w_y from training data. We use online algorithms because they are easy to implement and scale to huge instances. We try the Perceptron family of algorithms.

4 Experiments

4.1 Settings

We used JUMAN for morphological analysis and KNP[3] for dependency parsing. The dictionary of JUMAN was augmented with automatically acquired morphemes (Murawaki and Kurohashi, 2008). The number of manually registered morphemes was 120 thousands while there were 13,071 acquired morphemes, of which 12,615 morphemes were nouns.

We used a web corpus that was compiled through the procedures proposed by Kawahara and Kurohashi (2006). It consisted of 100 million pages.

We first extracted features from the web corpus. To keep the model size manageable, we used 447,082 features that appeared more than 100 times in the corpus.

We constructed training data from manually registered nouns and test data from automatically acquired nouns. For each noun, we combined text together until the number of features grew to more than 100. We started with a single web page, then merge pages that share a domain name and finally clustered texts across different domains. We split the web corpus into 40 subcorpora and applied this procedure in parallel. We used Bayon[4] for clustering domain texts. We sequentially read texts and applied the repeated bisections clustering every time some 5,000 pages were appended. The vectors for clustering were nouns, both registered and acquired, with their tf-idf scores. We obtained 4,843,085 instances for 10,613 registered nouns and 196,098 instances for 2,556 acquired nouns.

[2] Japanese personal pronouns are treated as common nouns because they show no special morphosyntactic behavior.

[3] http://nlp.kuee.kyoto-u.ac.jp/nl-resource/knp-e.html

[4] http://code.google.com/p/bayon/

Table 2: Results of semantic classification.

learning algorithms	acquired nouns	registered nouns
Averaged Perceptron	86.40% (432 / 500)	88.59% (123,113 / 138,971)
Passive-Aggressive	**87.00**% (435 / 500)	**91.68**% (127,407 / 138,971)
Confidence-Weighted	85.20% (426 / 500)	89.66% (124,604 / 138,971)
baseline[1]	69.60% (348 / 500)	79.14% (109,980 / 138,971)

[1] assign OTH−C to all instances.

Table 3: Examples of aggregated instances.

acquired nouns	instances	labels
ヒカル (*hikaru*, a person name)	84	PSN−P:58.33%, PSN−C:41.67%
チワワ (*chiwawa*, "Chihuahua")	128	ANI−C:54.69%, OTH−C:45.31%
かみさん (*kamisaN*, colloq. "wife")	131	PSN−C:100%
ラスベガス (*rasubegasu*, "Las Vegas")	136	LOC−P:97.06%, LOC−C:2.94%
アップル (*aQpuru*, "Apple/apple")	187	ORG−P:63.10%, PSN−C:34.76%, OTH−C:2.14%
メルマガ (*merumaga*, abbr. of "mail magazine")	1,622	OTH−C:99.32%, LOC−C:0.55%, PSN−C:0.06%

In order to handle polysemy, we evaluated semantic classification on an instance-by-instance basis. We randomly selected 500 instances from the test data and manually assigned the correct labels to them. For comparison purposes, we also classified registered nouns. We split the training data: 829 nouns or 138,971 instances for testing and the rest for training.

We trained the model with three online learning algorithms, (1) the averaged version (Collins, 2002) of Perceptron (Crammer and Singer, 2003), (2) the Passive-Aggressive algorithm (Crammer et al., 2006), and (3) the Confidence-Weighted algorithm (Crammer et al., 2009). For Passive-Aggressive algorithm, we used *PA-I* and set parameter C to 1. For Confidence-Weighted, we used the single-constraint updates. All algorithms iterated five times through the training data.

4.2 Results

Table 2 shows the results of semantic classification. All algorithms significantly improved over the baseline. As suggested by the gap in accuracy between acquired and registered nouns in the baseline method, the label distribution of the training data differed from that of the test data, but the decrease in accuracy was smaller than expected.

The Passive-Aggressive algorithm performed best on both acquired and registered nouns. For the rest of this paper, we report the results of the Passive-Aggressive algorithm.

Table 3 shows aggregated instances of some acquired nouns. Although classification sometimes failed, correct labels took the majority. However, it is noticeable that PSN−P was frequently misidentified as PSN−C while PSN−C was correctly identified. This phenomenon is clearly seen in the confusion matrix (Table 4). Half of PSN−P instances were misidentified as PSN−C but the percentage of errors in the opposite direction was just above 9%. We will investigate this in the next section.

4.3 Discussion

Our interest is in determining what kinds of features are effective in semantic classification. We first performed standard ablation experiments. We trained a series of models on the training data after removing each feature set. The training and test data were the same with those in Section 4.1.

Table 5 shows the results of ablation experiments. Significant decreases in accuracy are observed in the **cf** dataset. This is easily explained by the fact that more than half of features belonged to **cf**. The ratio of **ncf1** was much the same with that of **ncf2**, but the removal of **ncf1** resulted in a worse performance in classifying registered nouns than that of **ncf2**. This means that a modifiee of a noun explains more about the noun than its modifier.

The ablation experiments cannot capture interesting properties of features because each feature set has a great diversity within it. Next, we directly examine features instead. Since we use a simple linear classifier, a feature has $|Y|$ corresponding weights, each of which represents how likely a noun belongs to label y. For example, features whose weights for PSN−C are the largest

Table 4: Confusion matrix of acquired nouns.

		Actual								
		PSN-P	LOC-P	ORG-P	OTH-P	PSN-C	LOC-C	ORG-C	ANI-C	OTH-C
Predicted	PSN-P	16		1		4				1
	LOC-P									1
	ORG-P			4						
	OTH-P									
	PSN-C	16				39			1	2
	LOC-C	2	2	1			10			4
	ORG-C									2
	ANI-C								28	
	OTH-C	3	1	1		1	13		9	338

Table 5: Results of ablation experiments.

feature set	ratio[1]	acquired nouns	registered nouns
-call	0.23%	87.60% (438 / 500)	91.58% (127,276 / 138,971)
-cf	54.84%	84.80% (424 / 500)	88.96% (123,630 / 138,971)
-demo	2.40%	**88.00**% (440 / 500)	91.38% (126,996 / 138,971)
-ncf1	19.03%	87.20% (436 / 500)	89.23% (124,008 / 138,971)
-ncf2	18.40%	85.60% (428 / 500)	91.54% (127,220 / 138,971)
-suf	5.10%	87.40% (437 / 500)	91.30% (126,889 / 138,971)
all		87.00% (435 / 500)	**91.68**% (127,407 / 138,971)

[1] The proportion of each feature set that appears in the instances of the test data.

include:

- cf:*nakusu*:*wo* ("ϕ lose X to the disease"),
- cf:*oshieru*:*ni* ("ϕ_1 teach X ϕ_2"),
- ncf2:*ooku* ("many/much X"), and
- ncf2:<NUM>*niN* (X is modified by <NUM> plus a numeral classifier for persons).

As briefly mentioned in Section 2.3, Japanese numeral quantifiers received scholarly attention in the fields of linguistic philosophy and linguistics in relation to the count/mass distinction (Quine, 1969; Gil, 1987). In our feature sets, numeral quantifiers typically appear as **ncf2**, e.g. "ncf2:<NUM>*niN*." The weights given to them demonstrate their effectiveness in semantic classification. They discriminate common nouns from proper nouns as the weights given to common nouns are larger with wide margins. It is not surprising because, say, the phrase "two Johns" is semantically acceptable but extremely rare in reality. They are also informative about the distinction among PSN, LOC and others. For example, the classifier "*niN*" for persons suggest the noun in question is a person while "*keN*" for houses would modify a location-like noun. However, we found quite a few "noises" about these features in data. The modifiee of a numeral expression is not always the noun to be counted, as demonstrated by the following example:

(2) *saN niN no moNdai*
three CL GEN problem
matters among the three persons.

From the above, the feature "ncf2:<NUM>*niN*" is extracted although "*moNdai*" is OTH-C. Theis "noise" is attributed to the genitive case marker "*no*" because it can denote a wide range of relations between two nouns. We might be able to avoid this problem if we focus on "floating" numeral quantifiers. A floating numeral quantifier has no direct dependency relation to the noun to be counted, as in

(3) *seito ga saN niN keQseki shita*
student NOM three CL absence do
three students were absent,

where the numeral quantifier modifies the verb phrase instead of the noun. Further work is needed to anchor floating numeral quantifiers since they bring a different kind of ambiguity themselves (Bond et al., 1998).

Closely related to numeral quantifiers are quantificational nouns that appear as "ncf2:*ooku*" ("many/much"), "ncf2:*subete*" ("all") and others. They distinguish common nouns from proper

nouns but does not make a further classification. The same is true of other numeral expressions such as "cf:*hueru*:*ga*" ("X increase in number") and "cf:*nai*:*ga*" ("there is no X" or "X do not exist"). We found that, other than numeral expressions, some features distinguished common nouns from proper nouns because they indicated the noun denoted an attribute. Such features include "cf:*naru*:*ni*" ("ϕ become X") and "cf:*kaneru*:*wo*" ("ϕ double as X").

We expected that demonstratives (**demo**) served similar functions to quantificational expressions, but it turned out to be more complex. The distal demonstrative "*ano*" ("that") often modifies proper nouns to give emphasis. In fact, the model gave larger weights to proper nouns. On the other hand, interrogative demonstratives such as "*dono*" ("which") and "*doNna*" ("what kind of") are rarely used with proper nouns although semantically acceptable.

As seen above, there is an abundant variety of features that distinguish common nouns from proper nouns. Also, it is not difficult to make a distinction among PSN, LOC and others although the far largest cluster OTH−C sometimes absorbs other instances. The remaining question is how to distinguish proper nouns from common nouns, or specifically PSN−P from PSN−C. We examined features that gave larger weights to PSN−P than to PSN−C. They generally had smaller margins in weights than those which distinguish PSN−C from PSN−P. Among them, features such as "cf:*utau*:*ga*" ("X sing") and "cf:*hanasu*:*ni*" ("ϕ talk to X") have no problem with being used for common nouns in terms of both semantics and pragmatics. They seem to have resulted from over-training. There were seemingly appropriate features such as "suf:*saNchi*" ("X's house") and "suf:*seNshu*" (honorific suffix for players), but they were not ubiquitous in the corpus. PSN−P instances suffered from lack of distinctive features.

One solution to this problem is to combine additional knowledge about person names. For example, a Japanese family name is followed by a given name, and most Chinese names consist of three Chinese characters. However, quite a few person names in the web corpus do not follow the usual patterns of person names because they are handles (or nicknames) and names for fictional characters. Thus it would be desirable to be able to classify person names without additional knowledge.

5 Conclusion

In this paper, we presented the new task of semantic classification of Japanese nouns and applied it to nouns automatically acquired from text. Unlike in unknown morpheme identification in previous studies, we can exploit automatically parsed text. We explored lexico-syntactic clues and investigated their effects. We found plenty of features that distinguished common nouns from proper nouns, but few features worked in the opposite direction. Further work is needed to overcome this bias.

References

Asahara, Masayuki and Yuji Matsumoto. 2000. Extended models and tools for high-performance part-of-speech tagger. In *Proc. of COLING 2000*, pages 21–27.

Asahara, Masayuki and Yuji Matsumoto. 2003. Japanese named entity extraction with redundant morphological analysis. In *Proc. of HLT/NAACL 2003*, pages 8–15.

Asahara, Masayuki and Yuji Matsumoto. 2004. Japanese unknown word identification by character-based chunking. In *Proc. COLING 2004*, pages 459–465.

Bond, Francis, Daniela Kurz, and Satoshi Shirai. 1998. Anchoring floating quantifiers in Japanese-to-English machine translation. In *Proc. of COLING 1998*, pages 152–159.

Ciaramita, Massimiliano and Yasemin Altun. 2006. Broad-coverage sense disambiguation and information extraction with a supersense sequence tagger. In *Proc. of EMNLP 2006*, pages 594–602.

Ciaramita, Massimiliano and Mark Johnson. 2003. Supersense tagging of unknown nouns in WordNet. In *Proc. of EMNLP 2003*, pages 168–175.

Collins, Michael. 2002. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *Proc. of EMNLP 2002*, pages 1–8.

Crammer, Koby and Yoram Singer. 2003. Ultraconservative online algorithms for multiclass problems. *Journal of Machine Learning Research*, 3:951–991.

Crammer, Koby, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. 2006. Online passive-aggressive algorithms. *Journal of Machine Learning Research*, 7:551–585.

Crammer, Koby, Mark Dredze, and Alex Kulesza. 2009. Multi-class confidence weighted algorithms. In *Proc. of EMNLP 2009*, pages 496–504.

Curran, James R. 2005. Supersense tagging of unknown nouns using semantic similarity. In *Proc. of ACL 2005*, pages 26–33.

Fellbaum, Christiane, editor. 1998. *WordNet: An Electronic Lexical Database*. The MIT Press, Cambridge, MA.

Gil, David. 1987. Definiteness, NP configurationality and the count-mass distinction. In Reuland, Eric J. and Alice G. B. ter Meulen, editors, *The Representation of (In)definiteness*, pages 254–269. MIT Press.

Iida, Ryu, Kentaro Inui, and Yuji Matsumoto. 2009. Capturing salience with a trainable cache model for zero-anaphora resolution. In *Proc. of ACL/IJCNLP 2009*, pages 647–655.

Kawahara, Daisuke and Sadao Kurohashi. 2001. Japanese case frame construction by coupling the verb and its closest case component. In *Proc. of HLT 2001*, pages 204–210.

Kawahara, Daisuke and Sadao Kurohashi. 2006. Case frame compilation from the web using high-performance computing. In *Proc. of LREC-06*, pages 1344–1347.

Kazama, Jun'ichi and Kentaro Torisawa. 2008. Inducing gazetteers for named entity recognition by large-scale clustering of dependency relations. In *Proc. of ACL 2008*, pages 407–415, June.

Krishnan, Vijay and Christopher D. Manning. 2006. An effective two-stage model for exploiting non-local dependencies in named entity recognition. In *Proc. of COLING-ACL 2006*, pages 1121–1128.

Kudo, Taku and Yuji Matsumoto. 2002. Japanese dependency analysis using cascaded chunking. In *Proc. of CONLL 2002*, pages 1–7.

Kudo, Taku, Kaoru Yamamoto, and Yuji Matsumoto. 2004. Applying conditional random fields to Japanese morphological analysis. In *Proc. of EMNLP 2004*, pages 230–237.

Kurohashi, Sadao, Toshihisa Nakamura, Yuji Matsumoto, and Makoto Nagao. 1994. Improvements of Japanese morphological analyzer JUMAN. In *Proc. of The International Workshop on Sharable Natural Language Resources*, pages 22–38.

Marcus, Mitchell P., Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of English: the Penn treebank. *Computational Linguistics*, 19(2):313–330.

Mori, Shinsuke and Makoto Nagao. 1996. Word extraction from corpora and its part-of-speech estimation using distributional analysis. In *Proc. of COLING 1996*, volume 2, pages 1119–1122.

Murawaki, Yugo and Sadao Kurohashi. 2008. Online acquisition of Japanese unknown morphemes using morphological constraints. In *Proc. of EMNLP 2008*, pages 429–437.

Nagata, Masaaki. 1999. A part of speech estimation method for Japanese unknown words using a statistical model of morphology and context. In *Proc. of ACL 1999*, pages 277–284.

Nakagawa, Tetsuji and Yuji Matsumoto. 2006. Guessing parts-of-speech of unknown words using global information. In *Proc. of COLING-ACL 2006*, pages 705–712.

Quine, Willard Van. 1969. *Ontological Relativity and Other Essays*. Columbia University Press.

Saito, Kuniko, Jun Suzuki, and Kenji Imamura. 2007. Extraction of named entities from blogs using CRF. In *Proc. of The 13th Annual Meeting of The Association for Natural Language Processing*, pages 107–110. (in Japanese).

Sasano, Ryohei and Sadao Kurohashi. 2008. Japanese named entity recognition using structural natural language processing. In *Proc. of IJCNLP 2008*, pages 607–612.

Sasano, Ryohei and Sadao Kurohashi. 2009. A probabilistic model for associative anaphora resolution. In *Proc. of EMNLP 2009*, pages 1455–1464.

Sasano, Ryohei, Daisuke Kawahara, and Sadao Kurohashi. 2004. Automatic construction of nominal case frames and its application to indirect anaphora resolution. In *Proc. of COLING 2004*, pages 1201–1207.

Uchimoto, Kiyotaka, Satoshi Sekine, and Hitoshi Isahara. 2001. The unknown word problem: a morphological analysis of Japanese using maximum entropy aided by a dictionary. In *Proc. of EMNLP 2001*, pages 91–99.

Yokoi, Toshio. 1995. The EDR electronic dictionary. *Communications of the ACM*, 38(11):42–44.

A Learnable Constraint-based Grammar Formalism

Smaranda Muresan
School of Communication and Information
Rutgers University
smuresan@rci.rutgers.edu

Abstract

Lexicalized Well-Founded Grammar (LWFG) is a recently developed syntactic-semantic grammar formalism for deep language understanding, which balances expressiveness with provable learnability results. The learnability result for LWFGs assumes that the semantic composition constraints are learnable. In this paper, we show what are the properties and principles the semantic representation and grammar formalism require, in order to be able to learn these constraints from examples, and give a learning algorithm. We also introduce a LWFG parser as a deductive system, used as an inference engine during LWFG induction. An example for learning a grammar for noun compounds is given.

1 Introduction

Recently, several machine learning approaches have been proposed for mapping sentences to their formal meaning representations (Ge and Mooney, 2005; Zettlemoyer and Collins, 2005; Muresan, 2008; Wong and Mooney, 2007; Zettlemoyer and Collins, 2009). However, only few of them integrate the semantic representation with a grammar formalism: λ-expressions and Combinatory Categorial Grammars (CCGs) (Steedman, 1996) are used by Zettlemoyer and Collins (2005;2009), and ontology-based representations and Lexicalized Well-Founded Grammars (LWFGs) (Muresan and Rambow, 2007) are used by Muresan (2008).

An advantage of the LWFG formalism, compared to most constraint-based grammar formalisms developed for deep language understanding, is that it is accompanied by a learnability guarantee, the search space for LWFG induction being a complete grammar lattice (Muresan and Rambow, 2007). Like other constraint-based grammar formalisms, the semantic structures in LWFG are composed by constraint solving, semantic composition being realized through constraints at the grammar rule level. Moreover, semantic interpretation is also realized through constraints at the grammar rule level, providing access to meaning during parsing.

However, the learnability result given by Muresan and Rambow (2007) assumed that the grammar constraints were learnable. In this paper we present the properties and principles of the semantic representation and grammar formalism that allow us to learn the semantic composition constraints. These constraints are a simplified version of "path equations" (Shieber et al., 1983), and we present an algorithm for learning these constraints from examples (Section 5). We also present a LWFG parser as a deductive system (Shieber et al., 1995) (Section 3). The LWFG parser is used as an innate inference engine during LWFG learning, and we present an algorithm for learning LWFGs from examples (Section 4). A discussion and an example of learning a grammar for noun compounds are given is Section 6.

2 Lexicalized Well-Founded Grammars

Lexicalized Well-Founded Grammar (LWFG) is a recently developed formalism that balances expressiveness with provable learnability results (Muresan and Rambow, 2007). LWFGs are a type of Definite Clause Grammars (Pereira and Warren, 1980) in which (1) the context-free backbone is extended by introducing a partial ordering relation among nonterminals, 2) grammar nonterminals are augmented with strings and their syntactic-semantic representations, called *semantic molecules*, and (3) grammar rules can have

Coling 2008: Poster Volume, pages 885–893,
Beijing, August 2010

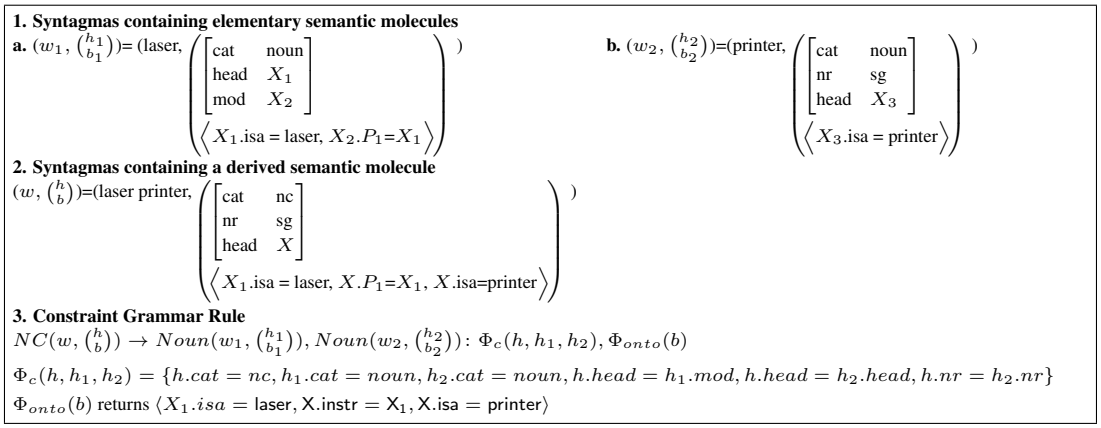


Figure 1: Syntagmas containing elementary semantic molecules (1) and a derived semantic molecule (2); A constraint grammar rule together with the semantic composition and ontology-based interpretation constraints, Φ_c and Φ_{onto} (3)

two types of constraints, one for semantic composition and one for semantic interpretation. The first property allows LWFG learning from a small set of examples. The last two properties make LWFGs a type of syntactic-semantic grammars.

Definition 1. *A semantic molecule associated with a natural language string w, is a syntactic-semantic representation, $w' = \binom{h}{b}$, where h (head) encodes compositional information, while b (body) is the actual semantic representation of the string w.*

Grammar nonterminals are augmented with pairs of strings and their semantic molecules. These pairs are called *syntagmas*, and are denoted by $\sigma = (w, w') = (w, \binom{h}{b})$.

Examples of semantic molecules for the nouns *laser* and *printer* and the noun-noun compound *laser printer* are given in Figure 1. When associated with lexical items, semantic molecules are called *elementary semantic molecules*. When semantic molecules are built by the combination of others, they are called *derived semantic molecules*. Formally, the semantic molecule head, h, is a one-level feature structure (i.e., values are atomic), while the semantic molecule body, b, is a logical form built as a conjunction of atomic predicates $\langle concept\rangle.\langle attr\rangle = \langle concept\rangle$, where variables are either concept or slot identifiers in an ontology.[1]

Muresan and Rambow (2007) formally defined LWFGs, and we present here a slight modification of their definition.

Definition 2. *A Lexicalized Well-Founded Grammar (LWFG) is a 7-tuple, $G = \langle \Sigma, \Sigma', N_G, \succeq, P_G, P_\Sigma, S\rangle$, where:*

1. *Σ is a finite set of terminal symbols.*
2. *Σ' is a finite set of elementary semantic molecules corresponding to the terminal symbols.*
3. *N_G is a finite set of nonterminal symbols. $N_G \cap \Sigma = \emptyset$. We denote $pre(N_G) \subseteq N_G$, the set of pre-terminals (a.k.a, parts of speech)*
4. *$\succeq$ is a partial ordering relation among nonterminals.*
5. *P_G is the set of constraint grammar rules. A constraint grammar rule is written $A(\sigma) \rightarrow B_1(\sigma_1), \ldots, B_n(\sigma_n)$: $\Phi(\bar{\sigma})$, where $A, B_i \in N_G$, $\bar{\sigma} = (\sigma, \sigma_1, ..., \sigma_n)$ such that $\sigma = (w, w'), \sigma_i = (w_i, w_i'), 1 \leq i \leq n, w = w_1 \cdots w_n, w' = w_1' \circ \cdots \circ w_n'$, and $\circ$ is the composition operator for semantic molecules (more details about the composition operator are given in Section 5). For brevity, we denote a rule by $A \rightarrow \beta$: Φ, where $A \in N_G, \beta \in N_G^+$. P_Σ is the set of constraint grammar rules whose left-hand side are pre-terminals, $A(\sigma) \rightarrow$, $A \in pre(N_G)$.*

[1]The body of a semantic molecule is called OntoSeR and is a flat ontology-based semantic representation.

We use the notation $A \rightarrow \sigma$ for this grammar rules. In LWFG due to partial ordering among nonterminals we can have ordered constraint grammar rules and non-ordered constraint grammar rules (both types can be recursive or non-recursive). A grammar rule $A(\sigma) \rightarrow B_1(\sigma_1), \ldots, B_n(\sigma_n)\colon \Phi(\bar{\sigma})$, is an ordered rule, if for all B_i, we have $A \succeq B_i$. In LWFGs, each nonterminal symbol is a left-hand side in at least one ordered non-recursive rule and the empty string cannot be derived from any nonterminal symbol.

6. *$S \in N_G$ is the start nonterminal symbol, and $\forall A \in N_G, S \succeq A$ (we use the same notation for the reflexive, transitive closure of $\succeq$).*

The partial ordering relation $\succeq$ makes the set of nonterminals well-founded[2] , which allows the ordering of the grammar rules, as well as the ordering of the syntagmas generated by LWFGs. This ordering allow LWFG learning from a small set of representative examples (Muresan and Rambow, 2007) (P_Σ is not learned).

An example of a LWFG rule is given in Figure 1(3). Nonterminals are augmented with syntagmas. Moreover, in LWFG the semantic composition and interpretation are realized via constraints at the grammar rule level ($\Phi(\bar{\sigma})$ in Definition 2). More precisely, syntagma composition means string concatenation ($w = w_1 w_2$) and semantic molecule composition ($\binom{h}{b} = \binom{h_1}{b_1} \circ \binom{h_2}{b_2}$) —- where the bodies of semantic molecules are concatenated through logical conjunction ($b = (b_1, b_2)\nu$, where ν is a variable substitution $\nu = \{X_2/X, X_3/X\}$), while the semantic molecules heads are composed through compositional constraints $\Phi_c(h, h_1, h_2)$, which are a simplified version of "path equations" (Shieber et al., 1983) (see Figure 1(3)). During LWFG learning, *compositional constraints Φ_c are learned* together with the grammar rules. Semantic interpretation, which is ontology-based in LWFG, is also encoded as constraints at the grammar rule level — Φ_{onto} — providing access to meaning during parsing. $\Phi_{onto}(b)$ constraints are applied to the body of the semantic molecule corresponding to the syntagma associated with the left-hand side nonterminal. *The ontology-based constraints are not learned*; rather, Φ_{onto} is a general predicate that succeed or fail as a result of querying an ontology — when it succeeds, it instantiates the variables of the semantic representation with concepts/slots in the ontology (see the example in Figure 1(3)).

[2] $\succeq$ should not be confused with information ordering derived from flat feature structures

2.1 Derivation in LWFG

The derivation in LWFG is called ground syntagma derivation, and it can be seen as the bottom up counterpart of the usual derivation. Given a LWFG, G, the *ground syntagma derivation* relation, $\stackrel{*G}{\Rightarrow}$, is defined as: $\frac{A \rightarrow \sigma}{A \stackrel{*G}{\Rightarrow} \sigma}$ (if $\sigma = (w, w'), w \in \Sigma, w' \in \Sigma'$, i.e., $A \in pre(N_G,)$, and $\frac{B_i \stackrel{*G}{\Rightarrow} \sigma_i,\ i=1,\ldots,n, \quad A(\sigma) \rightarrow B_1(\sigma_1),\ldots,B_n(\sigma_n)\colon \Phi(\bar{\sigma})}{A \stackrel{*G}{\Rightarrow} \sigma}$.

The *set of all syntagmas* generated by *a grammar G* is $L_\sigma(G) = \{\sigma | \sigma = (w, w'), w \in \Sigma^+, \exists A \in N_G, A \stackrel{*G}{\Rightarrow} \sigma\}$. Given a LWFG G, $E_\sigma \subseteq L_\sigma(G)$ is called a sublanguage of G. Extending the notation, given a LWFG G, the set of syntagmas generated by a *rule* $(A \rightarrow \beta\colon \Phi) \in P_G$ is $L_\sigma(A \rightarrow \beta\colon \Phi) = \{\sigma | \sigma = (w, w'), w \in \Sigma^+, (A \rightarrow \beta\colon \Phi) \stackrel{*G}{\Rightarrow} \sigma\}$, where $(A \rightarrow \beta\colon \Phi) \stackrel{*G}{\Rightarrow} \sigma$ denotes the ground derivation $A \stackrel{*G}{\Rightarrow} \sigma$ obtained using the rule $A \rightarrow \beta\colon \Phi$ in the last derivation step.

3 LWFG Parsing as Deduction

Following Shieber (1995), we present the Lexicalized Well-Founded Grammar parser as a deductive proof system in Table 1. The *items* of the logic are of the form $[i, j, \sigma_{ij}, A \rightarrow \alpha \bullet \beta \Phi^A]$, where $A \rightarrow \alpha\beta\colon \Phi^A$ is a grammar rule, Φ^A — the constraints corresponding to the grammar rule whose left-hand side nonterminal is A— can be true, $\bullet$ shows how much of the right-hand side of the rule has been recognized so far, i points to the parent node where the rule was invoked, and j points to the position in the input that the recognition has reached. We use the following notations: $\sigma^R_{ij} = (w^R_{ij}, \binom{h^R_{ij}}{b^R_{ij}})$ are syntagmas corresponding to the partially parsed right-hand side of a rule; $\sigma^L_{ij} = (w^L_{ij}, \binom{h^L_{ij}}{b^L_{ij}})$ are ground-derived syntagmas (i.e., they are augmenting the left-hand side non-

Item form	$[i, j, \sigma_{ij}, A \rightarrow \alpha \bullet \beta \Phi^A]$	$1 \leq i, j \leq n+1, A \in N_G, \alpha\beta \in N_G^*$ the Φ^A constraint can be true
Axioms	$[i, i+1, \sigma^L_{ii+1}, B_i \rightarrow \bullet]$	$1 \leq i \leq n, B_i \in pre(N_G), B_i \rightarrow \sigma^L_{ii+1} \in P_\Sigma$
Goals	$[i, j, \sigma^L_{ij}, A \rightarrow \alpha\Phi^A \bullet]$	$1 \leq i, j \leq n+1, A \in N_G, \alpha \in N_G^+$
Inference Rules		
Prediction	$\frac{[i,j,\sigma^L_{ij}, B \rightarrow \beta \Phi^B \bullet]}{[i,i,\sigma^R_{ii}, A \rightarrow \bullet B\gamma\Phi^A]}$ $\langle A \rightarrow B\gamma \colon \Phi^A \rangle$	$(A \rightarrow B\gamma \colon \Phi^A) \in P_G$ $\sigma^R_{ii} = \sigma_\emptyset$ (i.e., $w^R_{ii} = \epsilon$, $b^R_{ii} = true$ and $h^R_{ii} = \emptyset$)
Completion	$\frac{[i,j,\sigma^R_{ij}, A \rightarrow \alpha \bullet B\ \gamma\ \Phi^A] \quad [j,k,\sigma^L_{jk}, B \rightarrow \beta\ \Phi^B\ \bullet]}{[i,k,\sigma^R_{ik}, A \rightarrow \alpha\ B \bullet \gamma\ \Phi^A]}$	$\sigma^R_{ik} = \sigma^R_{ij} \circ \sigma^L_{jk}$, where $w^R_{ik} = w^R_{ij} w^L_{jk}$, $b^R_{ik} = b^R_{ij} b^L_{jk}$, $h^R_{ik} = h^R_{ij} \cup h^L_{jk}$
Constraint	$\frac{[i,j,\sigma^R_{ij}, A \rightarrow \alpha \bullet \Phi^A]}{[i,j,\sigma^L_{ij}, A \rightarrow \alpha\Phi^A \bullet]}$ $\langle \Phi^A$ is satisfiable $\rangle$	$\sigma^L_{ij} = \phi(\sigma^R_{ij})$

Table 1: LWFG parsing as deductive system

terminal of a LWFG rule). The goal items are of the form $[i, j, \sigma^L_{ij}, A \rightarrow \alpha\Phi^A \bullet]$, where σ^L_{ij} is ground-derived from the rule $A \rightarrow \alpha \colon \Phi^A$.

Compared to the deductive system in (Shieber et al., 1995), the LWFG parser has the following characteristics: each item is augmented with a syntagma; the *Constraint rule* is a *new inference rule*, and the goal items are associated to *every nonterminal in the grammar*, not only to the start symbol (i.e., LWFG parser is a robust parser). The **Constraint** inference rule is the only one that obtains an inactive edge[3], from an active edge by executing the grammar constraint Φ^A (the $\bullet$ is shifted across the constraint). By applying the Constraint rule as the last inference rule we obtain the ground-derived syntagmas σ^L_{ij}. Thus, the goal items are obtained only after the Constraint rule is applied. During this inference rule we have that $\sigma^L_{ij} = \phi(\sigma^R_{ij})$, where ϕ is defined by: $w^L_{ij} = w^R_{ij}$, $b^L_{ij} = b^R_{ij}\nu_{ij}$, and $h^L_{ij} = \varphi(h^R_{ij})$. The substitution ν_{ij} and the function φ are implicitly contained in the grammar constraint $\Phi^A_c(h^L_{ij}, h^R_{ij})$ (see Section 5 for details)

Definition 3 (Robust parsing provability)**.** *Robust parsing provability corresponds to reaching the goal item:* $\vdash_{rp} A(\sigma^L_{ij})$ *iff* $[i, j, \sigma^L_{ij}, A \rightarrow \alpha\Phi^A \bullet]$.

Thus, we can notice that the ground syntagma derivation is equivalent to robust parsing provability, i.e., $A \stackrel{*G}{\Rightarrow} \sigma$ iff $G \vdash_{rp} A(\sigma)$.

[3]We use Kay's terminology: items are edges, where the axioms and goals are inactive edges having $\bullet$ at the end, while the rest are active edges (Kay, 1986).

4 Learning LWFGs

The theoretical learning model for LWFG induction, Grammar Approximation by Representative Sublanguage (GARS), together with a learnability theorem was introduced in (Muresan and Rambow, 2007). LWFG's learning framework characterizes the "importance" of substructures in the model not simply by frequency, but rather linguistically, by defining a notion of "representative examples" that drives the acquisition process. Informally, representative examples are "building blocks" from which larger structures can be inferred via reference to a larger generalization corpus referred to as representative sublanguage in (Muresan and Rambow, 2007). The GARS model uses a polynomial algorithm for LWFG learning that take advantage of the building blocks nature of representative examples.

The LWFG induction algorithm belongs to the class of Inductive Logic Programming methods (ILP), based on entailment (Muggleton, 1995; Dzeroski, 2007). At each step a new constraint grammar rule is learned from the current representative example, σ. Then this rule is added to the grammar rule set. The process continues until all the representative examples are covered. We describe below the process of learning a grammar rule from the current representative example:

1. **Most Specific Grammar Rule Generation**. In the first step, the most specific grammar rule is generated from the current representative example σ. The category annotated

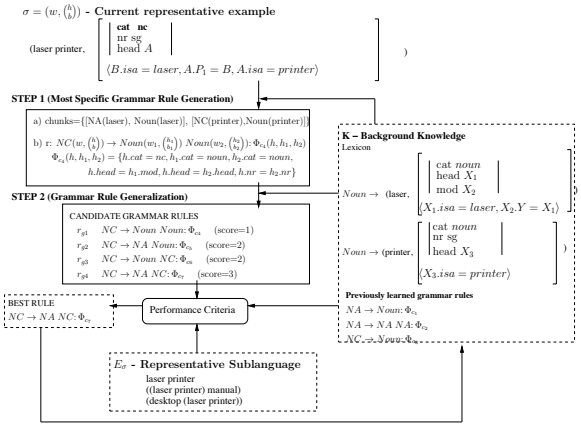


Figure 2: Example of Grammar Rule Learning

in the representative example gives the left-hand-side nonterminal, while a robust parser returns the minimum number of chunks covering the representative example. The categories of the chunks give the nonterminals of the right-hand side of the most specific rule. For example, in Figure 2, given the representative example *laser printer* annotated with its semantic molecule, and the background knowledge containing the already learned rules $NA \rightarrow Noun\colon \Phi_{c_1}$, $NA \rightarrow NA\ NA\colon \Phi_{c_2}$, $NC \rightarrow Noun\colon \Phi_{c_3}$ the robust parser generates the chunks corresponding to the noun *laser* and the noun *printer*: [NA(laser),Noun(laser)] and [NC(printer),Noun(printer)], respectively. The most specific rule is $NC \rightarrow Noun\ \ Noun\colon \Phi_{c_4}$, where the left-hand side nonterminal is given by the category of the representative example, in this case nc. Compositional constraints Φ_{c_4} are learned as well. In section 5 we give the algorithm for learning these constraints, and several properties and principles that are needed in order for these constraints to be learnable.

2. **Grammar Rule Generalization**. In the second step, this most specific rule is generalized, obtaining a set of candidate grammar rules (the generalization step is the inverse of the derivation step used to define the complete grammar lattice search space in (Muresan and Rambow, 2007)). The performance criterion in choosing the best grammar rule among these candidate hypotheses is the number of examples in the representative sublanguage E_σ (generalization corpus) that can be parsed using the candidate grammar rule, r_{gi} in the last ground derivation step, together with the previous learned rules, i.e., $|E_\sigma \cap L_\sigma(r_{gi})|$. In Figure 2 given the representative sublanguage E_σ={ *laser printer, laser printer manual, desktop laser printer*} the learner will generalize to the recursive rule $NC \rightarrow NA\ NC\colon \Phi_7$, since only this rule can parse all the examples in E_σ.

5 Learnable Composition Constraints

In LWFG, the semantic structures are composed by constraint solving, rather than functional application (with lambda expressions and lambda reduction). This section presents the properties and principles that guarantee the learnability of the compositional constraints,Φ_c, and presents an algorithm to generate these constraints from examples, which is a key result for LWFG learnability.

The information for semantic composition is encoded in the head of semantic molecules. There are three types of attributes that belong to the semantic molecule head h: category attributes $\mathcal{A}_h^c$, variable attributes $\mathcal{A}_h^v$, and feature attributes $\mathcal{A}_h^f$. Thus, $\mathcal{A}_h = \mathcal{A}_h^c \cup \mathcal{A}_h^v \cup \mathcal{A}_h^f$ and $\mathcal{A}_h^c, \mathcal{A}_h^v, \mathcal{A}_h^f$ are pairwise disjoint. For example, in Figure 1 for the noun-noun compound *laser printer*, we have that $\mathcal{A}_h^c = \{cat\}$, $\mathcal{A}_h^f = \{nr\}$, and $\mathcal{A}_h^v = \{head\}$, while for the noun *laser* we have that $\mathcal{A}_{h_1}^c = \{cat\}$, $\mathcal{A}_{h_1}^f = \emptyset$, and $\mathcal{A}_{h_1}^v = \{head, mod\}$ (nouns can be modifiers of other nouns, so their representation is similar to that of an adjective).

We describe in turn each of these types of attributes and their corresponding principles. All principles, except the first and the last mirror principles in other constraint-based linguistic formalisms, such as HPSG (Pollard and Sag, 1994).

The category attributes $\mathcal{A}_h^c$ are state attributes, and their value set gives the category of the semantic molecule. There is one attribute, cat $\in \mathcal{A}_h^c$, which is mandatory and whose value is the name of the category (e.g., $h.cat = nc$ in Figure

1). The category of a semantic molecule can be given by: 1) the cat attribute alone, or 2) the cat attribute together with other state attributes in $\mathcal{A}_h^c$ which are syntactic-semantic markers.

Principle 1 (Category Name Principle)**.** *The category name $h.cat$ of a syntagma $\sigma = (w, \binom{h}{b})$ is the same as the grammar nonterminal augmented with syntagma σ.*

When learning a LWFG rule from an example σ, the above principle allows us to determine the nonterminal in the left-hand side of the grammar rule. For example, when learning the LWFG rule from the syntagma corresponding to *laser printer* in Figure 2, the nonterminal in the left-hand side of the LWFG rule is NC since $h.cat = nc$.

The variable attributes $\mathcal{A}_h^v$ are attributes whose values are logical variables and represent the semantic valence of the molecule, which allows the binding of the semantic representations. These logical variables appear in the semantic molecule body as well. For example, in Figure 1(2) for the noun-noun compound *laser printer*, the value of the variable attribute $head \in \mathcal{A}_h^v$ is a variable X, which appears also in the body of the semantic molecule $\langle X_1.isa = laser, X.P_1 = X_1, X.isa = printer\rangle$. It can be noticed that the semantic molecule body contains other variables as well (X_1, P_1). However, only the variables present in the semantic molecule head as well (X) will participate in further composition.

Principle 2 (Semantic Representation Binding Principle)**.** *All the logical variables that the body b of a semantic molecule corresponding to a syntagma $\sigma = (w, \binom{h}{b})$, share with other syntagmas, are at the same time values of the variable attributes ($\mathcal{A}_h^v$) of the semantic molecule head.*

There is one variable attribute, $head \in \mathcal{A}_h^v$ that represents the head of a syntagma, giving the following principle:

Principle 3 (Semantic Head Principle)**.** *Given a syntagma $\sigma = (w, \binom{h}{b})$ ground derived from a grammar rule, r, there exists one and only one syntagma $\sigma_i = (w_i, \binom{h_i}{b_i})$ corresponding to a nonterminal B_i in rule r's right-hand side, which has the same value of the attribute $head$, i.e., $h.head = h_i.head$.*

The feature attributes $\mathcal{A}_h^f$ are the attributes whose values express the specific properties of the semantic molecules (e.g., number, person).

Principle 4 (Feature Inheritance Principle)**.** *If $\sigma_i = (w_i, \binom{h_i}{b_i})$ is the semantic head of a ground-derived syntagma $\sigma = (w, \binom{h}{b})$, then all feature attributes of σ inherit the values of the corresponding attributes that belong to the semantic head σ_i. That is, if $h.head = h_i.head$, then $h.f = h_i.f$, $\forall f \in \mathcal{A}_h^f \cap A_{h_i}^f$.*

Besides this principle, the feature attributes are used for category agreement. The categories that enter in agreement are maximum projection categories. This linguistic knowledge about agreement is used in the form of the following principle:

Principle 5 (Feature Agreement Principle)**.** *The agreeing categories and the agreement features are a-priori given based on linguistic knowledge, and are applied only at the semantic head level.*

Given all the above principles, we can now formulate the general Composition Principle:

Principle 6 (Composition Principle)**.** *A syntagma $\sigma = (w, w')$ corresponding to the left-hand side nonterminal of a grammar rule is obtained by string concatenation ($w = w_1 \dots w_n$) and the composition of semantic molecules corresponding to the nonterminals from the rule right-hand side:*

$$w' = \binom{h}{b} = (w_1 \cdots w_n)' = w_1' \circ \cdots \circ w_n'$$
$$= \binom{h_1}{b_1} \circ \cdots \circ \binom{h_n}{b_n} = \binom{h_1 \circ \cdots \circ h_n}{\langle b_1, \dots, b_n\rangle\nu}$$

The composition of the semantic molecule bodies is realized through conjunction after the application of a variable substitution ν. The body variable specialization substitution ν is the most general unifier (mgu) of b and $b_1, \dots, b_n$, s.t $b = (b_1, \dots, b_n)\nu$. It is a particular form of the commonly used substitution (Lloyd, 2003), i.e., a finite set of the form $\{X_1/Y_1, \dots, X_m/Y_m\}$, where $X_1, \dots, X_m, Y_1, \dots, Y_m$ are variables, and $X_1, \dots, X_m$ are distinct.

The composition of the semantic molecule heads is realized by a set of constraints $\Phi_c(h, h_1 ..., h_n)$, which is a system of equations

similar to "path equations" (Shieber et al., 1983; van Noord, 1993), but applied to flat feature structures:

$$\left\{\begin{array}{l} h_i.c = ct \\ h_i.v_i = h_j.v_j \\ h_i.f = ct\ or \\ h_i.f = h_j.f \end{array}\right\} \text{ where } \begin{array}{l} 0 \le i,j \le n, i \ne j \\ c \in \mathcal{A}^c_{h_i} \\ v_i \in \mathcal{A}^v_{h_i}, v_j \in \mathcal{A}^v_{h_j} \\ f \in \mathcal{A}^f_{h_i}, f \in \mathcal{A}^f_{h_j} \end{array}$$

When learning a LWFG rule from a representative example σ as in Figure 2, the robust parser returns the minimum number of chunks, n, covering σ. The body variable substitution ν is fully determined by the representative example as mgu of b and $b_1, \ldots, b_n$, and the compositional constraints $\Phi_c(h, h_1, \ldots, h_n)$ are learned using Alg 1. For example, in Figure 2, when learning from the representative example corresponding to the string *laser printer*, we have that $\nu = \{X_1/B, X_2/A, X_3/A, Y/P_1\}$.

In Alg 1 we use the notation $\sigma_0 = (w_0, \binom{h_0}{b_0})$ to denote the representative example σ.

Alg 1: Learn_Constraints($\sigma_0, \sigma_1, \ldots, \sigma_n$)

$\sigma_i = (w_i, \binom{h_i}{b_i}), 0 \le i \le n$
$\Phi_c \leftarrow \emptyset$
$\nu \leftarrow mgu(b_0, (b_1, \ldots, b_n))$
1 **foreach** $0 \le i \le n \wedge c \in \mathcal{A}^c_{h_i}$ **do**
  if $h_i.c = c1$ **then**
    $\Phi_c \leftarrow \Phi_c \cup \{h_i.c = c1\}$
2 **foreach** $0 \le i,j \le n \wedge i \ne j \wedge X/Y \in \nu \wedge$ $v_i \in \mathcal{A}^v_{h_i} \wedge v_j \in \mathcal{A}^v_{h_j}$ **do**
  if $h_i.v_i = X \wedge h_j.v_j = Y$ **then**
    $\Phi_c \leftarrow \Phi_c \cup \{h_i.v_i = h_j.v_j\}$
3 **if** $h_s.head = h_0.head, 1 \le s \le n$ **then**
  foreach $f \in \mathcal{A}^f_{h_0} \cap \mathcal{A}^f_{h_s}$ **do**
    if $h_0.f = c1 \wedge h_s.f = c1$ **then**
      $\Phi_c \leftarrow \Phi_c \cup \{h_0.f = h_s.f\}$
  if $h_s.cat = c_s \wedge h_i.cat = c_i \wedge agr(c_s, c_i)$, $1 \le i \le n$ **then**
    foreach $f \in agrFeatures(c_s, c_i)$ **do**
      if $h_s.f = c1 \wedge h_i.f = c1$ **then**
        $\Phi_c \leftarrow \Phi_c \cup \{h_s.f = h_i.f\}$
4 **for** *all other* $f \in \mathcal{A}^f_{h_i}, 0 \le i \le n$ **do**
  /*i.e., if we are not in case 3 */
  if $h_i.f = c1$ **then**
    $\Phi_c \leftarrow \Phi_c \cup \{h_i.f = c1\}$
return Φ_c /*i.e., $\Phi_c(h_0, h_1, \ldots, h_n)$ */

In the first step, the constraints corresponding to category attributes are fully determined by the values of these attributes that appear in the semantic molecule heads of $\sigma_0, \ldots \sigma_n$. In Figure 2, when learning the most specific rule r from the representative example *laser printer*, the set of constraints $\{h.cat = nc, h_1.cat = noun, h_2 = noun\} \subset \Phi_{c_4}$ are the constraints corresponding to category attributes. In the second step, the constraints corresponding to variable attributes are fully determined by the variables in the substitution ν that also appear as values of variable attributes $h_i.v_i, h_j.v_j$, where $0 \le i,j \le n$ and $i \ne j$. In Figure 2, only $\{X_2/A, X_3/A\} \subset \nu$ will be used, generating the set of constraints $\{h.head = h_1.mod, h.head = h_2.head\} \subset \Phi_{c_4}$. In the third step, the values of the feature attributes which obey Principles 4 and 5 are generalized — agr(c_s, c_i) is the predicate which gives us the agreement between the categories c_s and c_i (e.g., the subject agrees with the verb), and agrFeatures(c_s, c_i) gives us the set of feature attributes that participate in agreement (e.g., nr, pers, case). In Figure 2, the set of constraints $\{h.nr = h_2.nr\} \subset \Phi_{c_4}$ represents the generalization of the feature attribute values for nr, using Principle 4 . For all features attributes besides the ones that obey the above two principles, the generated constraints keep the particular values of these attributes (step 4 of Alg 1).

6 Examples

The LWFG formalism allows us to learn grammars for deep language understanding from examples. Instead of writing syntactic-semantic grammar by hand (both rules and constraints), we need to provide only a small set of representative examples — strings and their semantic molecules. Qualitative experiments on learning LWFGs showed that complex linguistic constructions can be learned and covered, such as complex noun phrases, relative clauses and reduced relative clauses, finite and non-finite verbal constructions (including, tense, aspect, negation, and subject-verb agreement), and raising and control constructions (Muresan and Rambow, 2007). In Figure 3 we show an example of learning a LWFG grammar for noun-noun compounds. The first four examples (1-4) are representative examples, while the last four examples are used for gener-

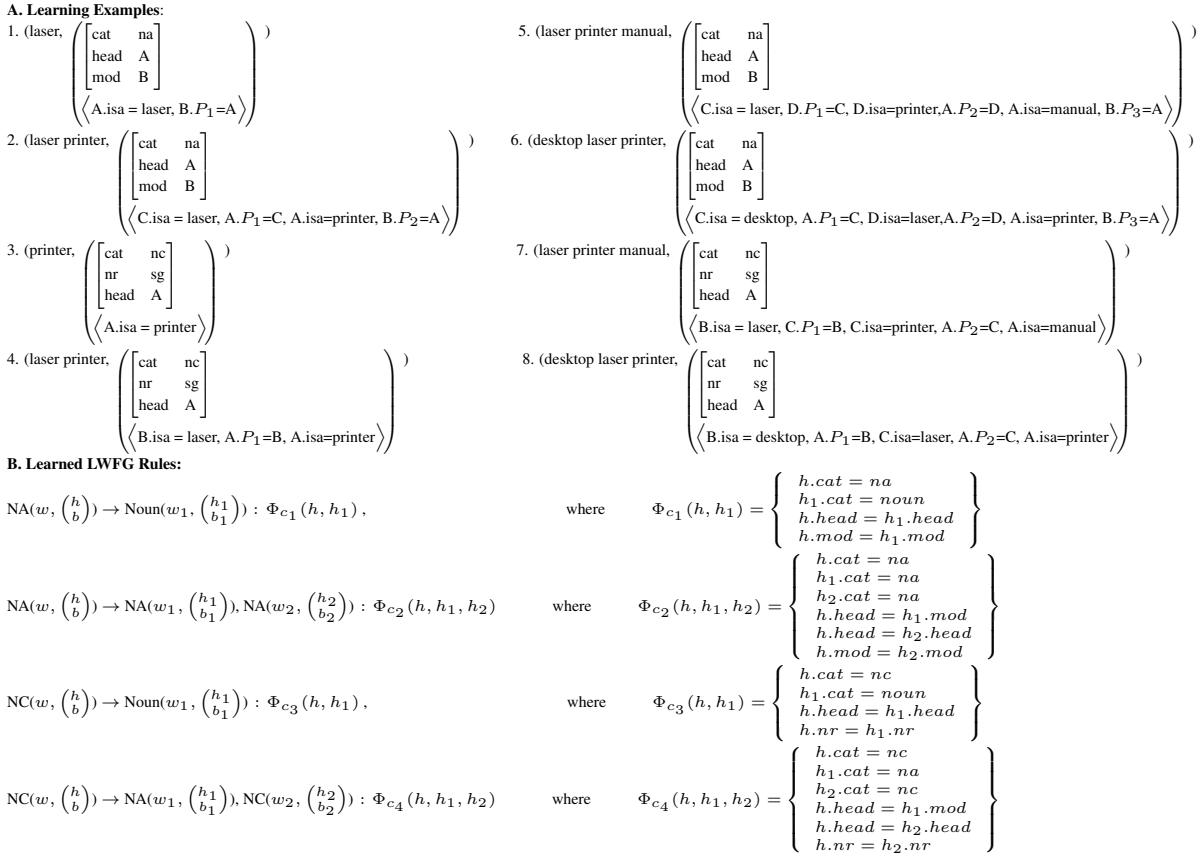


Figure 3: Learning LWFG Rules for Noun-Noun Compounds

alization (5-8). The learned grammar rules, including the learned composition constraints are also shown. The first two LWFG rules ground derive syntagmas for noun adjuncts, while the last two rules ground derive syntagmas for noun compounds. For example, "desktop laser printer" can be either a fully-formed noun compound (category nc), or it can be further combined with the noun "invoice" to obtain "desktop laser printer invoice", case in which it is a noun adjunct (category na). The learned rule for noun adjuncts is both left and right recursive, accounting for both left and right-branching noun compounds. Even though we can obtain overgeneralization in syntax, the ontology-based interpretation constraint at the rule level will prune some erroneous parses. Preliminary results in the medical domain show that Φ_{onto} can help remove erroneous parses even when using just a weak ontological model (semantic roles of verbs, prepositions, attributes of adjectives and adverbs, but no synonymy, or hierarchy of concepts or roles). However, more experiments need to be run for reporting quantitative results.

7 Conclusions

We have presented the properties and principles that the semantic representation integrated in LWFG requires so that the semantic compositional constraints are learnable from examples. These properties together with Alg 1 give a theoretical result that in conjunction with the learnability result of Muresan and Rambow (2007) show that LWFG is a learnable constraint-based grammar formalism that can be used for deep language understanding. Instead of writing grammar rules and constraints by hand, one needs to provide only a small set of annotated examples.[4]

[4]The author acknowledges the support of the NSF (SGER grant IIS-0838801). Any opinions, findings, or conclusions are those of the author, and do not necessarily reflect the views of the funding organization.

References

Dzeroski, Saso. 2007. Inductive logic programming in a nutshell. In Getoor, Lise and Ben Taskar, editors, *Introduction to Statistical Relational Learning*. The MIT Press.

Ge, Ruifang and Raymond J. Mooney. 2005. A statistical semantic parser that integrates syntax and semantics. In *Proceedings of CoNLL-2005*.

Kay, M. 1986. Algorithm schemata and data structures in syntactic processing. In *Readings in natural language processing*, pages 35–70. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Lloyd, John W. 2003. *Logic for Learning: Learning Comprehensible Theories from Structured Data*. Springer, Cognitive Technologies Series.

Muggleton, Stephen. 1995. Inverse Entailment and Progol. *New Generation Computing, Special Issue on Inductive Logic Programming*, 13(3-4):245–286.

Muresan, Smaranda and Owen Rambow. 2007. Grammar approximation by representative sublanguage: A new model for language learning. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Muresan, Smaranda. 2008. Learning to map text to graph-based meaning representations via grammar induction. In *Coling 2008: Proceedings of the 3rd Textgraphs workshop on Graph-based Algorithms for Natural Language Processing*, pages 9–16, Manchester, UK, August. Coling 2008 Organizing Committee.

Neumann, Günter and Gertjan van Noord. 1994. Reversibility and self-monitoring in natural language generation. In Strzalkowski, Tomek, editor, *Reversible Grammar in Natural Language Processing*, pages 59–96. Kluwer Academic Publishers, Boston.

Pollard, Carl and Ivan Sag. 1994. *Head-Driven Phrase Structure Grammar*. University of Chicago Press, Chicago, Illinois.

Shieber, Stuart, Hans Uszkoreit, Fernando Pereira, Jane Robinson, and Mabry Tyson. 1983. The formalism and implementation of PATR-II. In Grosz, Barbara J. and Mark Stickel, editors, *Research on Interactive Acquisition and Use of Knowledge*, pages 39–79. SRI International, Menlo Park, CA, November.

Shieber, Stuart, Yves Schabes, and Fernando Pereira. 1995. Principles and implementation of deductive parsing. *Journal of Logic Programming*, 24(1-2):3–36.

Steedman, Mark. 1996. *Surface Structure and Interpretation*. The MIT Press.

van Noord, Gertjan. 1993. *Reversibility in Natural Language Processing*. Ph.D. thesis, University of Utrecht.

Wong, Yuk Wah and Raymond Mooney. 2007. Learning synchronous grammars for semantic parsing with lambda calculus. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL-2007)*.

Zettlemoyer, Luke S. and Michael Collins. 2005. Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. In *Proceedings of UAI-05*.

Zettlemoyer, Luke and Michael Collins. 2009. Learning context-dependent mappings from sentences to logical form. In *Proceedings of the Association for Computational Linguistics (ACL'09)*.

Evaluating performance of grammatical error detection to maximize learning effect

Ryo Nagata
Konan University
rnagata @ konan-u.ac.jp.

Kazuhide Nakatani
Konan University

Abstract

This paper proposes a method for evaluating grammatical error detection methods to maximize the learning effect obtained by grammatical error detection. To achieve this, this paper sets out the following two hypotheses — imperfect, rather than perfect, error detection maximizes learning effect; and precision-oriented error detection is better than a recall-oriented one in terms of learning effect. Experiments reveal that (i) precision-oriented error detection has a learning effect comparable to that of feedback by a human tutor, although the first hypothesis is not supported; (ii) precision-oriented error detection is better than recall-oriented in terms of learning effect; (iii) F-measure is not always the best way of evaluating error detection methods.

1 Introduction

To reduce the efforts taken to correct grammatical errors in English writing, there has been a great deal of work on grammatical error detection (Brockett et al., 2006; Chodorow and Leacock, 2000; Chodorow and Leacock, 2002; Han et al., 2004; Han et al., 2006; Izumi et al., 2003; Nagata et al., 2004; Nagata et al., 2005; Nagata et al., 2006). One of its promising applications is writing learning assistance by detecting errors and showing the results to the learner as feedback that he or she can use to rewrite his or her essay. Grammatical error detection has greatly improved in detection performance as well as in the types of the errors it is able to detect, including errors in articles, number, prepositions, and agreement.

In view of writing learning assistance, however, one important factor has been missing in the previous work. In the application to writing learning assistance, error detection methods should be evaluated by learning effect obtained by error detection. Nevertheless, they have been evaluated only by detection performance such as F-measure.

This brings up a further research question — are any of the previous methods effective as writing learning assistance? It is very important to answer this question because it is almost impossible to develop a perfect method. In other words, one has to use an imperfect method to assist learners no matter how much improvement is achieved. In practice, it is crucial to reveal the lower bound of detection performance that has a learning effect.

Related to this, one should discuss the following question. Most error detection methods are adjustable to be recall-oriented/precision-oriented by tuning their parameters. Despite this fact, no one has examined which is better in terms of learning effect — recall-oriented or precision-oriented? (hereafter, this problem will be referred to as the *recall-precision problem*). Chodorow and Leacock (2000) and Chodorow et al. (2007) argue that precision-oriented is better, but they do not give any concrete reason. This means that the recall-precision problem has not yet been solved.

Accordingly, this paper explores the relation between detection performance and learning effect. To do this, this paper sets out two hypotheses:

Hypothesis I : imperfect, rather than perfect, error detection maximizes learning effect

Hypothesis II : precision-oriented is better than recall-oriented in terms of learning effect

Hypothesis I contradicts the intuition that the better the detection performance is, the higher the learning effect is. To see the motivation for this,

Coling 2008: Poster Volume, pages 894–900,
Beijing, August 2010

suppose that we had a perfect method. It would detect all errors in a given essay with no false-positives. In that case, the learner would not have to find any errors by himself or herself. Neither would he or she have to examine the causes of the errors. In the worst case, they just copy the detection results. By contrast, with an imperfect method, he or she has to do these activities, which is expected to result in better learning effect. Besides, researchers, including Robb et al. (1986), Bitchener et al. (2005), and Ferris and Roberts (2001), report that the amount of feedback that learners receive does not necessarily correspond to the amount of learning effect. For instance, Robb et al. (1986) compared four types of feedback ((1) error detection and correction, (2) error detection and error type, (3) error detection, and (4) number of errors per line) and reported that (1), the most-detailed feedback, did not necessarily have the highest learning effect.

Hypothesis II concerns the recall-precision problem. If a limited number of errors are detected with high precision (i.e., precision-oriented), learners have to carefully read their own essay to find the rest of the errors by examining whether their writing is correct or not, using several sources of information including (i) the information that can be obtained from the detected errors, which is useful for finding undetected errors similar to the detected ones; (ii) their knowledge on English grammar and writing, and (iii) dictionaries and textbooks. We believe that learning activities, especially learning from similar instances, have a favorable learning effect. By contrast, in a recall-oriented setting, these activities relatively decrease. Instead, learners focus on judging whether given detection results are correct or not. Besides, learning from similar instances is likely not to work well because a recall-oriented setting frequently makes false-positives.

This paper proposes a method for testing the two hypotheses in Sect. 2. It conducts experiments based on the method in Sect. 3. It discusses the experimental results in Sect. 4.

2 Method

We conducted a pre-experiment where ten subjects participated and wrote 5.6 essays on average. We used the obtained data to design the method.

2.1 Target Errors

To obtain general conclusions, one has to test **Hypothesis I** and **Hypothesis II** against a variety of errors and also a variety of error detection methods. However, it would not be reasonable or feasible to do this from the beginning.

Considering this, this paper targets errors in articles and number. The reasons for selecting these are that (a) articles and number are difficult for learners of English (Izumi et al., 2003; Nagata et al., 2005), and (b) there has been a great deal of work on the detection of these errors.

2.2 Error detection method

Among the previous methods for detecting errors in articles and number, this paper selects Nagata et al. (2006)'s method that detects errors in articles and number based on countability prediction. It has been shown to be effective in the detection of errors in articles and number (Nagata et al., 2005; Nagata et al., 2006). It also has the favorable property that it can be adjusted to be recall-oriented or precision-oriented by setting a threshold for the probability used in countability prediction. This subsection briefly describes Nagata et al. (2006)'s method (See Nagata et al. (2006) for the details).

The method, first, automatically generates training instances for countability prediction. Instances of each noun that head their noun phrase (NP) are collected from a corpus with their surrounding words. Then, the collected instances are tagged with their countability by a set of hand-coded rules. The resulting tagged instances are used as training data for countability prediction.

Decision lists (Yarowsky, 1995) are used to predict countability. Tree types of contextual cue are used as features: (i) words in the NP that the target noun heads; (ii) three words to the left of the NP; (iii) three words to its right. The log-likelihood ratio (Yarowsky, 1995) decides in which order rules in a decision list are applied to the target noun in countability prediction. It is the log ratio of the probabilities of the target noun being count and non-count when one of the features appears in its context. To predict countability in error detection, each rule in the decision list is tested on the target

noun in the sorted order until the first applicable one is found. The prediction is made by the first applicable one.

After countability prediction, errors in articles and number are detected by using a set of rules. For example, if the noun in question is plural and predicted to be non-count, then it is an error. Similarly, the noun in question has no article and is singular and is predicted to be count, then it is an error.

The balance of recall and precision in error detection can be adjusted by setting a certain threshold to the probabilities used to calculate the log-likelihood ratio[1]. If the probability of the applied rule in countability prediction is lower than a certain threshed, error detection is blocked. Namely, the higher the threshed is, the more precision-oriented the detection is.

2.3 Learning Activity

The proposed method is based on a learning activity consisting of essay writing, error detection, and rewriting. Table 1 shows the flow of the learning activity. In Step 1, an essay topic is assigned to learners. In Step 2, they have time to think about what to write with a piece of white paper for preparation (e.g., to summarize his or her ideas). In Step 3, they write an essay on a blog system in which the error detection method (Nagata et al., 2005) is implemented. This system allows them to write, submit, and rewrite their essays (though it does not allow them to access the others' essays or their own previous essays). They are not allowed to use any dictionary or textbook in this step. They are required to write ten sentences or more. In Step 4, the system detects errors in each essay. It displays each essay of which errors are indicated in red to the corresponding learner. Although the detection itself takes only a few seconds, five minutes are assigned to this step for two purposes: to take a short break for learners and to remove time differences between learners. Finally, in Step 5, learners rewrite their essay using the given feedback. Here, they are allowed to use a dictionary (Konishi and Minamide, 2007) and an A4 paper that briefly explains article and number usage, which was made based on grammar books (Hirota, 1992; Iizuka and Hagino, 1997). They are informed that the feedback may contain false-positives and false-negatives.

Table 1: Flow of learning activity

Procedure	Min
1. Learner is assigned an essay topic	–
2. Learner prepares for writing	5
3. Learner writes an essay	35
4. System detects errors in the essay	5
5. Learner rewrites the essay	15

2.4 How to Measure Learning Effect

Before discussing how to measure learning effect, one has to define the ability to write English. Considering that this paper aims at the evaluation of error detection, it is reasonable to define the ability as the degree of error occurrence (that is, the fewer errors, the better). To measure this, this paper uses error rate, which is defined by

$$e = \frac{\text{Number of target errors in Step 3} + 1}{\text{Number of NPs in Step 3} + 1}. \quad (1)$$

Ones ("+1") are added to the numerator and denominator for a mathematical reason that will be clear shortly. The addition also has the advantage that it can evaluate a longer essay to be better when no errors occur.

Having defined ability, it is natural to measure learning effect by a decrease in the error rate. Simply, it is estimated by applying the linear regression to the number of instances of learning and the corresponding error rates.

Having said this, this paper applies an exponential regression instead of the linear regression. There are two reasons for this. The first is that it becomes more difficult to decrease the error rate as it decreases (in other words, it becomes more difficulty to improve one's ability as one improves). The other is that the error rate is expected to asymptotically decrease to zero as learning proceeds. The exponential regression is defined by

$$e = \exp\{a(t + b)\} \quad (2)$$

[1] Setting a threshold to the probability is equivalent to setting a threshold to the log-likelihood and both has the same effect on the balance of recall and precision. However, we use the former because it is intuitive and easy to set a threshold

where t, a, and b denote the number of instances of learning, decrease in the error rate (learning effect), and the ability before the learning starts, respectively. The parameters a and b can be estimated from experimental data by least squares.

To examine **Hypothesis I** and **Hypothesis II**, the learning effect parameter a must be estimated for several error detection conditions. To do this, detection performance (recall, precision, and F-measure) is first defined. Recall and precision is defined by

$$r = \frac{\text{Number of errors correctly detected}}{\text{Number of errors}} \quad (3)$$

and

$$p = \frac{\text{Number of errors correctly detected}}{\text{Number of errors detected}}, \quad (4)$$

respectively. Using recall and precision, F-measure is defined by

$$f = \frac{2rp}{r+p}. \quad (5)$$

With these, this paper compares four conditions. In the first condition, the system detects no error at all. Thus, it plays a role as a baseline. The second and third conditions are recall-oriented and precision-oriented, respectively. The threshold that maximized F-measure, which was 0.60, was computed by applying the error detection method to the essays obtained in the pre-experiment (increasing the threshold from 0 to 1, 0.05 at a time). This was selected as the recall-oriented condition. Then, the threshold for the precision-oriented condition was determined to be 0.90 so that its precision became higher. The final condition corresponds to the perfect error detection. Because it was impossible to implement such error detection, a native speaker of English took this part. Hereafter, the four conditions will be referred to as **No-feedback**, **Recall-oriented**, **Precision-oriented**, and **Human**.

3 Experiments

As subjects, 26 Japanese college students (first to fourth grade) participated in the experiments. These 26 subjects were assigned to each condition as follows: **Human**: 6; **Recall-oriented**: 7; **Precision-oriented**: 7; **No-feedback** 6:.

Table 2: Essay topics used in the experiments

No.	Topic
1	University life
2	Summer vacation
3	Gardening
4	My hobby
5	My frightening experience
6	Reading
7	My home town
8	Traveling
9	My favorite thing
10	Cooking

The number of learning activities was ten. Essay Topics for each learning activity is shown in Table 2 They were selected based on a writing textbook (Okihara, 1985). The experiments were conducted from Oct. 2008 to Dec. 2008. The subjects basically did the learning activity twice a week on average. Some of them could not finish the ten-essays assignment during this term. Subjects who did not do the learning activity eight or more times were excluded from the experiments. As a result, 22 subjects were valid in the end (**Human**: 4; **Recall-oriented**: 7; **Precision-oriented**: 6; **No-feedback**: 5).

Figure 1 shows the experimental results. It shows the plots of Eq. (2) where a is calculated by averaging the estimated values of a over each condition (**No-feedback**: $a = -0.024$; **Recall-oriented**: $a = -0.015$; **Precision-oriented**: $a = -0.038$; **Human**: $a = -0.046$). The value of b is set to 0 for the purpose of comparison.

4 Discussion

Although **Hypothesis I** is not supported, the experimental results reveal that **Precision-oriented** has a learning effect comparable to **Human**. A concrete example makes this clearer. **Precision-oriented** takes 18 instances of learning to decrease the error rate 32%, which is the average of the subjects at the beginning, by half. This is very near the 16 instances of **Human**. By contrast, **No-feedback** takes nearly double that (29 times), and **Recall-oriented** far more (47 times).

From these results, it follows that one should

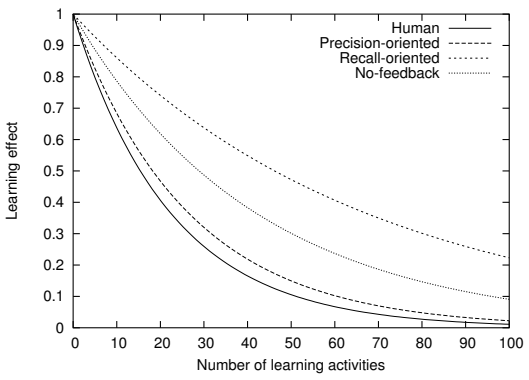


Figure 1: Experimental results

use precision-oriented error detection for writing learning assistance in a circumstance where feedback by a human tutor is not fully available (e.g., writing classroom consisting of a number of students). According to Burstein et al. (1998), the best way to improve one's writing skills is (i) to write, (ii) to receive feedback from a human tutor, (iii) to revise based on the feedback, and then repeat the whole process as often as possible. However, it is almost impossible to practice this in a writing classroom consisting of a number of students. In such circumstances, this can be done by using precision-oriented error detection. At the end, learners may have their essays corrected by a human tutor, which guarantees the quality of feedback, still reducing the efforts of human tutors.

At the same time, it should be emphasized that this is not a general but a limited conclusion because the experiments involve limited target errors and a limited number of subjects. In different conditions (e.g., setting a higher threshold), **Precision-oriented** may outperform **Human**, meaning that **Hypothesis I** is not conclusively rejected.

The experimental results support **Hypothesis II** as we expected. The learning effect of **Recall-oriented** is even less than **No-feedback**. A possible reason for this is that false-positives, which **Recall-oriented** frequently makes, confused the subjects. By contrast, **Precision-oriented** achieved better learning effect because it detected a few errors with a high precision. To be precise, **Recall-oriented** achieved a precision of 0.60 with a recall 0.31 whereas a precision of 0.72 with a recall of 0.25 in **Precision-oriented**. Besides, the fact that **Recall-oriented** detects errors more frequently with less precision (that is, the number of false-positives is higher) might make learners feel as if the precision is lower than is actually. This might have discouraged the subjects in **Recall-oriented** from learning.

These results suggest interesting findings from another point of view. In the past, overall performance of error detection has often been evaluated by F-measure, which considers both recall and precision. Following this convention, one comes to the conclusion that **Recall-oriented** ($F = 0.41$) is superior to **Precision-oriented** ($F = 0.37$). Contrary to this, the experimental results favor **Precision-oriented** over **Recall-oriented** in terms of learning effect. This suggest that F-measure is not always the best method of evaluation.

To conclude this section, let us discuss some problems with the proposed method that the experiments have revealed. To obtain more general conclusions, the amount of experimental data should be increased. However, it appeared to be difficult for the subjects to do the learning activity more than ten times; some subjects might have got bored with repeating the same learning activities. This is the problem that has to be solved in its actual use in learning assistance. Another problem is that detection performance tends to decrease relative to the original as learning proceeds because subjects improve (for instance, $F = 0.44$ for the first half and $F = 0.38$ for the last half in **Recall-oriented**). In order to investigate the relation between detection performance and learning effect more deeply, one should take this fact into consideration.

5 Conclusions

This paper tested the two hypotheses — imperfect, rather than perfect, error detection maximizes learning effect; and precision-oriented error detection is better than a recall-oriented one in terms of learning effect. The experiments revealed the interesting findings that precision-oriented error detection has learning effect similar to that of

feedback by a human tutor, although the first hypothesis was not supported. Considering the findings, this paper has come to the conclusion that one should use precision-oriented error detection to assist writing learning in a circumstance where feedback by human tutors is not fully available. By contrast, the experiments supported the second hypothesis. They also showed that F-measure was not always the best way of evaluation.

In future work, we will expand the experiments in terms of both the number of subjects and target errors, such as errors in preposition, to obtain more general conclusions. The essays which are collected and error-annotated[2] in the experiments are available as a learner corpus for research and education purposes. Those who are interested in the learner corpus should contact the author.

References

Bitchener, John, Stuart Young, and Denise Cameron. 2005. The effect of different types of corrective feedback on ESL student writing. *Journal of Second Language Writing*, 14(3):191–205.

Brockett, Chris, William B. Dolan, and Michael Gamon. 2006. Correcting ESL errors using phrasal SMT techniques. In *Proc. of 21th International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 249–256, Sydney, Australia, July.

Burstein, Jill, Karen Kukich, Susanne Wolff, Chi Lu, Martin Chodorow, Lisa Braden-Harder, and Mary D. Harris. 1998. Automated scoring using a hybrid feature identification technique. In *Proc. of 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, pages 206–210.

Chodorow, Martin and Claudia Leacock. 2000. An unsupervised method for detecting grammatical errors. In *Proc. of 1st Meeting of the North America Chapter of ACL*, pages 140–147.

Chodorow, Martin and Claudia Leacock. 2002. Techniques for detecting syntactic errors in text. In *IEICE Technical Report (TL2002-39)*, pages 37–41.

Chodorow, Martin, Joel R. Tetreault, and Na-Rae Han. 2007. Detection of grammatical errors involving prepositions. In *Proc. of 4th ACL-SIGSEM Workshop on Prepositions*, pages 25–30.

Ferris, Dana and Barrie Roberts. 2001. Error feedback in L2 writing classes: How explicit does it need to be? *Journal of Second Language Writing*, 10(3):161–184.

Han, Na-Rae, Martin Chodorow, and Claudia Leacock. 2004. Detecting errors in English article usage with a maximum entropy classifier trained on a large, diverse corpus. In *Proc. of 4th International Conference on Language Resources and Evaluation*, pages 1625–1628.

Han, Na-Rae, Martin Chodorow, and Claudia Leacock. 2006. Detecting errors in English article usage by non-native speakers. *Natural Language Engineering*, 12(2):115–129.

Hirota, Shigeaki. 1992. *Mastery (in Japanese)*. Kirihara Shoten, Tokyo.

Iizuka, Shigeru and Satoshi Hagino. 1997. *Prestige*. Buneido, Tokyo.

Izumi, Emi, Kiyotaka Uchimoto, Toyomi Saiga, Thepchai Supnithi, and Hitoshi Isahara. 2003. Automatic error detection in the Japanese learners' English spoken data. In *Proc. of 41st Annual Meeting of ACL*, pages 145–148.

Konishi, Tomoshichi and Kosei Minamide. 2007. *Genious English-Japanese dictionary, 4th ed.* Taishukan, Tokyo.

Nagata, Ryo, Fumito Masui, Atsuo Kawai, and Naoki Isu. 2004. Recognizing article errors based on the three head words. In *Proc. of Cognition and Exploratory Learning in Digital Age*, pages 184–191.

Nagata, Ryo, Takahiro Wakana, Fumito Masui, Atsuo Kawai, and Naoki Isu. 2005. Detecting article errors based on the mass count distinction. In *Proc. of 2nd International Joint Conference on Natural Language Processing*, pages 815–826.

Nagata, Ryo, Astuo Kawai, Koichiro Morihiro, and Naoki Isu. 2006. A feedback-augmented method for detecting errors in the writing of learners of English. In *Proc. of 44th Annual Meeting of ACL*, pages 241–248.

Okihara, Katsuaki. 1985. *English writing (in Japanese)*. Taishukan, Tokyo.

Robb, Thomas, Steven Ross, and Ian Shortreed. 1986. Salience of feedback on error and its effect on EFL writing quality. *TESOL QUARTERY*, 20(1):83–93.

[2]Including not only errors in articles and number but also other types of error.

Yarowsky, David. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *Proc. of 33rd Annual Meeting of ACL*, pages 189–196.

Kernel-based Reranking for Named-Entity Extraction

Truc-Vien T. Nguyen and **Alessandro Moschitti** and **Giuseppe Riccardi**
Department of Information Engineering and Computer Science
University of Trento
nguyenthi,moschitti,riccardi@disi.unitn.it

Abstract

We present novel kernels based on structured and unstructured features for reranking the *N*-best hypotheses of conditional random fields (CRFs) applied to entity extraction. The former features are generated by a polynomial kernel encoding entity features whereas tree kernels are used to model dependencies amongst tagged candidate examples. The experiments on two standard corpora in two languages, i.e. the Italian EVALITA 2009 and the English CoNLL 2003 datasets, show a large improvement on CRFs in F-measure, i.e. from 80.34% to 84.33% and from 84.86% to 88.16%, respectively. Our analysis reveals that both kernels provide a comparable improvement over the CRFs baseline. Additionally, their combination improves CRFs much more than the sum of the individual contributions, suggesting an interesting kernel synergy.

1 Introduction

Reranking is a promising computational framework, which has drawn special attention in the Natural Language Processing (NLP) community. Basically, this method first employs a probabilistic model to generate a list of top-*n* candidates and then reranks this *n*-best list with additional features. One appeal of this approach is its flexibility of incorporating arbitrary features into a model. These features help in discriminating good from bad hypotheses and consequently their automatic learning. Various algorithms have been applied for reranking in NLP applications (Huang, 2008; Shen et al., 2004; Collins, 2002b; Collins and Koo, 2000), including parsing, name tagging and machine translation. This work has exploited the disciminative property as one of the key criterion of the reranking algorithm.

Reranking appears extremely interesting if coupled with kernel methods (Dinarelli et al., 2009; Moschitti, 2004; Collins and Duffy, 2001), as the latter allow for extracting from the ranking hypotheses a huge amount of features along with their dependencies. Indeed, while feature-based learning algorithms involve only the dot-product between feature vectors, kernel methods allow for a higher generalization by replacing the dot-product with a function between pairs of linguistic objects. Such functions are a kind of similarity measure satisfying certain properties. An example is the tree kernel (Collins and Duffy, 2001), where the objects are syntactic trees that encode grammatical derivations and the kernel function computes the number of common *subtrees*. Similarly, sequence kernels (Lodhi et al., 2002) count the number of common *subsequences* shared by two input strings.

Named-entities (NEs) are essential for defining the semantics of a document. NEs are objects that can be referred by names (Chinchor and Robinson, 1998), such as people, organizations, and locations. The research on NER has been promoted by the Message Understanding Conferences (MUCs, 1987-1998), the shared task of the Conference on Natural Language Learning (CoNLL, 2002-2003), and the Automatic Content Extraction program (ACE, 2002-2005). In the literature, there exist various learning approaches to extract named-entities from text. A NER sys-

Coling 2008: Poster Volume, pages 901–909,
Beijing, August 2010

tem often builds some generative/discriminative model, then, either uses only one classifier (Carreras et al., 2002) or combines many classifiers using some heuristics (Florian et al., 2003).

To the best of our knowledge, reranking has not been applied to NER except for the reranking algorithms defined in (Collins, 2002b; Collins, 2002a), which only targeted the entity detection (and not entity classification) task. Besides, since kernel methods offer a natural way to exploit linguistic properties, applying kernels for NE reranking is worthwhile.

In this paper, we describe how kernel methods can be applied for reranking, i.e. detection and classification of named-entities, in standard corpora for Italian and English. The key aspect of our reranking approach is how structured and flat features can be employed in discriminating candidate tagged sequences. For this purpose, we apply tree kernels to a tree structure encoding NE tags of a sentence and combined them with a polynomial kernel, which efficiently exploits global features.

Our main contribution is to show that (a) tree kernels can be used to define general features (not merely syntactic) and (b) using appropriate algorithms and features, reranking can be very effective for named-entity recognition. Our study demonstrates that the composite kernel is very effective for reranking named-entity sequences. Without the need of producing and heuristically combining learning models like previous work on NER, the composite kernel not only captures most of the flat features but also efficiently exploits structured features. More interestingly, this kernel yields significant improvement when applied to two corpora of two different languages. The evaluation in the Italian corpus shows that our method outperforms the best reported methods whereas on the English data it reaches the state-of-the-art.

2 Background

2.1 The data

Different languages exhibit different linguistic phenomena and challenges. A robust NER system is expected to be well-adapted to multiple domains and languages. Therefore, we experimented with two datasets: the EVALITA 2009 Italian corpus and the well-known CoNLL 2003 English shared task corpus.

The EVALITA 2009 Italian dataset is based on I-CAB, the Italian Content Annotation Bank (Magnini et al., 2006), annotated with four entity types: Person (PER), Organization (ORG), Geo-Political Entity (GPE) and Location (LOC). The training data, taken from the local newspaper "L'Adige", consists of 525 news stories which belong to five categories: News Stories, Cultural News, Economic News, Sports News and Local News. Test data, on the other hand, consist of completely new data, taken from the same newspaper and consists of 180 news stories.

The CoNLL 2003 English dataset is created within the shared task of CoNLL-2003 (Sang and Meulder, 2003). It is a collection of news wire articles from the Reuters Corpus, annotated with four entity types: Person (PER), Location (LOC), Organization (ORG) and Miscellaneous name (MISC). The training and the development datasets are news feeds from August 1996, while the test set contains news feeds from December 1996. Accordingly, the named entities in the test dataset are considerably different from those that appear in the training or the development set.

Italian	GPE	LOC	ORG	PER
Train	2813	362	3658	4577
	24.65%	3.17%	32.06%	40.11%
Test	1143	156	1289	2378
	23.02%	3.14%	25.96%	47.89%

English	LOC	MISC	ORG	PER
Train	7140	3438	6321	6600
	30.38%	14.63%	26.90%	28.09%
Dev	1837	922	1341	1842
	30.92%	15.52%	22.57%	31.00%
Test	1668	702	1661	1617
	29.53%	12.43%	29.41%	28.63%

Table 1: Statistics on the Italian EVALITA 2009 and English CoNLL 2003 corpora.

2.2 The baseline algorithm

We selected Conditional Random Fields (Lafferty et al., 2001) as the baseline model. Conditional

random fields (CRFs) are a probabilistic framework for labeling and segmenting sequence data. They present several advantages over other purely generative models such as Hidden Markov models (HMMs) by relaxing the independence assumptions required by HMMs. Besides, HMMs and other discriminative Markov models are prone to the label bias problem, which is effectively solved by CRFs.

The named-entity recognition (NER) task is framed as assigning label sequences to a set of observation sequences. We follow the IOB notation where the NE tags have the format B-TYPE, I-TYPE or O, which mean that the word is a beginning, a continuation of an entity, or not part of an entity at all. For example, consider the sentence with their corresponding NE tags, each word is labeled with a tag indicating its appropriate named-entity, resulting in annotated text, such as:

Il/O **presidente**/O **della**/O **Fifa**/B-ORG **Sepp**/B-PER **Blatter**/I-PER **affermando**/O **che**/O **il**/O **torneo**/O **era**/O **stato**/O **ottimo**/O (FIFA president Sepp Blatter says that the tournament was excellent)

For our experiments, we used CRF++ [1] to build our recognizer, which is a model trained discriminatively with the unigram and bigram features. These are extracted from a window at k words centered in the target word *w* (i.e. the one we want to classify with the B, O, I tags). More in detail such features are:

- The **word itself**, its **prefixes**, **suffixes**, and **part-of-speech**
- **Orthographic/Word features**. These are binary and mutually exclusive features that test whether a word contains *all upper-cased, initial letter upper-cased, all lower-cased, roman-number, dots, hyphens, acronym, lonely initial, punctuation mark, single-char, and functional-word*.
- **Gazetteer features**. Class (geographical, first name, surname, organization prefix, location prefix) of words in the window.
- **Left Predictions**. The predicted tags on the left of the word in the current classification.

[1] http://crfpp.sourceforge.net

The gazetteer lists are built with names imported from different sources. For English, the geographic features are imported from NIMA's GEOnet Names Server (GNS)[2], The Alexandria Digital Library (ADL) gazetteer[3]. The company data is included with all the publicly traded companies listed in Google directory[4], the European business directory[5]. For Italian, the generic proper nouns are extracted from Wikipedia and various Italian sites.

2.3 Support Vector Machines (SVMs)

Support Vector Machines refer to a supervised machine learning technique based on the latest results of the statistical learning theory. Given a vector space and a set of training points, i.e. positive and negative examples, SVMs find a separating hyperplane $H(\vec{x}) = \vec{\omega} \times \vec{x} + b = 0$ where $\omega \in R^n$ and $b \in R$ are learned by applying the Structural Risk Minimization principle (Vapnik, 1998). SVMs are a binary classifier, but they can be easily extended to multi-class classifier, e.g. by means of the *one-vs-all* method (Rifkin and Poggio, 2002).

One strong point of SVMs is the possibility to apply kernel methods to implicitly map data in a new space where the examples are *more easily* separable as described in the next section.

2.4 Kernel methods

Kernel methods (Schölkopf and Smola, 2001) are an attractive alternative to feature-based methods since the applied learning algorithm only needs to compute a product between a pair of objects (by means of kernel functions), avoiding the explicit feature representation. A kernel function is a scalar product in a possibly unknown feature space. More precisely, The object *o* is mapped in $\vec{x}$ with a feature function $\phi : \mathcal{O} \rightarrow \Re^n$, where $\mathcal{O}$ is the set of the objects.

The kernel trick allows us to rewrite the decision hyperplane as:

$$H(\vec{x}) = \Big(\sum_{i=1..l} y_i \alpha_i \vec{x}_i \Big) \cdot \vec{x} + b =$$

[2] http://www.nima.mil/gns/html
[3] http://www.alexandria.ucsb.edu
[4] http://directory.google.com/Top/Business
[5] http://www.europages.net

$$\sum_{i=1..l} y_i \alpha_i \vec{x}_i \cdot \vec{x} + b = \sum_{i=1..l} y_i \alpha_i \phi(o_i) \cdot \phi(o) + b,$$

where y_i is equal to 1 for positive and -1 for negative examples, $\alpha_i \in \Re$ with $\alpha_i \geq 0$, o_i $\forall i \in \{1, .., l\}$ are the training instances and the product $K(o_i, o) = \langle \phi(o_i) \cdot \phi(o) \rangle$ is the kernel function associated with the mapping ϕ.

Kernel engineering can be carried out by combining basic kernels with additive or multiplicative operators or by designing specific data objects (vectors, sequences and tree structures) for the target tasks.

Regarding NLP applications, kernel methods have attracted much interest due to the ability of implicitly exploring huge amounts of structural features. The parse tree kernel (Collins and Duffy, 2001) and string kernel (Lodhi et al., 2002) are examples of the well-known convolution kernels used in various NLP tasks.

2.5 Tree Kernels

Tree kernels represent trees in terms of their substructures (called tree fragments). Such fragments form a feature space which, in turn, is mapped into a vector space. Tree kernels measure the similarity between pair of trees by counting the number of fragments in common. There are three important characterizations of fragment type: the SubTrees (ST), the SubSet Trees (SST) and the Partial Trees (PT). For sake of space, we do not report the mathematical description of them, which is available in (Vishwanathan and Smola, 2002), (Collins and Duffy, 2001) and (Moschitti, 2006), respectively. In contrast, we report some descriptions in terms of feature space that may be useful to understand the new engineered kernels.

In principle, a SubTree (ST) is defined by taking any node along with its descendants. A SubSet Tree (SST) is a more general structure which does not necessarily include all the descendants. The distinction is that an SST must be generated by applying the same grammatical rule set which generated the original tree, as pointed out in (Collins and Duffy, 2001). A Partial Tree (PT) is a more general form of sub-structures obtained by relaxing constraints over the SSTs. Figure 1 shows the overall fragment set of the ST, SST and PT kernels for the syntactic parse tree of the sentence fragment: *gives a talk* .

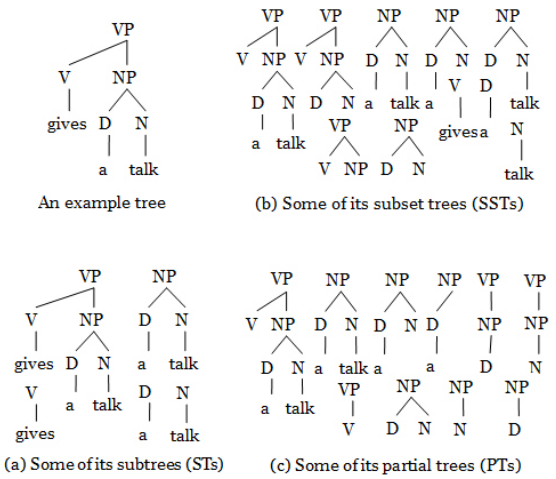


Figure 1: Three kinds of tree kernels.

In the next section, we will define new structures for tagged sequences of NEs which along with the application of the PT kernel produce innovative tagging kernels for reranking.

3 Reranking Method

3.1 Reranking Strategy

As a baseline we trained the CRFs model to generate *10*-best candidates per sentence, along with their probabilities. Each candidate was then represented by a semantic tree together with a feature vector. We consider our reranking task as a binary classification problem where examples are pairs of hypotheses $< H_i, H_j >$.

Given a sentence "**South African Breweries Ltd bought stakes in the Lech and Tychy brewers**" and three of its candidate tagged sequences:

H_1 *B-ORG I-ORG I-ORG I-ORG O O O O B-ORG O B-ORG O* (the correct sequence)

H_2 *B-MISC I-MISC B-ORG I-ORG O O O O B-ORG I-ORG I-ORG O*

H_3 *B-ORG I-ORG I-ORG I-ORG O O O O B-ORG O B-LOC O*

where B-ORG, I-ORG, B-LOC, O are the generated NE tags according to IOB notation as described in Section 3.2.

With the above data (an original sentence together with a list of candidate tagged sequences), the following pairs of hypotheses will be gener-

ated $< H_1, H_2 >$, $< H_1, H_3 >$,$< H_2, H_1 >$ and $< H_3, H_1 >$, where the first two pairs are positive and the latter pairs are negative instances. Then a binary classifier based on SVMs and kernel methods can be trained to discriminate between the best hypothesis, i.e. $< H_1 >$ and the others. At testing time the hypothesis receiving the highest score is selected (Collins and Duffy, 2001).

3.2 Representation of Tagged Sequences in Semantic Trees

We now consider the representation that exploits the most discriminative aspects of candidate structures. As in the case of NER, an input candidate is a sequence of word/tag pairs $x = \{w_1/t_1...w_n/t_n\}$ where w_i is the i'th word and t_i is the i'th NE tag for that word. The first representation we consider is the tree structure. See figure 2 as an example of candidate tagged sequence and its semantic tree.

With the sentence "**South African Breweries Ltd bought stakes in the Lech and Tychy brewers**" and three of its candidate tagged sequences in the previous section, the training algorithm considers to construct a tree for each sequence, with the named-entity tags as pre-terminals and the words as leaves. See figure 2 for an example of the semantic tree for the first tagged sequence.

With this tree representation, for a word w_i, the target NE tag would be set at parent and the features for this word are at child nodes. This allows us to best exploit the inner product between competing candidates. Indeed, in the kernel space, the inner product counts the number of common subtrees thus sequences with similar NE tags are likely to have higher score. For example, the similarity between H_1 and H_3 will be higher than the similarity of the previous hypotheses with H_2; this is reasonable since these two also have higher F_1.

It is worth noting that another useful modification is the flexibility of incorporate diverse, arbitrary features into this tree structure by adding children to the parent node that contains entity tag. These characteristics can be exploited efficiently with the PT kernel, which relaxes constraints of production rules. The inner product can implicitly include these features and deal better with sparse data.

3.3 Global features

Mixed *n-grams* features

In previous works, some global features have been used (Collins, 2002b; Collins, 2002a) but the employed algorithm just exploited arbitrary information regarding word types and linguistic patterns. In contrast, we define and study diverse features by also considering *n-grams* patterns preceding, and following the target entity.

Complementary *context*

In supervised learning, NER systems often suffer from low recall, which is caused by lack of both resource and context. For example, a word like "Arkansas" may not appear in the training set and in the test set, there may not be enough context to infer its NE tag. In such cases, neither global features (Chieu and Ng, 2002) nor aggregated contexts (Chieu and Ng, 2003) can help.

To overcome this deficiency, we employed the following unsupervised procedure: first, the baseline NER is applied to the target un-annotated corpus. Second, we associate each word of the corpus with the most frequent NE category assigned in the previous step. Finally, the above tags are used as features during the training of the improved NER and also for building the feature representation for a new classification instance.

This way, for any unknown word w of the test set, we can rely on the most probable NE category as feature. The advantage is that we derived it by using the average over many possible contexts of w, which are in the different instances of the unnanotated corpus.

The unlabeled corpus for Italian was collected from La Repubblica [6] and it contains over 20 millions words. Whereas the unlabeled corpus for English was collected mainly from The New York Times [7] and BBC news stories [8] with more than 35 millions words.

Head word

As the head word of an entity plays an important role in information extraction (Bunescu and Mooney, 2005a; Surdeanu et al., 2003), it is in-

[6] http://www.repubblica.it/
[7] http://www.nytimes.com/
[8] http://news.bbc.co.uk/

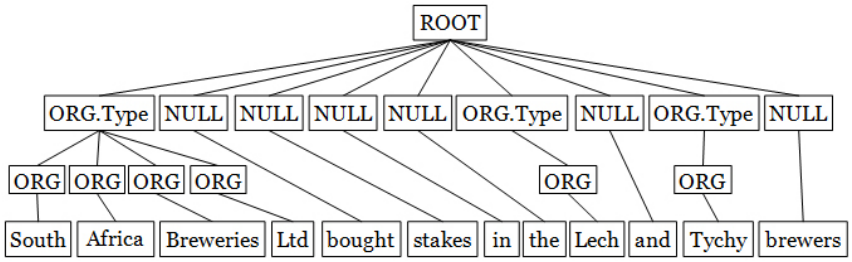


Figure 2: Semantic structure of the first sequence

cluded in the global set together with its orthographic feature. We now describe some primitives for our global feature framework.

1. w_i for $i = 1 \ldots n$ is the i'th word
2. t_i is the NE tag of w_i
3. g_i is the gazetteer feature of the word w_i
4. f_i is the most frequent NE tag seen in a large corpus of w_i
5. h_i is the head word of the entity. We normally set the head word of an entity as its last word. However, when a preposition exists in the entity string, its head word is set as the last word before the preposition. For example, the head word of the entity "University of Pennsylvania" is "University".
6. Mixed *n-grams* features of the words and their gazetteers/frequent-tag before/after the start/end of an entity. In addition to the normal *n-grams* solely based on words, we mixed words with gazetteers/frequent-tag seen from a large corpus and create mixed *n-grams* features.

Table 2 shows the full set of global features in our reranking framework. Features are anchored to each entity instance and adapted to entity types. This helps to discriminate different entities with the same surface forms. Moreover, they can be combined with *n-grams* patterns to learn and explicitly push the score of the correct sequence above the score of competing sequences.

3.4 Reranking with Composite Kernel

In this section we describe our novel tagging kernels based on diverse global features as well as semantic trees for reranking candidate tagged sequences. As mentioned in the previous section, we can engineer kernels by combining tree and entity kernels. Thus we focus on the problem to define structure embedding the desired relational information among tagged sequences.

The Partial Tree Kernel

Let $F = f_1, f_2, \ldots, f_{|F|}$ be a tree fragment space of type PTs and let the indicator function $I_i(n)$ be equal to 1 if the target f_1 is rooted at node n and 0 otherwise, we define the PT kernel as:

$$K(T_1, T_2) = \sum_{n_1 \in N_{T_1}} \sum_{n_2 \in N_{T_2}} \Delta(n_1, n_2)$$

where N_{T_1} and N_{T_2} are the set of nodes in T_1 and T_2 respectively and $\Delta(n_1, n_2) = \sum_{i=1}^{|F|} I_i(n_1) I_i(n_2)$, i.e. the number of common fragments rooted at the n_1 and n_2 nodes of the type shown in Figure 1.c.

The Polynomial Kernel

The polynomial kernel between two candidate tagged sequences is defined as:

$$K(x, y) = (1 + \vec{x}_1 \cdot \vec{x}_2)^2,$$

where $\vec{x}_1$ and $\vec{x}_2$ are two feature vectors extracted from the two sequences with the global feature template.

The Tagging Kernels

In our reranking framework, we incorporate the probability from the original model with the tree structure as well as the feature vectors. Let us consider the following notations:

Feature	Description
$w_s\_w_{s+1}\_\ldots\_w_e$	Entity string
$g_s\_g_{s+1}\_\ldots\_g_e$	The gazetteer feature within the entity
$f_s\_f_{s+1}\_\ldots\_f_e$	The most frequent NE tag feature (seen from a large corpus) within the entity
hw	The head word of the entity
lhw	Indicates whether the head word is lower-cased
$w_{s-1}\_w_s$; $w_{s-1}\_g_s$; $g_{s-1}\_w_s$; $g_{s-1}\_g_s$	Mixed bigrams of the words/gazetteer features before/after the start of the entity
$w_e\_w_{e+1}$; $w_e\_g_{e+1}$; $g_e\_w_{e+1}$; $g_e\_g_{e+1}$	Mixed bigrams of the words/gazetteer features before/after the end of the entity
$w_{s-1}\_w_s$; $w_{s-1}\_f_s$; $f_{s-1}\_w_s$; $f_{s-1}\_f_s$	Mixed bigrams of the words/frequent-tag features before/after the start of the entity
$w_e\_w_{e+1}$; $w_e\_f_{e+1}$; $f_e\_w_{e+1}$; $f_e\_f_{e+1}$	Mixed bigrams of the words/frequent-tag features before/after the end of the entity
$w_{s-2}\_w_{s-1}\_w_s$; $w_{s-1}\_w_s\_w_{s+1}$; $w_{e-1}\_w_e\_w_{e+1}$; $w_{e-2}\_w_{e-1}\_w_e$	Trigram features of the words before/after the start/end of the entity
$w_{s-2}\_w_{s-1}\_g_s$; $w_{s-2}\_g_{s-1}\_w_s$; $w_{s-2}\_g_{s-1}\_g_s$; $g_{s-2}\_w_{s-1}\_w_s$; $g_{s-2}\_w_{s-1}\_g_s$; $g_{s-2}\_g_{s-1}\_w_s$; $g_{s-2}\_g_{s-1}\_g_s$; $w_{s-1}\_w_s\_g_{s+1}$; $w_{s-1}\_g_s\_w_{s+1}$; $w_{s-1}\_g_s\_g_{s+1}$; $g_{s-1}\_w_s\_w_{s+1}$; $g_{s-1}\_w_s\_g_{s+1}$; $g_{s-1}\_g_s\_w_{s+1}$; $g_{s-1}\_g_s\_g_{s+1}$	Mixed trigrams of the words/gazetteer features before/after the start of the entity
$w_{e-1}\_w_e\_g_{e+1}$; $w_{e-1}\_g_e\_w_{e+1}$; $w_{e-1}\_g_e\_g_{e+1}$; $g_{e-1}\_w_e\_w_{e+1}$; $g_{e-1}\_w_e\_g_{e+1}$; $g_{e-1}\_g_e\_w_{e+1}$; $g_{e-1}\_g_e\_g_{e+1}$; $w_{e-2}\_w_{e-1}\_g_e$; $w_{e-2}\_g_{e-1}\_w_e$; $w_{e-2}\_g_{e-1}\_g_e$; $g_{e-2}\_w_{e-1}\_w_e$; $g_{e-2}\_w_{e-1}\_g_e$; $g_{e-2}\_g_{e-1}\_w_e$; $g_{e-2}\_g_{e-1}\_g_e$	Mixed trigrams of the words/gazetteer features before/after the end of the entity
$w_{s-2}\_w_{s-1}\_f_s$; $w_{s-2}\_f_{s-1}\_w_s$; $w_{s-2}\_f_{s-1}\_f_s$; $f_{s-2}\_w_{s-1}\_w_s$; $f_{s-2}\_w_{s-1}\_f_s$; $f_{s-2}\_f_{s-1}\_w_s$; $f_{s-2}\_f_{s-1}\_f_s$; $w_{s-1}\_w_s\_f_{s+1}$; $w_{s-1}\_f_s\_w_{s+1}$; $w_{s-1}\_f_s\_f_{s+1}$; $f_{s-1}\_w_s\_w_{s+1}$; $f_{s-1}\_w_s\_f_{s+1}$; $f_{s-1}\_f_s\_w_{s+1}$; $f_{s-1}\_f_s\_f_{s+1}$	Mixed trigrams of the words/frequent-tag features before/after the start of the entity
$w_{e-1}\_w_e\_f_{e+1}$; $w_{e-1}\_f_e\_w_{e+1}$; $w_{e-1}\_f_e\_f_{e+1}$; $f_{e-1}\_w_e\_w_{e+1}$; $f_{e-1}\_w_e\_f_{e+1}$; $f_{e-1}\_f_e\_w_{e+1}$; $f_{e-1}\_f_e\_f_{e+1}$; $w_{e-2}\_w_{e-1}\_f_e$; $w_{e-2}\_f_{e-1}\_w_e$; $w_{e-2}\_f_{e-1}\_f_e$; $f_{e-2}\_w_{e-1}\_w_e$; $f_{e-2}\_w_{e-1}\_f_e$; $f_{e-2}\_f_{e-1}\_w_e$; $f_{e-2}\_f_{e-1}\_f_e$	Mixed trigrams of the words/frequent-tag features before/after the end of the entity

Table 2: Global features in the entity kernel for reranking. These features are anchored for each entity instance and adapted to entity categories. For example, the entity string (first feature) of the entity "United Nations" with entity type "ORG" is "ORG United Nations".

- $K(x, y) = L(x) \cdot L(y)$ is the basic kernel where $L(x)$ is the log probability of a candidate tagged sequence x under the original probability model.
- $TK(x, y) = t(x) \cdot t(y)$ is the partial tree kernel under the structure representation
- $FK(x, y) = f(x) \cdot f(y)$ is the polynomial kernel under the global features

The tagging kernels between two tagged sequences are defined in the following combinations:

1. $CTK = \alpha \cdot K + (1 - \alpha) \cdot TK$
2. $CFK = \beta \cdot K + (1 - \beta) \cdot FK$
3. $CTFK = \gamma \cdot K + (1 - \gamma) \cdot (TK + FK)$

where α, β, γ are parameters weighting the two participating terms. Experiments on the validation set showed that these combinations yield the best performance with $\alpha = 0.2$ for both languages, $\beta = 0.4$ for English and $\beta = 0.3$ for and Italian, $\gamma = 0.24$ for English and $\gamma = 0.2$ for Italian.

4 Experimens and Results

4.1 Experimental Setup

As a baseline we trained the CRFs classifier on the full training portion (11,227 sentences in the Italian and 14,987 sentences in the English corpus). In developing a reranking strategy for both English and Italian, the training data was split into 5 sections, and in each case the baseline classifier was trained on 4/5 of the data, then used to decode the remaining 1/5.

The top 10 hypotheses together with their log probabilities were recovered for each training sentence. Similarly, a model trained on the whole training data was used to produce 10 hypotheses for each sentence in the development set. For the reranking experiments, we applied different kernel setups to the two corpora described in Section 2.1. The three kernels were trained on the training portion.

Italian Test	P	R	F
$CRFs$	83.43	77.48	80.34
CTK	84.97	78.03	81.35
CFK	84.93	79.13	81.93
CTFK	**85.99**	**82.73**	**84.33**
(Zanoli et al., 2009)	*84.07*	*80.02*	*82.00*

English Test	P	R	F
$CRFs$	85.37	84.35	84.86
CTK	87.19	84.79	85.97
CFK	86.53	86.75	86.64
CTFK	**88.07**	**88.25**	**88.16**
(Ratinov and Roth,)	*N/A*	*N/A*	*90.57*

Table 3: Reranking results of the three tagging kernels on the Italian and English testset.

4.2 Discussion

Table 3 presents the reranking results on the test data of both corpora. The results show a 20.29% relative improvement in F-measure for Italian and 21.79% for English.

CFK based on unstructured features achieves higher accuracy than CTK based on structured features. However, the huge amount of subtrees generated by the PT kernel may limit the expressivity of some structural features, e.g. many fragments may only generate noise. This problem is less important with the polynomial kernel where global features are tailored for individual entities.

In any case, the experiments demonstrate that both tagging kernels CTK and CFK give improvement over the CRFs baseline in both languages. This suggests that structured and unstructured features are effective in discriminating between competing NE annotations.

Furthermore, the combination of the two tagging kernels on both standard corpora shows a large improvement in F-measure from 80.34% to 84.33% for Italian and from 84.86% to 88.16% for English data. This suggests that these two kernels, corresponding to two kinds of feature, complement each other.

To better collocate our results with previous work, we report the best NER outcome on the Italian (Zanoli et al., 2009) and the English (Ratinov and Roth,) datasets, in the last row (in italic) of each table. This shows that our model outperforms the best Italian NER system and it is close to the state-of-art model for English, which exploits many complex features[9]. Also note that we are very close to the F1 achieved by the best system of CoNLL 2003, i.e. 88.8.

5 Conclusion

We analyzed the impact of kernel-based approaches for modeling dependencies between tagged sequences for NER. Our study illustrates that each individual kernel, either with structured or with flat features clearly gives improvement to the base model. Most interestingly, as we showed, these contributions are independent and, the approaches can be used together to yield better results. The composite kernel, which combines both kinds of features, can outperform the state-of-the-art.

In the future, it will be very interesting to use syntactic/semantic kernels, as for example in (Basili et al., 2005; Bloehdorn and Moschitti, 2007a; Bloehdorn and Moschitti, 2007b). Another promising direction is the use of syntactic trees, feature sequences and pairs of instances, e.g. (Nguyen et al., 2009; Moschitti, 2008).

Acknowledgments

We would like to thank Roberto Zanoli and Marco Dinarelli for helpful explanation about their work. This work has been partially funded by the LiveMemories project (http://www.livememories.org/) and Expert System (http://www.expertsystem.net/) research grant.

[9] In the future we will be able to integrate them with the authors collaboration.

References

Basili, Roberto, Marco Cammisa, and Alessandro Moschitti. 2005. Effective use of WordNet semantics via kernel-based learning. In *CoNLL*.

Bloehdorn, Stephan and Alessandro Moschitti. 2007a. Combined syntactic and semantic kernels for text classification. In *ECIR*.

Bloehdorn, Stephan and Alessandro Moschitti. 2007b. Structure and semantics for expressive text kernels. In *CIKM*.

Bunescu, Razvan C. and Raymond J. Mooney. 2005a. A shortest path dependency kernel for relation extraction. In *EMNLP*.

Carreras, Xavier, Lluís Màrques, and Llus Padró. 2002. Named entity extraction using Adaboost. In *CoNLL*.

Chieu, Hai Leong and Hwee Tou Ng. 2002. Named entity recognition: A maximum entropy approach using global information. In *COLING*.

Chieu, Hai Leong and Hwee Tou Ng. 2003. Named entity recognition with a maximum entropy approach. In *CoNLL*.

Chinchor, Nancy and Patricia Robinson. 1998. Muc-7 named entity task definition. In *the MUC*.

Collins, Michael and Nigel Duffy. 2001. Convolution kernels for natural language. In *NIPS*.

Collins, Michael and Terry Koo. 2000. Discriminative reranking for natural language parsing. In *ICML*.

Collins, Michael. 2002a. New ranking algorithms for parsing and tagging: Kernels over discrete structures, and the voted perceptron. In *ACL*.

Collins, Michael. 2002b. Ranking algorithms for named-entity extraction boosting and the voted perceptron. In *ACL*.

Dinarelli, Marco, Alessandro Moschitti, and Giuseppe Riccardi. 2009. Re-ranking models based on small training data for spoken language understanding. In *EMNLP*.

Florian, Radu, Abe Ittycheriah, Hongyan Jing, and Tong Zhang. 2003. Named entity recognition through classifier combination. In *CoNLL*.

Huang, Liang. 2008. Forest reranking: Discriminative parsing with non-local features. In *ACL-HLT*.

Lafferty, John, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*.

Lodhi, Huma, Craig Saunders, John Shawe Taylor, Nello Cristianini, , and Chris Watkins. 2002. Text classification using string kernels. *Journal of Machine Learning Research*, pages 419–444.

Magnini, Bernardo, Emmanuele Pianta, Christian Girardi, Matteo Negri, Lorenza Romano, Manuela Speranza, Valentina Bartalesi Lenzi, and Rachele Sprugnoli. 2006. I-CAB: the italian content annotation bank. In *LREC*.

Moschitti, Alessandro. 2004. A study on convolution kernels for shallow semantic parsing. In *ACL*.

Moschitti, Alessandro. 2006. Efficient convolution kernels for dependency and constituent syntactic trees. In *ICML*.

Moschitti, Alessandro. 2008. Kernel methods, syntax and semantics for relational text categorization. In *CIKM*.

Nguyen, Truc-Vien T., Alessandro Moschitti, and Giuseppe Riccardi. 2009. Convolution kernels on constituent, dependency and sequential structures for relation extraction. In *EMNLP*.

Ratinov, Lev and Dan Roth. Design challenges and misconceptions in named entity recognition. In *CoNLL*.

Rifkin, Ryan Michael and Tomaso Poggio. 2002. *Everything old is new again: a fresh look at historical approaches in machine learning*. PhD thesis, MIT.

Sang, Erik F. Tjong Kim and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *CoNLL*.

Schölkopf, Bernhard and Alexander J. Smola. 2001. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, USA.

Shen, Libin, Anoop Sarkar, and Franz Josef Och. 2004. Discriminative reranking for machine translation. In *HLT-NAACL*, Boston, Massachusetts, USA.

Surdeanu, Mihai, Sanda Harabagiu, John Williams, and Paul Aarseth. 2003. Using predicate-argument structures for information extraction. In *ACL*.

Vapnik, Vladimir N. 1998. *Statistical Learning Theory*. John Wiley and Sons, New York.

Vishwanathan, S.V.N. and Alexander J. Smola. 2002. Fast kernels on strings and trees. In *NIPS*.

Zanoli, Roberto, Emanuele Pianta, and Claudio Giuliano. 2009. Named entity recognition through redundancy driven classifiers. In *EVALITA*.

Opinion Summarization with Integer Linear Programming Formulation for Sentence Extraction and Ordering

Hitoshi Nishikawa, Takaaki Hasegawa, Yoshihiro Matsuo and **Genichiro Kikui**
NTT Cyber Space Laboratories, NTT Corporation
{ nishikawa.hitoshi, hasegawa.takaaki
  matsuo.yoshihiro, kikui.genichiro } @lab.ntt.co.jp

Abstract

In this paper we propose a novel algorithm for opinion summarization that takes account of content and coherence, simultaneously. We consider a summary as a sequence of sentences and directly acquire the optimum sequence from multiple review documents by extracting and ordering the sentences. We achieve this with a novel Integer Linear Programming (ILP) formulation. Our proposed formulation is a powerful mixture of the Maximum Coverage Problem and the Traveling Salesman Problem, and is widely applicable to text generation and summarization tasks. We score each candidate sequence according to its content and coherence. Since our research goal is to summarize reviews, the content score is defined by opinions and the coherence score is developed in training against the review document corpus. We evaluate our method using the reviews of commodities and restaurants. Our method outperforms existing opinion summarizers as indicated by its ROUGE score. We also report the results of human readability experiments.

1 Introduction

The Web now holds a massive number of reviews describing the opinions of customers about products and services. These reviews can help the customer to reach purchasing decisions and guide the business activities of companies such as product improvement. It is, however, almost impossible to read all reviews given their sheer number.

Automatic text summarization, particularly opinion summarization, is expected to allow all possible reviews to be efficiently utilized. Given multiple review documents, our summarizer outputs text consisting of ordered sentences. A typical summary is shown in Table 1. This task is considered as multidocument summarization.

This restaurant offers customers a delicious menu and a relaxing atmosphere. The staff are very friendly but the price is a little high.

Table 1: A typical summary.

Existing summarizers focus on organizing sentences so as to include important information in the given document into a summary under some size limitation. A serious problem is that most of these summarizers completely ignore coherence of the summary, which improves reader's comprehension as reported by Barzilay et al. (2002).

To make summaries coherent, the extracted sentences must be appropriately ordered. However, most summarization systems delink sentence extraction from sentence ordering, so a sentence can be extracted that can never be ordered naturally with the other extracted sentences. Moreover, due to recent advances in decoding techniques for text summarization, the summarizers tend to select shorter sentences to optimize summary content. It aggravates this problem.

Although a preceding work tackles this problem by performing sentence extraction and ordering simultaneously (Nishikawa et al., 2010), they adopt beam search and dynamic programming to search for the optimal solution, so their proposed method may fail to locate it.

To overcome this weakness, this paper proposes a novel Integer Linear Programming (ILP) formulation for searching for the optimal solution efficiently. We formulate the multidocument summarization task as an ILP problem that tries to optimize the content and coherence of the summary by extracting and ordering sentences simultaneously. We apply our method to opinion summarization and show that it outperforms state-of-the-art opinion summarizers in terms of ROUGE evaluations. Although in this paper we challenge

Coling 2008: Poster Volume, pages 910–918,
Beijing, August 2010

our method with opinion summarization, it can be widely applied to other text generation and summarization tasks.

This paper is organized as follows: Section 2 describes related work. Section 3 describes our proposal. Section 4 reports our evaluation experiments. We conclude this paper in Section 5.

2 Related Work

2.1 Sentence Extraction

Although a lot of summarization algorithms have been proposed, most of them solely extract sentences from a set of sentences in the source document set. These methods perform *extractive summarization* and can be formalized as follows:

$$\hat{S} = \underset{S \subseteq T}{\operatorname{argmax}} \mathcal{L}(S) \quad (1)$$
$$\text{s.t. } \operatorname{length}(S) \leq K$$

T stands for all sentences in the source document set and S is an arbitrary subset of T. $\mathcal{L}(\mathcal{S})$ is a function indicating the score of S as determined by one or more criteria. $\operatorname{length}(S)$ indicates the length of S, K is the maximum size of the summary. That is, most summarization algorithms search for, or decode, the set of sentences $\hat{S}$ that maximizes function $\mathcal{L}$ under the given maximum size of the summary K. Thus most studies focus on the design of function $\mathcal{L}$ and efficient search algorithms (i.e. argmax operation in Eq.1).

Objective Function

Many useful $\mathcal{L}$ functions have been proposed including the cosine similarity of given sentences (Carbonell and Goldstein, 1998) and centroid (Radev et al., 2004); some approaches directly learn function $\mathcal{L}$ from references (Kupiec et al., 1995; Hirao et al., 2002).

There are two approaches to defining the score of the summary. One defines the weight on each sentence forming the summary. The other defines a weight for a sub-sentence, *concept*, that the summary contains.

McDonald (2007) and Martins and Smith (2009) directly weight sentences and use MMR to avoid redundancy (Carbonell and Goldstein, 1998). In contrast to their approaches, we set weights on concepts, not sentences. Gillick and Favre (2009) reported that the concept-based model achieves better performance and scalability than the sentence-based model when it is formulated as ILP.

There is a wide range of choice with regard to the unit of the concept. Concepts include words and the relationship between named entities (Filatova and Hatzivassiloglou, 2004), bigrams (Gillick and Favre, 2009), and word stems (Takamura and Okumura, 2009).

Some summarization systems that target reviews, opinion summarizers, extract particular information, *opinion*, from the input sentences and leverage them to select important sentences (Carenini et al., 2006; Lerman et al., 2009). In this paper, since we aim to summarize reviews, the objective function is defined through opinion as the concept that the reviews contain. We explain our detailed objective function in Section 3. We describe features of above existing summarizers in Section 4 and compare our method to them as baselines.

Decoding Method

The algorithms proposed for argmax operation include the greedy method (Filatova and Hatzivassiloglou, 2004), stack decoding (Yih et al., 2007; Takamura and Okumura, 2009) and Integer Linear Programming (Clarke and Lapata, 2007; McDonald, 2007; Gillick and Favre, 2009; Martins and Smith, 2009). Gillick and Favre (2009) and Takamura and Okumura (2009) formulate summarization as a Maximum Coverage Problem. We also use this formulation. While these methods focus on extracting a set of sentences from the source document set, our method performs extraction and ordering simultaneously.

Some studies attempt to generate a single sentence (i.e. headline) from the source document (Banko et al., 2000; Deshpande et al., 2007). While they extract and order *words* from the source document as a unit, our model uses the unit of *sentences*. This problem can be formulated as the Traveling Salesman Problem and its variants. Banko et al. (2000) uses beam search to identify approximate solutions. Deshpande et al. (2007) uses ILP and a randomized algorithm to find the optimal solution.

2.2 Sentence Ordering

It is known that the readability of a collection of sentences, a summary, can be greatly improved by appropriately ordering them (Barzilay et al., 2002). Features proposed to create the appropriate order include publication date of document (Barzilay et al., 2002), content words (Lapata, 2003; Althaus et al., 2004), and syntactic role of

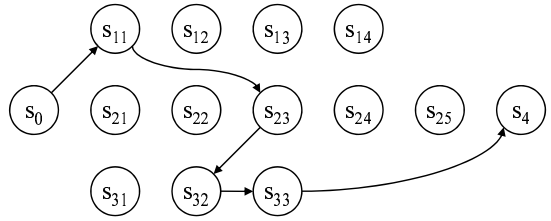


Figure 1: Graph representation of summarization.

words (Barzilay and Lapata, 2005). Some approaches use machine learning to integrate these features (Soricut and Marcu, 2006; Elsner et al., 2007). Generally speaking, these methods score the discourse coherence of a fixed set of sentences. These methods are separated from the extraction step so they may fail if the set includes sentences that are impossible to order naturally.

As mentioned above, there is a preceding work that attempted to perform sentence extraction and ordering simultaneously (Nishikawa et al., 2010). Differences between this paper and that work are as follows:

- This work adopts ILP solver as a decoder. ILP solver allows the summarizer to search for the optimal solution much more rapidly than beam search (Deshpande et al., 2007), which was adopted by the prior work. To permit ILP solver incorporation, we propose in this paper a totally new ILP formulation. The formulation can be widely used for text summarization and generation.
- Moreover, to learn better discourse coherence, we adopt the Passive-Aggressive algorithm (Crammer et al., 2006) and use Kendall's tau (Lapata, 2006) as the loss function. In contrast, the above work adopts Averaged Perceptron (Collins, 2002) and has no explicit loss function.

These advances make this work very different from that work.

3 Our Method

3.1 The Model

We consider a summary as a sequence of sentences. As an example, document set $D = \{d_1, d_2, d_3\}$ is given to a summarizer. We define d as a single document. Document d_1, which consists of four sentences, is describe by $d_1 = \{s_{11}, s_{12}, s_{13}, s_{14}\}$. Documents d_2 and d_3 consist of five sentences and three sentences (i.e. $d_2 = \{s_{21}, s_{22}, s_{23}, s_{24}, s_{25}\}$, $d_3 = \{s_{31}, s_{32}, s_{33}\}$). If the summary consists of four sentences $s_{11}, s_{23}, s_{32}, s_{33}$ and they are ordered as $s_{11} \rightarrow s_{23} \rightarrow s_{32} \rightarrow s_{33}$, we add symbols indicating the beginning of the summary s_0 and the end of the summary s_4, and describe the summary as $S = \langle s_0, s_{11}, s_{23}, s_{32}, s_{33}, s_4 \rangle$. Summary S can be represented as a directed path that starts at s_0 and ends at s_4 as shown in Fig. 1.

	e_1	e_2	e_3	$\ldots$	e_6	e_7	e_8
s_{11}	1	0	0		1	0	0
s_{12}	0	1	0		0	0	0
s_{13}	0	0	0		0	0	1
$\vdots$				$\ddots$			
s_{31}	0	0	0		0	0	0
s_{32}	0	0	1		0	1	0
s_{33}	0	0	0		0	0	1

Table 2: Sentence-Concept Matrix.

We describe a directed arc between s_i and s_j as $a_{i,j} \in A$. The directed path shown in Fig. 1 is decomposed into nodes, $s_0, s_{11}, s_{23}, s_{32}, s_{33}, s_4$, and arcs, $a_{0,11}, a_{11,23}, a_{23,32}, a_{32,33}, a_{33,4}$.

To represent the discourse coherence of two adjacent sentences, we define weight $c_{i,j} \in C$ as the coherence score on the directed arc $a_{i,j}$. We assume that better summaries have higher coherence scores, i.e. if the sum of the scores of the arcs $\sum_{a_{i,j} \in S} c_{i,j} a_{i,j}$ is high, the summary is coherent.

We also assume that the source document set D includes set of concepts $e \in E$. Each concept e is covered by one or more of the sentences in the document set. We show this schema in Table 2. According to Table 2, document set D has eight concepts $e_1, e_2, \ldots, e_7, e_8$ and sentence s_{11} includes concepts e_1 and e_6 while sentence s_{12} includes e_2.

We consider each concept e_i has a weight w_i. We assume that concept e_i will have high weight w_i if it is important. This paper improves summary quality by maximizing the sum of these weights.

We define, based on the above assumption, the following objective function:

$$\mathcal{L}(S) = \sum_{e_i \in S} w_i e_i + \sum_{a_{i,j} \in S} c_{i,j} a_{i,j} \quad (2)$$
$$\text{s.t. } \text{length}(S) \leq K$$

Summarization is, in this paper, realized by maximizing the sum of weights of concepts included in the summary and the coherence score of all adjacent sentences in the summary under the

limit of maximum summary size. Note that while S and T represents the *set* of sentences in Eq.1, they represent the *sequence* of sentences in Eq.2.

Maximizing Eq.2 is NP-hard. If each sentence in the source document set has one concept (i.e. Table 2 is a diagonal matrix), Eq.2 becomes the Prize Collecting Traveling Salesman Problem (Balas, 1989). Therefore, a highly efficient decoding method is essential.

3.2 Parameter Estimation

Our method requires two parameters: weights $w \in W$ of concepts and coherence $c \in C$ of two adjacent sentences. We describe them here.

Content Score

In this paper, as mentioned above, since we attempt to summarize reviews, we adopt *opinion* as a concept. We define opinion $e = \langle t, a, p \rangle$ as the tuple of *target* t, *aspect* a and its *polarity* $p \in \{-1, 0, 1\}$. We define target t as the target of an opinion. For example, the target t of the sentence "This digital camera has good image quality." is *digital camera*. We define aspect a as a word that represents a standpoint appropriate for evaluating products and services. With regard to digital cameras, aspects include *image quality*, *design* and *battery life*. In the above example sentence, the aspect is *image quality*. Polarity p represents whether the opinion is positive or negative. In this paper, we define $p = -1$ as negative, $p = 0$ as neutral and $p = 1$ as positive. Thus the example sentence contains opinion $e = \langle digital\ camera, image\ quality, 1 \rangle$.

Opinions are extracted using a sentiment expression dictionary and pattern matching from dependency trees of sentences. This opinion extractor is the same as that used in Nishikawa et al. (2010).

As the weight w_i of concept e_i, we use only the frequency of each opinion in the input document set, i.e. we assume that an opinion that appears frequently in the input is important. While this weighting is relatively naive compared to Lerman et al. (2009)'s method, our ROUGE evaluation shows that this approach is effective.

Coherence Score

In this section, we define coherence score c. Since it is not easy to model the global coherence of a set of sentences, we approximate the global coherence by the sum of local coherence i.e. the sum of coherence scores of sentence pairs. We define local coherence score $c_{i,j}$ of two sentences $x = \{s_i, s_j\}$ and their order $y = \langle s_i, s_j \rangle$ representing $s_i \rightarrow s_j$ as follows:

$$c_{i,j} = \mathbf{w} \cdot \phi(x, y) \tag{3}$$

$\mathbf{w} \cdot \phi(x, y)$ is the inner product of $\mathbf{w}$ and $\phi(x, y)$, $\mathbf{w}$ is a parameter vector and $\phi(x, y)$ is a feature vector of the two sentences s_i and s_j.

Since coherence consists of many different elements and it is difficult to model all of them, we approximate the features of coherence as the Cartesian product of the following features: content words, POS tags of content words, named entity tags (e.g. `LOC`, `ORG`) and conjunctions. Lapata (2003) proposed most of these features.

We also define feature vector $\Phi(\mathbf{x}, \mathbf{y})$ of the bag of sentences $\mathbf{x} = \{s_0, s_1, \ldots, s_n, s_{n+1}\}$ and its entire order $\mathbf{y} = \langle s_0, s_1, \ldots, s_n, s_{n+1} \rangle$ as follows:

$$\Phi(\mathbf{x}, \mathbf{y}) = \sum_{x,y} \phi(x, y) \tag{4}$$

Therefore, the score of order $\mathbf{y}$ is $\mathbf{w} \cdot \Phi(\mathbf{x}, \mathbf{y})$. Given a training set, if trained parameter vector $\mathbf{w}$ assigns score $\mathbf{w} \cdot \Phi(\mathbf{x}, \mathbf{y}_t)$ to correct order $\mathbf{y}_t$ that is higher than score $\mathbf{w} \cdot \Phi(\mathbf{x}, \hat{\mathbf{y}})$ assigned to incorrect order $\hat{\mathbf{y}}$, it is expected that the trained parameter vector will give a higher score to coherently ordered sentences than to incoherently ordered sentences.

We use the Passive-Aggressive algorithm (Crammer et al., 2006) to find $\mathbf{w}$. The Passive-Aggressive algorithm is an online learning algorithm that updates the parameter vector by taking up one example from the training examples and outputting the solution that has the highest score under the current parameter vector. If the output differs from the training example, the parameter vector is updated as follows;

$$\begin{gathered} \min ||\mathbf{w}^{i+1} - \mathbf{w}^{i}|| \\ \text{s.t. } \mathrm{s}(\mathbf{x}, \mathbf{y}_t; \mathbf{w}^{i+1}) - \mathrm{s}(\mathbf{x}, \hat{\mathbf{y}}; \mathbf{w}^{i+1}) \geq \ell(\hat{\mathbf{y}}; \mathbf{y}_t) \\ \mathrm{s}(\mathbf{x}, \mathbf{y}; \mathbf{w}) = \mathbf{w} \cdot \Phi(\mathbf{x}, \mathbf{y}) \end{gathered} \tag{5}$$

$\mathbf{w}^i$ is the current parameter vector and $\mathbf{w}^{i+1}$ is the updated parameter vector. That is, Eq.5 means that the score of the correct order must exceed the score of an incorrect order by more than loss function $\ell(\hat{\mathbf{y}}; \mathbf{y}_t)$ while minimizing the change in parameters.

When updating the parameter vector, this algorithm requires the solution that has the highest score under the current parameter vector, so we have to run an argmax operation. Since we are

attempting to order a set of sentences, the operation is regarded as solving the Traveling Salesman Problem (Althaus et al., 2004); that is, we locate the path that offers the maximum score through all n sentences where s_0 and s_{n+1} are starting and ending points, respectively. This operation is NP-hard and it is difficult to find the global optimal solution. To overcome this, we find an approximate solution by beam search.[1]

We define loss function $\ell(\hat{\mathbf{y}}; \mathbf{y}_t)$ as follows:

$$\ell(\hat{\mathbf{y}}; \mathbf{y}_t) = 1 - \tau \quad (6)$$

$$\tau = 1 - 4\frac{\mathrm{S}(\hat{\mathbf{y}}, \mathbf{y}_t)}{N(N-1)} \quad (7)$$

τ indicates Kendall's tau. $\mathrm{S}(\hat{\mathbf{y}}, \mathbf{y}_t)$ is the minimum number of operations that swap adjacent elements (i.e. sentences) needed to bring $\hat{\mathbf{y}}$ to $\mathbf{y}_t$ (Lapata, 2006). N indicates the number of elements. Since Lapata (2006) reported that Kendall's tau reliably reproduces human ratings with regard to sentence ordering, using it to minimize the loss function is expected to yield more reliable parameters.

We omit detailed derivations due to space limitations. Parameters are updated as per the following equation.

$$\mathbf{w}^{i+1} = \mathbf{w}^i + \eta^i(\Phi(\mathbf{x}, \mathbf{y}_t) - \Phi(\mathbf{x}, \hat{\mathbf{y}})) \quad (8)$$

$$\eta^i = \frac{\ell(\hat{\mathbf{y}}; \mathbf{y}_t) - \mathrm{s}(\mathbf{x}, \mathbf{y}_t; \mathbf{w}^i) + \mathrm{s}(\mathbf{x}, \hat{\mathbf{y}}; \mathbf{w}^i)}{||\Phi(\mathbf{x}, \mathbf{y}_t) - \Phi(\mathbf{x}, \hat{\mathbf{y}})||^2 + \frac{1}{2C}} \quad (9)$$

C in Eq.9 is the *aggressiveness parameter* that controls the degree of parameter change.

Note that our method learns $\mathbf{w}$ from documents automatically annotated by a POS tagger and a named entity tagger. That is, manual annotation isn't required.

3.3 Decoding with Integer Linear Programming Formulation

This section describes an ILP formulation of the above model. We use the same notation convention as introduced in Section 3.1. We use $s \in S, a \in A, e \in E$ as the decision variable. Variable $s_i \in S$ indicates the inclusion of the i th sentence. If the i th sentence is part of the summary, then s_i is 1. If it is not part of the summary, then s_i is 0. Variable $a_{i,j} \in A$ indicates the adjacency of the i th and j th sentences. If these two sentences are ordered as $s_i \rightarrow s_j$, then $a_{i,j}$ is 1. Variable $e_i \in E$ indicates the inclusion of the i th concept e_i. Taking Fig.1 as an example, variables $s_0, s_{11}, s_{23}, s_{32}, s_{33}, s_4$ and $a_{0,11}, a_{11,23}, a_{23,32}, a_{32,33}, a_{33,4}$ are 1. e_i, which correspond to the concepts in the above extracted sentences, are also 1.

We represent the above objective function (Eq.2) as follows:

$$\max\left\{\lambda \sum_{e_i \in E} w_i e_i + (1-\lambda) \sum_{a_{i,j} \in A} c_{i,j} a_{i,j}\right\} \quad (10)$$

Eq.10 attempts to cover as much of the concepts included in input document set as possible according to their weights $w \in W$ and orders sentences according to discourse coherence $c \in C$. λ is a scaling factor to balance w and c.

We then impose some constraints on Eq.10 to acquire the optimum solution.

First, we range the above three variables $s \in S, a \in A, e \in E$.

$$s_i, a_{i,j}, e_i \in \{0, 1\} \ \ \forall i, j$$

In our model, a summary can't include the same sentence, arc, or concept twice. Taking Table 2 for example, if s_{13} and s_{33} are included in a summary, the summary has two e_8, but e_8 is 1. This constraint avoids summary redundancy.

The summary must meet the condition of maximum summary size. The following inequality represents the size constraint:

$$\sum_{s_i \in S} l_i s_i \leq K$$

$l_i \in L$ indicates the length of sentence s_i. K is the maximum size of the summary.

The following inequality represents the relationship between sentences and concepts in the sentences.

$$\sum_i m_{ij} s_i \geq e_j \ \ \forall j$$

The above constraint represents Table 2. $m_{i,j}$ is an element of Table 2. If s_i is not included in the summary, the concepts in s_i are not included.

Symbols indicating the beginning and end of the summary must be part of the summary.

[1] Obviously, ILP can be used to search for the path that maximizes the score. While beam search tends to fail to find out the optimal solution, it is tractable and the learning algorithm can estimate the parameter from approximate solutions. For these reasons we use beam search.

$$
\begin{aligned}
s_0 &= 1 \\
s_{n+1} &= 1
\end{aligned}
$$

n is the number of sentences in the input document set.

Next, we describe the constraints placed on arcs.

The beginning symbol must be followed by a sentence or a symbol and must not have any preceding sentences/symbols. The end symbol must be preceded by a sentence or a symbol and must not have any following sentences/symbols. The following equations represent these constraints:

$$
\begin{aligned}
\sum_i a_{0,i} &= 1 \\
\sum_i a_{i,0} &= 0 \\
\sum_i a_{n+1,i} &= 0 \\
\sum_i a_{i,n+1} &= 1
\end{aligned}
$$

Each sentence in the summary must be preceded and followed by a sentence/symbol.

$$
\begin{aligned}
\sum_i a_{i,j} + \sum_i a_{j,i} &= 2s_j \ \ \forall j \\
\sum_i a_{i,j} &= \sum_i a_{j,i} \ \ \forall j
\end{aligned}
$$

The above constraints fail to prevent cycles. To rectify this, we set the following constraints.

$$
\begin{aligned}
\sum_i f_{0,i} &= n \\
\sum_i f_{i,0} &\geq 1 \\
\sum_i f_{i,j} - \sum_i f_{j,i} &= s_j \ \ \forall j \\
f_{i,j} &\leq n a_{i,j} \ \ \forall i,j
\end{aligned}
$$

The above constraints indicate that *flows* f are sent from s_0 as a source to s_{n+1} as a sink. n unit flows are sent from the source and each node expends one unit of flows. More than one flow has to arrive at the sink. By setting these constraints, the nodes consisting of a cycle have no flow. Thus solutions that contain a cycle are prevented. These constraints have also been used to avoid cycles in headline generation (Deshpande et al., 2007).

4 Experiments

This section evaluates our method in terms of ROUGE score and readability. We tested our method and two baselines in two domains: reviews of commodities and restaurants. We collected 4,475 reviews of 100 commodities and 2,940 reviews of 100 restaurants from websites. The commodities included items such as digital cameras, printers, video games, and wines. The average document size was 10,173 bytes in the commodity domain and 5,343 bytes in the restaurant domain. We attempted to generate 300 byte summaries, so the summarization rates were about 3% and 6%, respectively.

We prepared 4 references for each review, thus there were 400 references in each domain. The authors were not those who made up the references. These references were used for ROUGE and readability evaluation.

Since our method requires the parameter vector $\mathbf{w}$ for determining the coherence scores. We trained the parameter vector for each domain. Each parameter vector was trained using 10-fold cross validation. We used 8 samples to train, 1 to develop, and 1 to test. In the restaurant domain, we added 4,390 reviews to each training set to alleviate data sparseness. In the commodity domain, we add 47,570 reviews.[2]

As the solver, we used glpk.[3] According to the development set, λ in Eq.10 was set as 0.1.

4.1 Baselines

We compare our method to the references (which also provide the upper bound) and the opinion summarizers proposed by Carenini et al. (2006) and Lerman et al. (2009) as the baselines.

In the ROUGE evaluations, Human indicates ROUGE scores between references. To compare our summarizer to human summarization, we calculated ROUGE scores between each reference and the other three references, and averaged them.

In the readability evaluations, we randomly selected one reference for each commodity and each restaurant and compared them to the results of the three summarizers.

Carenini et al. (2006)

Carenini et al. (2006) proposed two opinion

[2]The commodities domain suffers from stronger review variation than the restaurant domain so more training data was needed.

[3]`http://www.gnu.org/software/glpk/`

summarizers. One uses a natural language generation module, and other is based on MEAD (Radev et al., 2004). Since it is difficult to mimic the natural language generation module, we implemented the latter one. The objective function Carenini et al. (2006) proposed is as follows:

$$\mathcal{L}_1(S) = \sum_{a \in S} \sum_{s \in D} |\mathrm{polarity}_s(a)| \quad (11)$$

$\mathrm{polarity}_s(a)$ indicates the polarity of aspect a in sentence s present in source document set D. That is, this function gives a high score to a summary that covers aspects frequently mentioned in the input, and whose polarities tend to be either positive or negative.

The solution is identified using the greedy method. If there is more than one sentence that has the same score, the sentence that has the higher centroid score (Radev et al., 2004) is extracted.

Lerman et al. (2009)

Lerman et al. (2009) proposed three objective functions for opinion summarization, and we implemented one of them. The function is as follows:

$$\mathcal{L}_2(S) = -(\mathrm{KL}(p_S(a), p_D(a)) \quad (12)$$
$$+ \sum_{a \in A} \mathrm{KL}(\mathcal{N}(x|\mu_{a_S}, \sigma^2_{a_S}), \mathcal{N}(x|\mu_{a_D}, \sigma^2_{a_D})))$$

$\mathrm{KL}(p, q)$ means the Kullback-Leibler divergence between probability distribution p and q. $p_S(a)$ and $p_D(a)$ are probability distributions indicating how often aspect $a \in A$ occurs in summary S and source document set D respectively. $\mathcal{N}(x|\mu, \sigma^2)$ is a Gaussian distribution indicating distribution of polarity of an aspect whose mean is μ and variance is σ^2. μ_{a_S}, μ_{a_D} and $\sigma^2_{a_S}, \sigma^2_{a_D}$ are the means and the variances of aspect a in summary S and source document set D, respectively. These parameters are determined using maximum-likelihood estimation.

That is, the above objective function gives high score to a summary whose distributions of aspects and polarities mirror those of the source document set.

To identify the optimal solution, Lerman et al. (2009) use a randomized algorithm. First, the summarizer randomly extracts sentences from the source document set, then iteratively performs insert/delete/swap operations on the summary to increase Eq.12 until summary improvement saturates. While this method is prone to lock onto local solutions, the summarizer can reach the optimal solution by changing the starting sentences and repeating the process. In this experiment, we used 100 randomly selected starting points.

Commodity	R-2	R-SU4	R-SU9
(Carenini et al., 2006)	0.158	0.202	0.186
(Lerman et al., 2009)	0.205	0.247	0.227
Our Method	**0.231**	**0.251**	**0.230**
Human	0.384	0.392	0.358

Restaurant	R-2	R-SU4	R-SU9
(Carenini et al., 2006)	0.251	0.281	0.258
(Lerman et al., 2009)	0.260	0.296	**0.273**
Our Method	**0.285**	**0.303**	**0.273**
Human	0.358	0.370	0.335

Table 3: Automatic ROUGE evaluation.

	# of Sentences
(Carenini et al., 2006)	3.79
(Lerman et al., 2009)	6.28
Our Method	7.88
Human	5.83

Table 4: Average number of sentences in the summary.

4.2 ROUGE

We used ROUGE (Lin, 2004) for evaluating the content of summaries. We chose ROUGE-2, ROUGE-SU4 and ROUGE-SU9. We prepared four reference summaries for each document set.

The results of these experiments are shown in Table 3. ROUGE scores increase in the order of (Carenini et al., 2006), (Lerman et al., 2009) and our method, but no method could match the performance of Human. Our method significantly outperformed Lerman et al. (2009)'s method over ROUGE-2 according to the Wilcoxon signed-rank test, while it shows no advantage over ROUGE-SU4 and ROUGE-SU9.

Although our weighting of the set of sentences is relatively naive compared to the weighting proposed by Lerman et al. (2009), our method outperforms their method. There are two reasons for this; one is that we adopt ILP for decoding, so we can acquire preferable solutions efficiently. While the score of Lerman et al. (2009)'s method may be improved by adopting ILP, it is difficult to do so because their objective function is extremely complex. The other reason is the coherence score. Since our coherence score is based on

Commodity	(Carenini et al., 2006)	(Lerman et al., 2009)	Our Method	Human
(Carenini et al., 2006)	-	27/45	18/29	8/46
(Lerman et al., 2009)	18/45	-	29/48	11/47
Our Method	11/29	19/48	-	5/46
Human	38/46	36/47	41/46	-

Restaurant	(Carenini et al., 2006)	(Lerman et al., 2009)	Our Method	Human
(Carenini et al., 2006)	-	31/45	17/31	8/48
(Lerman et al., 2009)	14/45	-	25/47	7/46
Our Method	14/31	22/47	-	8/50
Human	40/48	39/46	42/50	-

Table 5: Readability evaluation.

content words, it may impact the content of the summary.

4.3 Readability

Readability was evaluated by human judges. Since it is difficult to perform absolute evaluation to judge the readability of summaries, we performed a paired comparison test. The judges were shown two summaries of the same input and decided which was more readable. The judges weren't informed which method generated which summary. We randomly chose 50 sets of reviews from each domain, so there were 600 paired summaries.[4] However, as shown in Table 4, the average numbers of sentences in the summary differed widely from the methods and this might affect the readability evaluation. It was not fair to include the pairs that were too different in terms of the number of sentences. Therefore, we removed the pairs that differed by more than five sentences. In the experiment, 523 pairs were used, and 21 judges evaluated about 25 summaries each. We drew on DUC 2007 quality questions[5] for readability assessment.

Table 5 shows the results of the experiment. Each element in the table indicates the number of times the corresponding method won against other method. For example, in the commodity domain, the summaries that Lerman et al. (2009)'s method generated were compared with the summaries that Carenini et al. (2006)'s method generated 45 times, and Lerman et al. (2009)'s method won 18 times. The judges significantly preferred the references in both domains. There were no significant differences between our method and the other two methods. In the restaurant domain, there was a significant difference between (Carenini et al., 2006) and (Lerman et al., 2009).

Since we adopt ILP, our method tends to pack shorter sentences into the summary. However, our coherence score prevents this from degrading summary readability.

[4] ${}_4\mathrm{C}_2 \times 100 = 600$

[5] http://www-nlpir.nist.gov/projects/duc/duc2007/quality-questions.txt

5 Conclusion

This paper proposed a novel algorithm for opinion summarization that takes account of content and coherence, simultaneously. Our method directly searches for the optimum sentence sequence by extracting and ordering sentences present in the input document set. We proposed a novel ILP formulation against selection-and-ordering problems; it is a powerful mixture of the Maximum Coverage Problem and the Traveling Salesman Problem. Experiments revealed that the algorithm creates summaries that have higher ROUGE scores than existing opinion summarizers. We also performed readability experiments. While our summarizer tends to extract shorter sentences to optimize summary content, our proposed coherence score prevented this from degrading the readability of the summary.

One future work includes enriching the features used to determine the coherence score. We expect that features such as entity grid (Barzilay and Lapata, 2005) will improve overall algorithm performance. We also plan to apply our model to tasks other than opinion summarization.

Acknowledgments

We would like to sincerely thank Tsutomu Hirao for his comments and discussions. We would also like to thank the anonymous reviewers for their comments.

References

Althaus, Ernst, Nikiforos Karamanis and Alexander Koller. 2004. Computing Locally Coherent Discourses. In *Proc. of the 42nd Annual Meeting of the Association for Computational Linguistics*.

Balas, Egon. 1989. The prize collecting traveling salesman problem. *Networks*, 19(6):621–636.

Banko, Michele, Vibhu O. Mittal and Michael J. Witbrock. 2000. Headline Generation Based on Statistical Translation. In *Proc. of the 38th Annual Meeting of the Association for Computational Linguistics*.

Barzilay, Regina, Noemie Elhadad and Kathleen McKeown. 2002. Inferring Strategies for Sentence Ordering in Multidocument Summarization. *Journal of Artificial Intelligence Research*, 17:35–55.

Barzilay, Regina and Mirella Lapata. 2005. Modeling Local Coherence: An Entity-based Approach. In *Proc. of the 43rd Annual Meeting of the Association for Computational Linguistics*.

Carbonell, Jaime and Jade Goldstein. 1998. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proc. of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Carenini, Giuseppe, Raymond Ng and Adam Pauls. 2006. Multi-Document Summarization of Evaluative Text. In *Proc. of the 11th Conference of the European Chapter of the Association for Computational Linguistics*.

Clarke, James and Mirella Lapata. 2007. Modelling Compression with Discourse Constraints. In *Proc. of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*.

Collins, Michael. 2002. Discriminative Training Methods for Hidden Markov Models: Theory and Experiments with Perceptron Algorithms. In *Proc. of the 2002 Conference on Empirical Methods in Natural Language Processing*.

Crammer, Koby, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz and Yoram Singer. 2006. Online passive-aggressive algorithms. *Journal of Machine Learning Research*, 7:551–585.

Deshpande, Pawan, Regina Barzilay and David R. Karger. 2007. Randomized Decoding for Selection-and-Ordering Problems. In *Proc. of Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics*.

Elsner, Micha, Joseph Austerweil and Eugene Charniak. 2007. A unified local and global model for discourse coherence. In *Proc. of Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics*.

Filatova, Elena and Vasileios Hatzivassiloglou. 2004. A Formal Model for Information Selection in Multi-Sentence Text Extraction. In *Proc. of the 20th International Conference on Computational Linguistics*.

Gillick, Dan and Benoit Favre. 2009. A Scalable Global Model for Summarization. In *Proc. of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics Workshop on Integer Linear Programming for NLP*.

Hirao, Tsutomu, Hideki Isozaki, Eisaku Maeda and Yuji Matsumoto. 2002. Extracting important sentences with support vector machines. In *Proc. of the 19th International Conference on Computational Linguistics*.

Kupiec, Julian, Jan Pedersen and Francine Chen. 1995. A Trainable Document Summarizer. In *Proc. of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Lapata, Mirella. 2003. Probabilistic Text Structuring: Experiments with Sentence Ordering. In *Proc. of the 41st Annual Meeting of the Association for Computational Linguistics*.

Lapata, Mirella. 2006. Automatic Evaluation of Information Ordering: Kendall's Tau. *Computational Linguistics*, 32(4):471–484.

Lerman, Kevin, Sasha Blair-Goldensohn and Ryan McDonald. 2009. Sentiment Summarization: Evaluating and Learning User Preferences. In *Proc. of the 12th Conference of the European Chapter of the Association for Computational Linguistics*.

Lin, Chin-Yew. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Proc. of Text Summarization Branches Out*.

Martins, Andre F. T., and Noah A. Smith. 2009. Summarization with a Joint Model for Sentence Extraction and Compression. In *Proc. of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics Workshop on Integer Linear Programming for NLP*.

McDonald, Ryan. 2007. A Study of Global Inference Algorithms in Multi-document Summarization. In *Proc. of the 29th European Conference on Information Retrieval*.

Nishikawa, Hitoshi, Takaaki Hasegawa, Yoshihiro Matsuo and Genichiro Kikui. 2010. Optimizing Informativeness and Readability for Sentiment Summarization. In *Proc. of the 48th Annual Meeting of the Association for Computational Linguistics*.

Radev, Dragomir R., Hongyan Jing, Magorzata Sty and Daniel Tam. 2004. Centroid-based summarization of multiple documents. *Information Processing and Management*, 40(6):919–938.

Soricut, Radu and Daniel Marcu. 2006. Discourse Generation Using Utility-Trained Coherence Models. In *Proc. of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*.

Takamura, Hiroya and Manabu Okumura. 2009. Text Summarization Model based on Maximum Coverage Problem and its Variant. In *Proc. of the 12th Conference of the European Chapter of the Association for Computational Linguistics*.

Yih, Wen-tau, Joshua Goodman, Lucy Vanderwende and Hisami Suzuki. 2007. Multi-Document Summarization by Maximizing Informative Content-Words. In *Proc. of the 20th International Joint Conference on Artificial Intelligence*.

A Study on Position Information in Document Summarization

You Ouyang Wenjie Li Qin Lu Renxian Zhang
Department of Computing, the Hong Kong Polytechnic University
{csyouyang,cswjli,csluqin,csrzhang}@comp.polyu.edu.hk

Abstract

Position information has been proved to be very effective in document summarization, especially in generic summarization. Existing approaches mostly consider the information of sentence positions in a document, based on a sentence position hypothesis that the importance of a sentence decreases with its distance from the beginning of the document. In this paper, we consider another kind of position information, i.e., the word position information, which is based on the ordinal positions of word appearances instead of sentence positions. An extractive summarization model is proposed to provide an evaluation framework for the position information. The resulting systems are evaluated on various data sets to demonstrate the effectiveness of the position information in different summarization tasks. Experimental results show that word position information is more effective and adaptive than sentence position information.

1 Introduction

Position information has been frequently used in document summarization. It springs from human's tendency of writing sentences of greater topic centrality at particular positions in a document. For example, in newswire documents, topic sentences are usually written earlier. A *sentence position hypothesis* is then given as: the first sentence in a document is the most important and the importance decreases as the sentence gets further away from the beginning. Based on this sentence position hypothesis, sentence position features are defined by the ordinal position of sentences. These position features have been proved to be very effective in generic document summarization. In more recent summarization tasks, such as query-focused and update summarization tasks, position features are also widely used.

Although in these tasks position features may be used in different ways, they are all based on the sentence position hypothesis. So we regard them as providing the sentence position information. In this paper, we study a new kind of position information, i.e., the word position information. The motivation of word position information comes from the idea of assigning different importance to multiple appearances of one word in a document.

As to many language models such as the bag-of-words model, it is well acknowledged that a word which appears more frequently is usually more important. If we take a closer look at all the appearances of one word, we can view this as a process that the different appearances of the same word raise the importance of each other. Now let's also take the order of the appearances into account. When reading a document, we can view it as a word token stream from the first token to the last. When a new token is read, we attach more importance to previous tokens that have the same lemma because they are just repeated by the new token. Inspired by this, we postulate a *word position hypothesis* here: for all the appearances of a fixed word, the importance of each appearance depends on all its following appearances. Therefore, the first appearance of a word is the most important and the importance decreases with the ordinal

Coling 2008: Poster Volume, pages 919–927,
Beijing, August 2010

positions of the appearances. Then, a novel kind of position features can be defined for the word appearances based on their ordinal positions. We believe that these word position features have some advantages when compared to traditional sentence position features. According to the sentence position hypothesis, sentence position features generally prefer earlier sentences in a document. As to the word position features that attempt to differentiate word appearances instead of sentences, a sentence which is not the first one in the document may still not be penalized as long as its words do not appear in previous sentences. Therefore, word position features are able to discover topic sentences in deep positions of the document. On the other hand, the assertion that the first sentence is always the most important is not true in actual data. It depends on the writing style indeed. For example, some authors may like to write some background sentences before topic sentences. In conclusion, we can expect word position features to be more adaptive to documents with different structures.

In the study of this paper, we define several word position features based on the ordinal positions of word appearances. We also develop a word-based summarization system to evaluate the effectiveness of the proposed word position features on a series of summarization data sets. The main contributions of our work are:

(1) representation of word position information, which is a new kind of position information in document summarization area.

(2) empirical results on various data sets that demonstrate the impact of position information in different summarization tasks.

2 Related Work

The use of position information in document summarization has a long history. In the seminal work by (Luhn, 1958), position information was already considered as a good indicator of significant sentences. In (Edmundson, 1969), a location method was proposed that assigns positive weights to the sentences to their ordinal positions in the document. Position information has since been adopted by many successful summarization systems, usually in the form of sentence position features. For example, Radev et al. (2004) developed a feature-based system MEAD based on word frequencies and sentence positions. The position feature was defined as a descending function of the sentence position. The MEAD system performed very well in the generic multi-document summarization task of the DUC 2004 competition. Later, position information is also applied to more summarization tasks. For example, in query-focused task, sentence position features are widely used in learning-based summarization systems as a component feature for calculating the composite sentence score (Ouyang et al, 2007; Toutanova et al, 2007). However, the effect of position features alone was not studied in these works.

There were also studies aimed at analyzing and explaining the effectiveness of position information. Lin and Hovy (1997) provided an empirical validation on the sentence position hypothesis. For each position, the *sentence position yield* was defined as the average value of the significance of the sentences with the fixed position. It was observed that the average significance at earlier positions was indeed larger. Nenkova (2005) did a conclusive overview on the DUC 2001-2004 evaluation results. It was reported that position information is very effective in generic summarization. In generic single-document summarization, a lead-based baseline that simply takes the leading sentences as the summary can outperform most submitted summarization system in DUC 2001 and 2002. As in multi-document summarization, the position-based baseline system is competitive in generating short summaries but not in longer summaries. Schilder and Kondadadi (2008) analyzed the effectiveness of the features that are used in their learning-based sentence scoring model for query-focused summarization. By comparing the ROUGE-2 results of each individual feature, it was reported that position-based features are less effective than frequency-based features. In (Gillick et al., 2009), the effect of position information in the update summarization task was studied. By using ROUGE to measure the density of valuable words at each sentence position, it was observed that the first sentence of newswire document was especially important for composing update summaries. They defined a binary sentence position feature based on the

observation and the feature did improve the performance on the update summarization data.

3 Methodology

In the section, we first describe the word-based summarization model. The word position features are then defined and incorporated into the summarization model.

3.1 Basic Summarization Model

To test the effectiveness of position information in document summarization, we first propose a word-based summarization model for applying the position information. The system follows a typical extractive style that constructs the target summary by selecting the most salient sentences.

Under the bag-of-words model, the probability of a word w in a document set D can be scaled by its frequency, i.e., $p(w)=freq(w)/|D|$, where $freq(w)$ indicates the frequency of w in D and $|D|$ indicates the total number of words in D. The probability of a sentence $s=\{w_1, \ldots, w_N\}$ is then calculated as the product of the word probabilities, i.e., $p(s)=\Pi_i\, p(w_i)$. Moreover, the probability of a summary consisting a set of sentences, denoted as $S=\{s_1, \ldots, s_M\}$, can be calculated by the product of the sentence probabilities, i.e., $p(S)=\Pi_j\, p(s_j)$. To obtain the optimum summary, an intuitive idea is to select the sentences to maximize the overall summary probability $p(S)$, equivalent to maximizing $\log(p(S)) = \Sigma_j\Sigma_i \log(p(w_{ji})) = \Sigma_j\Sigma_i (\log freq(w_{ji}) - \log|D|) = \Sigma_j\Sigma_i \log freq(w_{ji}) - |S|\cdot\log|D|$,
where w_{ji} indicates the ith word in s_j and $|S|$ indicates the total number of words in S. As to practical summarization tasks, a maximum summary length is usually postulated. So here we just assume that the length of the summary is fixed. Then, the above optimization target is equivalent to maximizing $\Sigma_j\Sigma_i \log freq(w_{ji})$. From the view of information theory, the sum can also be interpreted as a simple measure on the total information amount of the summary. In this interpretation, the information of a single word w_{ji} is measured by $\log freq(w_{ji})$ and the summary information is the sum of the word information. So the optimization target can also be interpreted as including the most informative words to form the most informative summary given the length limit.

In extractive summarization, summaries are composed by sentence selection. As to the above optimization target, the sentence scoring function for ranking the sentences should be calculated as the average word information, i.e., $score(s) = \Sigma_i \log freq(w_i) / |s|$.

After ranking the sentences by their ranking scores, we can select the sentences into the summary by the descending order of their score until the length limit is reached. By this process, the summary with the largest $p(S)$ can be composed.

3.2 Word Position Features

With the above model, word position features are defined to represent the word position information and are then incorporated into the model. According to the motivation, the features are defined by the ordinal positions of word appearances, based on the position hypothesis that earlier appearances of a word are more informative. Formally, for the ith appearance among the total n appearances of a word w, four position features are defined based on i and n using different formulas as described below.

(1) Direct proportion (DP) With the word position hypothesis, an intuitive idea is to regard the information degree of the first appearance as 1 and the last one as $1/n$, and then let the degree decrease linearly to the position i. So we can obtain the first position feature defined by the direct proportion function, i.e., $f(i)=(n-i+1)/n$.

(2) Inverse proportion (IP). Besides the linear function, other functions can also be used to characterize the relationship between the position and the importance. The second position feature adopts another widely-used function, the inversed proportion function, i.e., $f(i)=1/i$. This measure is similar to the above one, but the information degree decreases by the inverse proportional function. Therefore, the degree decreases more quickly at smaller positions, which implies a stronger preference for leading sentences.

(3) Geometric sequence (GS). For the third feature, we make an assumption that the degree of every appearance is the sum of the degree of all the following appearances, i.e., $f(i) = f(i+1)+ f(i+2)+\ldots+ f(n)$. It can be easily derived that the sequence also satisfies $f(i) = 2\cdot f(i-1)$. That is, the information degree of each new appearance is

halved. Then the feature value of the ith appearance can be calculated as $f(i) = (1/2)^{i-1}$.

(4) Binary function (BF). The final feature is a binary position feature that regards the first appearance as much more informative than the all the other appearances, i.e., $f(i)=1$, if $i=1$; λ else, where λ is a small positive real number.

3.3 Incorporating the Position Features

To incorporate the position features into the word-based summarization model, we use them to adjust the importance of the word appearance. For the ith appearance of a word w, its original importance is multiplied by the position feature value, i.e., $\log freq(w) \cdot pos(w, i)$, where $pos(w, i)$ is calculated by one of the four position features introduced above. By this, the position feature is also incorporated into the sentence scores, i.e.,

$$score'(s) = \Sigma_i [\log freq(w_i) \cdot pos(w_i)] / |s|$$

3.4 Sentence Position Features

In our study, another type of position features, which model sentence position information, is defined for comparison with the word position features. The sentence position features are also defined by the above four formulas. However, for each appearance, the definition of i and n in the formulas are changed to the ordinal position of the sentence that contains this appearance and the total number of sentences in the document respectively. In fact, the effects of the features defined in this way are equivalent to traditional sentence position features. Since i and n are now defined by sentence positions, the feature values of the word tokens in the same sentence s are all equal. Denote it by $pos(s)$, and the sentence score with the position feature can be written as

$$score'(s) = (\Sigma_{w \text{ in } s} \log freq(w) \cdot pos(s))/|s|$$
$$= pos(s) \cdot (\Sigma \log_{w \text{ in } s} freq(w)/|s|),$$

which can just be viewed as the product of the original score and a sentence position feature.

3.5 Discussion

By using the four functions to measure word or sentence position information, we can generate a total of eight position features. Among the four functions, the importance drops fastest under the binary function and the order is **BF** > **GS** > **IP** > **DP**. Therefore, the features based on the binary function are the most biased to the leading sentences in the document and the features based on the direct proportion function are the least. On the other hand, as mentioned in the introduction, sentence-based features have larger preferences for leading sentences than word-based position features.

An example is given below to illustrate the difference between word and sentence position features. This is a document from DUC 2001.

1. GENERAL ACCIDENT, the leading British insurer, said yesterday that insurance claims arising from Hurricane Andrew could 'cost it as much as Dollars 40m.'
2. Lord Airlie, the chairman who was addressing an extraordinary shareholders' meeting, said: 'On the basis of emerging information, General Accident advise that the losses to their US operations arising from Hurricane Andrew, which struck Florida and Louisiana, might in total reach the level at which external catastrophe reinsurance covers would become exposed'.
3. What this means is that GA is able to pass on its losses to external reinsurers once a certain claims **threshold** has been breached.
4. It believes this **threshold** may be breached in respect of Hurricane Andrew claims.
5. However, if this happens, it would suffer a post-tax loss of Dollars 40m (Pounds 20m).
6. Mr Nelson Robertson, GA's chief general manager, explained later that the company has a 1/2 per cent share of the Florida market.
7. It has a branch in Orlando.
8. The company's loss adjusters are in the area trying to **estimate** the losses.
9. Their guess is that losses to be faced by all insurers may total more than Dollars 8bn.
10. Not all damaged property in the area is insured and there have been **estimates** that the storm caused more than Dollars 20bn of damage.
11. However, other insurers have **estimated** that losses could be as low as Dollars 1bn in total.
12 Mr Robertson said: 'No one knows at this time what the exact loss is'.

For the word "threshold" which appears twice in the document, its original importance is log(2), for the appearance of "threshold" in the 4th sentence, the modified score based on word position feature with the direct proportion function is $1/2 \cdot \log(2)$. In contrast, the score based on sentence position feature with the

same function is 9/12·log(2), which is larger. For the appearance of the word "estimate" in the 8th sentence, its original importance is log(3) (the three boldfaced tokens are regarded as one word with stemming). The word-based and sentence-based scores are log(3) and 5/12·log(3) respectively. So its importance is larger under word position feature. Therefore, the system with word position features may prefer the 8th sentence that is in deeper positions but the system with sentence position feature may prefer the 4th sentence. As for this document, the top 5 sentences selected by sentence position feature are {1, 4, 3, 5, 2} and the those selected by the word position features are {1, 8, 3, 6, 9}. This clearly demonstrates the difference between the position features.

4 Experimental Results

4.1 Experiment Settings

We conduct the experiments on the data sets from the Document Understanding Conference (DUC) run by NIST. The DUC competition started at year 2001 and has successfully evaluated various summarization tasks up to now. In the experiments, we evaluate the effectiveness of position information on several DUC data sets that involve various summarization tasks. One of the evaluation criteria used in DUC, the automatic summarization evaluation package ROUGE, is used to evaluate the effectiveness of the proposed word position features in the context of document summarization[1]. The recall scores of ROUGE-1 and ROUGE-2, which are based on unigram and bigram matching between system summaries and reference summaries, are adopted as the evaluation criteria.

In the data sets used in the experiments, the original documents are all pre-processed by sentence segmentation, stop-word removal and word stemming. Based on the word-based summarization model, a total of nine systems are evaluated in the experiments, including the system with the original ranking model (denoted as **None**), four systems with each word position feature (denoted as **WP**) and four systems with each sentence position feature (denoted as **SP**). For reference, the average ROUGE scores of all the human summarizers and all the submitted systems from the official results of NIST are also given (denoted as **Hum** and **NIST** respectively).

4.2 Redundancy Removal

To reduce the redundancy in the generated summaries, we use an approach similar to the maximum marginal relevance (MMR) approach in the sentence selection process (Carbonell and Goldstein, 1998). In each round of the sentence selection, the candidate sentence is compared against the already-selected sentences. The sentence is added to the summary only if it is not significantly similar to any already-selected sentence, which is judged by the condition that the cosine similarity between the two sentences is less than 0.7.

4.3 Generic Summarization

In the first experiment, we use the DUC 2001 data set for generic single-document summarization and the DUC 2004 data set for generic multi-document summarization. The DUC 2001 data set contains 303 document-summary pairs; the DUC 2004 data set contains 45 document sets, with each set consisting of 10 documents. A summary is required for each document set. Here we need to adjust the ranking model for the multi-document task, i.e., the importance of a word is calculated as its total frequency in the whole document set instead of a single document. For both tasks, the summary length limit is 100 words.

Table 1 and 2 below provide the average ROUGE-1 and ROUGE-2 scores (denoted as **R-1** and **R-2**) of all the systems. Moreover, we used paired two sample t-test to calculate the significance of the differences between a pair of word and sentence position features. The bolded score in the tables indicates that that score is significantly better than the corresponding paired one. For example, in Table 1, the bolded **R-1** score of system **WP DP** means that it is significantly better than the **R-1** score of system **SP DP**. Besides the ROUGE scores, two statistics, the number of "first sentences[2]" among the selected sentences (**FS-N**) and the

[1] We run ROUGE-1.5.5 with the parameters "-x -m -n 2 -2 4 -u -c 95 -p 0.5 -t 0"

[2] A "first sentence" is the sentence at the fist position of a document.

average position of the selected sentences (**A-SP**), are also reported in the tables for analysis.

System	R-1	R-2	FS-N	A-SP
WP DP	**0.4473**	**0.1942**	301	4.00
SP DP	0.4396	0.1844	300	3.69
WP IP	**0.4543**	**0.2023**	290	4.30
SP IP	0.4502	0.1964	303	3.08
WP GS	**0.4544**	**0.2041**	278	4.50
SP GS	0.4509	0.1974	303	2.93
WP BF	**0.4544**	**0.2036**	253	5.57
SP BF	0.4239	0.1668	303	9.64
None	0.4193	0.1626	265	10.06
NIST	0.4445	0.1865	-	-
Hum	0.4568	0.1740	-	-

Table 1. Results on the DUC 2001 data set

System	R-1	R-2	FS-N	A-SP
WP DP	0.3728	0.0911	89	4.16
SP DP	0.3724	0.0908	112	2.68
WP IP	**0.3756**	0.0912	108	3.77
SP IP	0.3690	0.0905	201	1.01
WP GS	**0.3751**	0.0916	110	3.67
SP GS	0.3690	0.0905	201	1.01
WP BF	**0.3740**	**0.0926**	127	3.14
SP BF	0.3685	0.0903	203	1
None	0.3550	0.0745	36	10.98
NIST	0.3340	0.0686	-	-
Hum	0.4002	0.0962	-	-

Table 2. Results on the DUC 2004 data set

From Table 1 and Table 2, it is observed that position information is indeed very effective in generic summarization so that all the systems with position features performed better than the system **None** which does not use any position information. Moreover, it is also clear that the proposed word position features consistently outperform the corresponding sentence position features. Though the gaps between the ROUGE scores are not large, the t-tests proved that word position features are significantly better on the DUC 2001 data set. On the other hand, the advantages of word position features over sentence position features are less significant on the DUC 2004 data set. One reason may be that the multiple documents have provided more candidate sentences for composing the summary. Thus it is possible to generate a good summary only from the leading sentences in the documents. According to Table 2, the average-sentence-position of system **SP BF** is 1, which means that all the selected sentences are "first sentences". Even under this extreme condition, the performance is not much worse.

The two statistics also show the different preferences of the features. Compared to word position features, sentence position features are likely to select more "first sentences" and also have smaller average-sentence-positions. The abnormally large average-sentence-position of **SP BF** in DUC 2001 is because it does not differentiate all the other sentences except the first one. The corresponding word-position-based system **WP BF** can differentiate the sentences since it is based on word positions, so its average-sentence-position is not that large.

4.4 Query-focused Summarization

Since year 2005, DUC has adopted query-focused multi-document summarization tasks that require creating a summary from a set of documents to a given query. This task has been specified as the main evaluation task over three years (2005-2007). The data set of each year contains about 50 DUC topics, with each topic including 25-50 documents and a query. In this experiment, we adjust the calculation of the word importance again for the query-focused issue. It is changed to the total number of the appearances that fall into the sentences with at least one word in the query. Formally, given the query which is viewed as a set of words $Q=\{w_1, \ldots, w_T\}$, a sentence set S_Q is defined as the set of sentences that contain at least one w_i in Q. Then the importance of a word w is calculated by its frequency in S_Q. For the query-focused task, the summary length limit is 250 words.

Table 3 below provides the average ROUGE-1 and ROUGE-2 scores of all the systems on the DUC 2005-2007 data sets. The boldfaced terms in the tables indicate the best results in each column. According to the results, on query-focused summarization, position information seems to be not as effective as on generic summarization. The systems with position features can not outperform the system **None**. In fact, this is reasonable due to the requirement specified by the pre-defined query. Given the query, the content of interest may be in any

position of the document and thus the position information becomes less meaningful.

On the other hand, we find that though the systems with word position features cannot outperform the system **None**, it does significantly outperform the systems with sentence position features. This is also due to the role of the query. Since it may refer to the specified content in any position of the documents, sentence position features are more likely to fail in discovering the desired sentences since they always prefer leading sentences. In contrast, word position features are less sensitive to this problem and thus perform better. Similarly, we can see that the direct proportion (**DP**), which has the least bias for leading sentences, has the best performance among the four functions.

System	2005		2006		2007	
	R-1	R-2	R-1	R-2	R-1	R-2
WP DP	**0.3791**	**0.0805**	0.3909	0.0917	0.4158	0.1135
SP DP	0.3727	0.0776	0.3832	0.0869	0.4118	0.1103
WP IP	0.3772	0.0791	0.3830	0.0886	0.4106	0.1121
SP IP	0.3618	0.0715	0.3590	0.0739	0.3909	0.1027
WP GS	0.3767	0.0794	0.3836	0.0879	0.4109	0.1119
SP GS	0.3616	0.0716	0.3590	0.0739	0.3909	0.1027
WP BF	0.3740	0.0741	0.3642	0.0796	0.3962	0.1037
SP BF	0.3647	0.0686	0.3547	0.0742	0.3852	0.1013
NONE	0.3788	0.0791	**0.3936**	**0.0924**	**0.4193**	**0.1140**
NIST	0.3353	0.0592	0.3707	0.0741	0.0962	0.3978
Hum	0.4392	0.1022	0.4532	0.1101	0.4757	0.1402

Table 3. Results on the DUC 2005 - 2007 data sets

System	2008 A		2008 B		2009 A		2009 B	
	R-1	R-2	R-1	R-2	R-1	R-2	R-1	R-2
WP DP	0.3687	0.0978	**0.3758**	0.1036	0.3759	0.1015	0.3693	0.0922
SP DP	0.3687	0.0971	0.3723	0.1011	0.3763	0.1031	0.3704	0.0946
WP IP	**0.3709**	**0.1014**	0.3741	**0.1058**	0.3758	0.1030	0.3723	0.0906
SP IP	0.3619	0.0975	0.3723	0.1037	0.3693	0.0994	0.3690	**0.0956**
WP GS	0.3705	0.1004	0.3732	0.1048	**0.3770**	0.1051	**0.3731**	0.0917
SP GS	0.3625	0.0975	0.3723	0.1037	0.3693	0.0994	0.3690	**0.0956**
WP BF	0.3661	0.0975	0.3678	0.0992	0.3720	**0.1069**	0.3650	0.0936
SP BF	0.3658	0.0965	0.3674	0.0980	0.3683	0.1043	0.3654	0.0945
NONE	0.3697	0.0978	0.3656	0.0915	0.3653	0.0934	0.3595	0.0834
NIST	0.3389	0.0799	0.3192	0.0676	0.3468	0.0890	0.3315	0.0761
Hum	0.4105	0.1156	0.3948	0.1134	0.4235	0.1249	0.3901	0.1059

Table 4. Results on the TAC 2008 - 2009 data sets

4.5 Update Summarization

Since year 2008, the DUC summarization track has become a part of the Text Analysis Conference (TAC). In the update summarization task, each document set is divided into two ordered sets A and B. The summarization target on set A is the same as the query-focused task in DUC 2005-2007. As to the set B, the target is to write an update summary of the documents in set B, under the assumption that the reader has already read the documents in set A. The data set of each year contains about 50 topics, and each topic includes 10 documents for set A, 10 documents for set B and an additional query. For set A, we follow exactly the same method used in section 4.4; for set B, we make an additional novelty check for the sentences in B with the MMR approach. Each candidate sentence for set B is now compared to both the selected sentences in set B and in set A to

ensure its novelty. In the update task, the summary length limit is 100 words.

Table 4 above provides the average ROUGE-1 and ROUGE-2 scores of all the systems on the TAC 2008-2009 data sets. The results on set A and set B are shown individually. For the task on set A which is almost the same as the DUC 2005-2007 tasks, the results are also very similar. A small difference is that the systems with position features perform slightly better than the system **None** on these two data sets. Also, the difference between word position features and sentence position features becomes smaller. One reason may be that the shorter summary length increases the chance of generating good summaries only from the leading sentences. This is somewhat similar to the results reported in (Nenkova, 2005) that position information is more effective for short summaries.

For the update set B, the results show that position information is indeed very effective. In the results, all the systems with position features significantly outperform the system **None**. We attribute the reason to the fact that we are more concerned with novel information when summarizing update set B. Therefore, the effect of the query is less on set B, which means that the effect of position information may be more pronounced in contrast. On the other hand, when comparing the position features, we can see that though the difference of the position features is quite small, word position features are still better in most cases.

4.6 Discussion

Based on the experiments, we briefly conclude the effectiveness of position information in document summarization. In different tasks, the effectiveness varies indeed. It depends on whether the given task has a preference for the sentences at particular positions. Generally, in generic summarization, the position hypothesis works well and thus the ordinal position information is effective. In this case, those position features that are more distinctive, such as **GS** and **BF**, can achieve better performances. In contrast, in the query-focused task that relates to specified content in the documents, ordinal position information is not so useful. Therefore, the more distinctive a position feature is, the worse performance it leads to. However, in the update summarization task that also involves queries, position information becomes effective again since the role of the query is less dominant on the update document set.

On the other hand, by comparing the sentence position features and word position features on all the data sets, we can draw an overall conclusion that word position features are consistently more appreciated. For both generic tasks in which position information is effective and query-focused tasks in which it is not so effective, word position features show their advantages over sentence position features. This is because of the looser position hypothesis postulated by them. By avoiding arbitrarily regarding the leading sentences as more important, they are more adaptive to different tasks and data sets.

5 Conclusion and Future Work

In this paper, we proposed a novel kind of word position features which consider the positions of word appearances instead of sentence positions. The word position features were compared to sentence position features under the proposed sentence ranking model. From the results on a series of DUC data sets, we drew the conclusion that the word position features are more effective and adaptive than traditional sentence position features. Moreover, we also discussed the effectiveness of position information in different summarization tasks.

In our future work, we'd like to conduct more detailed analysis on position information. Besides the ordinal positions, more kinds of position information can be considered to better model the document structures. Moreover, since position hypothesis is not always correct in all documents, we'd also like to consider a pre-classification method, aiming at identifying the documents for which position information is more suitable.

Acknowledgement The work described in this paper was supported by Hong Kong RGC Projects (PolyU5217/07E). We are grateful to professor Chu-Ren Huang for his insightful suggestions and discussions with us.

References

Edmundson, H. P.. 1969. *New methods in automatic Extracting*. Journal of the ACM, volume 16, issue 2, pp 264-285.

Gillick, D., Favre, B., Hakkani-Tur, D., Bohnet, B., Liu, Y., Xie, S.. 2009. *The ICSI/UTD Summarization System at TAC 2009*. Proceedings of Text Analysis Conference 2009.

Jaime G. Carbonell and Jade Goldstein. 1998. *The use of MMR, diversity-based reranking for reordering documents and producing summaries*. Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval, pp 335-336.

Lin, C. and Hovy, E.. 1997. *Identifying Topics by Position*. Proceedings of the fifth conference on Applied natural language processing 1997, pp 283-290.

Luhn, H. P.. 1958. *The automatic creation of literature abstracts*. IBM J. Res. Develop. 2, 2, pp 159-165.

Nenkova. 2005. *Automatic text summarization of newswire: lessons learned from the document understanding conference*. Proceedings of the 20th National Conference on Artificial Intelligence, pp 1436-1441.

Ouyang, Y., Li, S., Li, W.. 2007. *Developing learning strategies for topic-based summarization*. Proceedings of the sixteenth ACM conference on Conference on information and knowledge management, pp 79-86.

Radev, D., Jing, H., Sty´s, M. and Tam, D.. 2004. *Centroid-based summarization of multiple documents*. Information Processing and Management, volume 40, pp 919–938.

Schilder, F., Kondadadi, R.. 2008. *FastSum: fast and accurate query-based multi-document summarization*. Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies, short paper session, pp 205-208.

Toutanova, K. et al. 2007. *The PYTHY summarization system: Microsoft research at DUC 2007*. Proceedings of Document Understanding Conference 2007.

Evaluating FrameNet-style semantic parsing: the role of coverage gaps in FrameNet

Alexis Palmer and Caroline Sporleder
Computational Linguistics
Saarland University
{apalmer, csporled}@coli.uni-saarland.de

Abstract

Supervised semantic role labeling (SRL) systems are generally claimed to have accuracies in the range of 80% and higher (Erk and Padó, 2006). These numbers, though, are the result of highly-restricted evaluations, i.e., typically evaluating on hand-picked lemmas for which training data is available. In this paper we consider performance of such systems when we evaluate at the document level rather than on the lemma level. While it is well-known that coverage gaps exist in the resources available for training supervised SRL systems, what we have been lacking until now is an understanding of the precise nature of this coverage problem and its impact on the performance of SRL systems. We present a typology of five different types of coverage gaps in FrameNet. We then analyze the impact of the coverage gaps on performance of a supervised semantic role labeling system on full texts, showing an average oracle upper bound of 46.8%.

1 Introduction

A lot of progress has been made in semantic role labeling over the past years, but the performance of state-of-the-art systems is still relatively low, especially for deep, FrameNet-style semantic parsing. Furthermore, many of the reported performance figures are somewhat unrealistic because system performance is evaluated on hand-selected lemmas, usually under the implicit assumptions that (i) all relevant word senses (frames) of each lemma are known, and (ii) there is a suitable amount of training data for each sense. This approach to evaluation arises from the limited coverage of the available hand-coded data against which to evaluate. More realistic evaluations test systems on full text, but these same coverage limitations mean that the assumptions made in more restricted evaluations do not necessarily hold for full text. This paper provides an analysis of the extent and nature of the coverage gaps in FrameNet. A more precise understanding of the limitations of existing resources with respect to robust semantic analysis of texts is an important foundational component both for improving existing systems and for developing future systems, and it is in this spirit that we make our analysis.

Full-text semantic analysis

Automated frame-semantic analysis aims to extract from text the key event-denoting predicates and the semantic argument structure for those predicates. The semantic argument structure of a predicate describing an event encodes relationships between the participants involved in the event, e.g. who did what to whom. Knowledge of semantic argument structure is essential for language understanding and thus important for applications such as information extraction (Moschitti et al., 2003; Surdeanu et al., 2003), question answering (Shen and Lapata, 2007), or recognizing textual entailment (Burchardt et al., 2009). Evaluating an existing system for its ability to aid such tasks is unrealistic if the evaluation is lemma-based rather than text-based. Consequently, there continues to be significant interest in developing semantic role labeling (SRL) systems able to automatically compute the semantic argument structures in an input text.

Performance on the full text task, though, is typically much lower than for the more restricted evaluations. The SemEval 2007 Task on "Frame Semantic Structure Extraction," for example, required systems to identify key predicates in texts,

Coling 2008: Poster Volume, pages 928–936,
Beijing, August 2010

assign a semantic frame to the relevant predicates, identify the semantic arguments for the predicates, and finally label those arguments with their semantic roles. The systems participating in this task only obtained F-Scores between 55% and 78% for frame assignment, despite the fact that the task organizers adopted a lenient evaluation scheme which gave partial credit for near-misses (Baker et al., 2007). For the combined task of frame assigment and role labeling the performance was even lower, ranging from 35% to 54% F-Score.

Note that this distinction between evaluation schemes for SRL systems corresponds to the distinction between "lexical sample" and "all words" evaluations in word sense disambiguation, where results for the latter scheme are also typically lower (McCarthy, 2009).

The low performances are at least partly due to coverage problems. For example, Baker et al. (2007) annotated three new texts for their SemEval 2007 task. Although these new texts overlap in domain with existing FrameNet data, the task organizers had to create 40 new frames in order to complete annotation. The new frames were for word senses found in the test set but missing from FrameNet. The test set contained only 272 frames (types), meaning that nearly 15% of the frames therein were not yet defined in FrameNet. Obviously, coverage issues of this degree make full SRL a difficult task, but this is a realistic scenario that will be encountered in real applications as well.

As mentioned above, for many tasks it is necessary to compute the semantic argument structures for the whole text, or at least for multi-sentence passages. Due to non-local relations between argument structures this is also true for tasks like question answering, where it might be possible to automatically determine a subset of lemmas which are relevant for the task. For example, in (1) it might be possible to determine that the second sentence contains the answer to the question "*Was Thomas Preston acquitted of theft?*" However, to correctly answer this question, it is necessary to resolve the null instantiation of the CHARGES role of the VERDICT frame. This null instantiation links back to the previous sentence, and resolving it might require obtaining an analysis of the word *tried*.

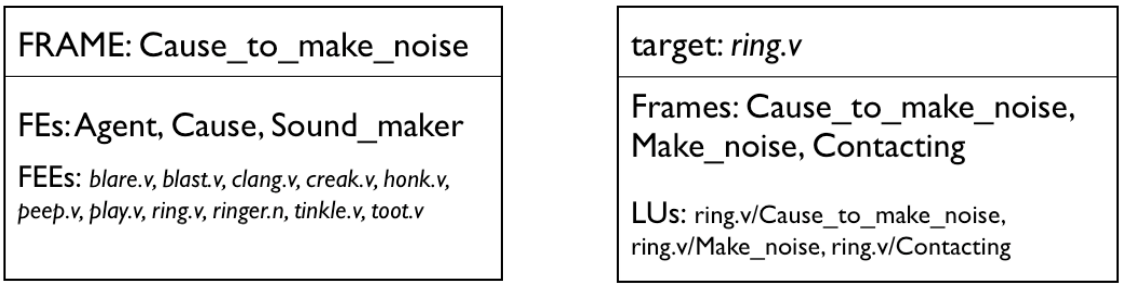
(1) [Captain Thomas Preston]$_{Defendant_i}$ was **tried**$_{Try\_defendant_i}$ for [murder]$_{Charges_{i,j}}$.

In the end [he]$_{Defendant_j}$ was **acquitted**$_{Verdict_j}$ [Ø]$_{Charges_j}$.

Performance levels obtained for full text are usually not sufficient for this kind of real-world task. FrameNet-style semantic role labeling has been shown to, in principle, be beneficial for applications that need to generalise over individual lemmas, such as recognizing textual entailment or question answering. However, studies also found that state-of-the-art FrameNet-style SRL systems perform too poorly to provide any substantial benefit to real applications (Burchardt et al., 2009; Shen and Lapata, 2007).

Extending the value of automated semantic parsing for a variety of applications requires improving the ability of systems to process unrestricted text. Several methods have been proposed to address different aspects of the coverage problem, ranging from automatic data expansion and semi-supervised semantic role labelling (Fürstenau and Lapata, 2009b; Fürstenau and Lapata, 2009a; Deschacht and Moens, 2009; Gordon and Swanson, 2007; Padó et al., 2008) to systems which can infer missing word senses (Pennacchiotti et al., 2008b; Pennacchiotti et al., 2008a; Cao et al., 2008; Burchardt et al., 2005). However, so far there has not been a detailed analysis of the problem. In this paper we provide that detailed analysis, by defining different types of coverage problems and performing analysis of both coverage and performance of an automated SRL system on three different data sets.

Section 2 of the paper provides an introduction to FrameNet and introduces the basic terminology. Section 4 describes our approach to coverage evaluation, Section 3 discusses the texts analyzed, and the analysis itself appears in Section 5. Section 6 then looks at one possibility for addressing the coverage problem. The final section presents some discussion and conclusions.



(a) (b)

Figure 1: Terminology: (a) Frame with core frame elements (FEs) and frame-evoking elements (FEEs) (b) Target with possible frame assignments and resultant lexical units (LUs)

2 FrameNet

Manual annotation of corpora with semantic argument structure information has enabled the development of statistical and supervised machine learning techniques for semantic role labeling (Toutanova et al., 2008; Moschitti et al., 2008; Gildea and Jurafsky, 2002).

The two main resources are PropBank (Palmer et al., 2005) and FrameNet (Ruppenhofer et al., 2006). PropBank aims to provide a semantic role annotation for every verb in the Penn TreeBank (Marcus et al., 1994) and assigns roles on a verb-by-verb basis, without making higher-level generalizations. Whether two distinct usages of a given verb are viewed as different senses or not is thus driven by both syntax (namely, differences in syntactic argument structure) and semantics (via basic, easily-discernable differences in meaning).

FrameNet[1] is a lexicographic project whose aim it is to create a lexical resource documenting valence structures for different word senses and their possible mappings to underlying semantic argument structure (Ruppenhofer et al., 2006). In contrast to PropBank, FrameNet is primarily semantically driven; word senses (*frames*)[2] are defined mainly based on sometimes-subtle meaning differences and can thus generalise across individual lemmas, and often also across different parts-of-speech. Because FrameNet focusses on semantics it is not restricted to verbs but also provides semantic argument annotations for nouns, adjectives, adverbs, prepositions and even multi-word expressions. For example, the sentence in (2) and the NP in (3) have identical argument structures because the verb *speak* and the noun *comment* evoke the same frame STATEMENT.

(2) [The politician]$_{Speaker}$ **spoke**$_{Statement}$ [about recent developments on the labour market]$_{Topic}$.

(3) [The politician's]$_{Speaker}$ **comments**$_{Statement}$ [on recent developments on the labour market]$_{Topic}$

Since FrameNet annotations are semantically driven they are considerably more time-consuming to create than PropBank annotations. However, FrameNet also provides 'deeper' and more informative annotations than PropBank analyses (Ellsworth et al., 2004). For instance, the fact that (2) and (3) refer to the same state-of-affairs is not captured by PropBank sense distinctions.

FrameNet Terminology

The English FrameNet data consist of an inventory of frames (i.e. word senses), a set of lexical entries, and a set of annotated examples exemplifying different syntactic realizations for selected frames (known as the *lexicographic annotations*). **Frames** are conceptual structures that describe types of situations or events together with their participants. **Frame-evoking elements (FEEs)** are predicate usages which evoke a particular frame. A given lemma can evoke different

[1] http://framenet.icsi.berkeley.edu/

[2] We follow Erk (2005) in treating frame assignment as a word sense disambiguation task. Thus in this paper we use the terms *frame* and *sense* interchangeably.

frames in different contexts; each instance of the lemma is a separate **target** for semantic analysis. For example, (4) and (5) illustrate two different frames of the lemma *speak*.

(4) [The politician]$_{Speaker}$ **spoke**$_{Statement}$ [about recent developments on the labour market]$_{Topic}$.

(5) [She]$_{Interlocutor_1}$ doesn't **speak**$_{Chatting}$ to [anyone]$_{Interlocutor_2}$.

In this paper we follow standard use of FrameNet terminology, with the possible exception of the term *lexical unit*. Figure 1 illustrates our use of FrameNet-related terminology, focussing on (a) the CAUSE_TO_MAKE_NOISE frame and (b) the target verb lemma *ring*.

The definition of a frame determines the available roles (**frame elements** or **FEs**) of the semantic argument structure for the particular use of the predicate, as well as the status—core or peripheral—of those roles. For example, the FE TOPIC is a core role under the STATEMENT frame, but a peripheral role under the CHATTING frame.

The lexical entry of a lemma in FrameNet specifies a list of frames which the lemma can evoke, and the pairing of a word with a particular frame is called a **lexical unit (LU)**. Ideally there should be annotated examples for each lexical unit, exemplifying different syntactic constructions which can realize this LU. However, as we will see later (Section 5) annotated examples can be missing. Also, because FrameNet is a lexicographic project, the examples were extracted to illustrate particular usages, i.e., they are not meant to be statistically representative.

3 Data

Having introduced the basic FrameNet terminology, we now describe in more detail the data sets used in the analysis. FrameNet Release 1.3 (FN1.3), the latest release from the Berkeley FrameNet project, includes both a corpus of lexicographic annotations (FNL), which we referred to in Section 2, and a corpus of texts fully-annotated with frames and semantic role labels (FNF). Annotations in the two corpora of course cover different sets of predicates and frames, and FNL is the corpus commonly used as the basis for training supervised FrameNet-based SRL systems (Erk and Padó, 2006).

In our analysis, we look at three data sets: the lexicographic annotations from FN1.3, the full text annotations from FN1.3, and a new data set of running text that was annotated for the SemEval 2010 Task-10 (see Table 1 for details).

FrameNet Lexicographic (FNL) FrameNet started as a lexicographic project, aiming to draw up an inventory of frames and lexical units, supported by corpus evidence, to document the range of syntactic and semantic usages of each lexical unit. The annotated example sentences in this part of FN1.3 are taken from the British National Corpus (BNC). BNC is a balanced corpus, hence FNL covers, in principle, a variety of domains.

For each LU, a subset of the sentences in which it occurs was selected for annotation, and in each extracted sentence, only the target LU was annotated. The sentences were not chosen randomly but with a set of lexicographic constraints in mind. In particular the sentences should exemplify different usage. Thus ideally selected sentences would be easy to understand and not too long or complex. As a consequence of this linguistically-driven selection procedure, the annotated sentences are not statistically representative in any way. FNL provides annotations for just under 140,000 FEEs (tokens). On average, around 20 sentences are annotated for each LU. FrameNet's frame inventory contains 722 frames.[3]

FrameNet Full Texts (FNF) Starting with release 1.3, FrameNet also provides annotations of running texts. In this annotation mode, all LUs in a sentence and all sentences in a text are annotated. FN1.3 contains two subsets of full text annotations. The first of these (**PB**) contains five texts which were also annotated by the PropBank project. While all texts come from the Wall Street Journal, they are not prototypical examples of the financial domain, rather they are longer essays covering a wide variety of general interest topics

[3] Only lexical frames are included in this number. In addition to those, FrameNet 1.3 defines another 74 frames which cannot be lexicalised but are included because they provide useful generalisations in the frame hierarchy.

Data	Genre / Domain	FEEs Tokens	FEEs Types	Frames Types
FNL	mixed	139,439	8370	722
PB	essays, general interest	1580	680	319
NTI	reports, foreign affairs	8271	1305	434
SE	fiction, crime	1530	680	320

Table 1: Statistics for the three data sets

(ranging from 'Bell Ringing' to 'Earthquakes'). The second subset (**NTI**) contains 12 texts from the Nuclear Threat Initiative website.[4] These texts are intelligence reports which summarize and discuss the status of various countries with regard to the development of weapons and missile systems. Statistics for both data sets are given in Table 1.

SemEval 2010 Task-10 Full Texts (SE) While the FrameNet full texts allow us to estimate coverage gaps that arise from limited training data, they do not allow us to gauge coverage problems arising from missing frames in the FN1.3 inventory. The reason for this is that the frame inventory reflects the annotations of both the lexicographic and the full text part of FN1.3, i.e., every frame annotated in one of these subsets will also be part of the inventory. To estimate the frame coverage problem on completely new texts, we therefore included a third (full text) data set that was annotated for the SemEval 2010 Task 10 on "Linking Events and Their Participants in Discourse" (Ruppenhofer et al., 2009).[5] The text is taken from Arthur Conan Doyle's "The Adventure of Wisteria Lodge". It thus comes from the fiction domain.

The text was manually annotated with frame-semantic argument structure by two experienced annotators. Similar to the FNF texts, the annotators aimed to annotate all LUs in the text. To do so, some new frames had to be created for previously un-encountered LUs. These new frames are not part of FN1.3 and we can thus use them to estimate coverage problems arising from missing frames. Details for the data set can be found in Table 1. This data set is very similar to the PB set in terms of size, FEE type-token ratio and number of frames (types).

[4] http://www.nti.org

[5] The data set is available from http://semeval2.fbk.eu/semeval2.php?location=data.

4 Types of Coverage Gaps

Semantic role labelling systems have to perform two sub-tasks: (i) identifying the correct frame for a given lemma and context, and (ii) identifying and labeling the frame elements. The most severe coverage problems typically arise with the first subtask. Furthermore, coverage problems related to frame identification have a knock-on effect on role identification and labeling because the choice of the correct frame determines which roles are available. Therefore, we focus on the frame identification task in this paper.

Attempts to do automated frame assignment on unrestricted text invariably encounter problems associated with limited coverage of frame-evoking elements in FrameNet. However, not every coverage gap is the same, and the precise nature of a coverage gap influences potential strategies for addressing it. In this section we describe the different types of coverage gaps. We proceed from less problematic coverage gaps to more problematic ones, in the sense that the former can be addressed more straighforwardly by automated systems than can the latter.

4.1 NOTR gaps

Some coverage gaps occur when lexical units (LUs) defined in FrameNet lack corresponding annotated examples; these gaps are the result of lacking training data, hence we call them **NOTR** gaps. To give a sense of the abundance of such gaps, of the 10,191 LUs defined in FN1.3, annotated examples are available for only 6727.

NOTR-LU: lexical unit with no training data. In many cases, an LU — a specific pairing of a target lemma with one frame — may be defined in FrameNet, thus potentially accessible to an automated system, but lacking labeled training material. For example, FrameNet defines two LUs for the noun *ringer*: with the frames CAUSE TO MAKE NOISE and SIMILARITY. It is clear that the occurrence of *ringer* in (6) belongs to the former LU, even given a very limited context. The lexicographic annotations, though, provide training material only for the SIMILARITY frame.

(6) Then, at a signal, the **ringers** begin varying the order in which the bells sound without altering the steady rhythm of the striking.

NOTR-LU gaps pose particular problems to a fully-supervised SRL system, because such a system cannot learn anything about the context in which the CAUSE TO MAKE NOISE frame is more appropriate. A NOTR-LU gap is identified for an LU even if training data is available for other senses (i.e. other LUs) of the target lemma.

NOTR-TGT: target with no training data. In other cases, a target lemma may be defined as participating in one or more LUs, but with no training data available for *any* of them. In other words, a supervised automated system trained only on the available annotated examples will fail to learn any potential frame assignments for the target lemma. Such is the case for *art*, which in FrameNet is assigned the single frame CRAFT, but for which FNL contains no training data.

(7) The **art** of change-ringing is peculiar to the English, and, like most English peculiarities, unintelligible to the rest of the world.

Whereas a NOTR-LU gap obscures a particular frame assignment for a target lemma, a NOTR-TGT gap indicates a complete absence in the lexicographic corpus of annotated data for the lemma.

4.2 UNDEF gaps

The previous coverage problems arise from a lack of annotated data, an issue which conceivably could be addressed through further annotation. More serious problems arise when a text contains word senses, words, or frames not contained in FrameNet. We call such elements 'undefined'; specifically, they receive no treatment in FN1.3.

UNDEF-LU: lexical unit not defined. Coverage gaps of this sort occur when the frame inventory for a given lemma is not complete. In other words, at least one LU for the lemma exists in FrameNet, but one or more other LUs are missing. For example, the noun *installation* occurs in FrameNet with the frames LOCALE BY USE and INSTALLING. The sense of an art installation, which is an instance of the frame PHYSICAL ARTWORKS, is missing.

UNDEF-TGT: target not addressed. In the worst case, all LUs for a target lemma might be missing, i.e., the lemma does not occur in the FrameNet lexicon at all. The noun *fabric* is an example. Though it has at least two distinct senses—that of cloth or material and that of a framework (e.g. *the fabric of society*)—FrameNet provides no help for determining appropriate frames for instances of this lemma.

UNDEF-FR: frame not defined. Finally, it may be not only that the LU is missing, but that there is no definition in FrameNet for the correct frame given the context. For example, in the sports domain the lemma *ringer* can have the sense of (*a horseshoe thrown so that it encircles the peg*); to our knowledge, this sense is not available in FrameNet.

5 Coverage gaps and automated processing

With the exception of work on extending coverage, most FrameNet-style semantic role labeling studies draw both training and evaluation data from FNL. This is an unrealistic evaluation scenario for full-text semantic analysis, as such evaluation limits the domain for which prediction can occur to those lexical entries treated in FNL. For systems which do not attempt any generalization beyond those lexical entries with training data, this limits the system to 5864 lemmas for which it can make predictions regarding frame assignment and role labeling.

Disregarding whether annotations have yet been provided for the lexical units in FNL still limits us to 8370 frame-evoking elements (targets). To better understand the potential of current frame-semantic resources for semantic analysis of unrestricted text, we evaluate coverage of the FNL annotations against the texts in FNF, as well as against the SemEval text. We then analyze the performance of an off-the-shelf, supervised SRL system, Shalmaneser (Erk and Padó, 2006), on the same texts, with a focus on the types of

Dataset	TR-LU	NOTR-LU	NOTR-TGT	UNDEF-LU	UNDEF-FR
PB	42.66	9.56	47.78	–	–
NTI	46.77	7.77	45.46	–	–
SE	51.64	6.86	26.01	3.40	12.09

Table 2: FrameNet coverage for analyzed texts

errors made and the upper bound on performance for this system.

5.1 FrameNet coverage

As described in Section 4, in many cases a lexical unit, a frame-evoking element, or a frame may simply not be represented in FrameNet. In other cases, the entity may be in FN1.3 but lacking training data. Of the 722 frames defined in FN1.3, for example, annotations exist for 502.

For the three data sets analyzed, Table 2 shows the degree of coverage provided by FNL for the gold-standard frame annotations. First, the TR-LU column shows the non-problematic cases, for which the correct frame annotation is available in FrameNet, with training data. The next two columns represent training gaps related to lack of training data: NOTR-LU are cases for which training data exists for the target, but not for the correct sense of the target, and NOTR-TGT instances are those for which no training data at all exists for the target.

Because all targets annotated in the FNF texts (i.e. PB and NTI above) are incorporated in FN1.3, gaps due to missing LUs, targets, or frames do not exist for those texts. The same does not hold for the SemEval (SE) text. For 3.4% of the annotated SemEval targets, an LU is entirely missing from the lemma's frame inventory in FrameNet, and in just over 12% of cases both the lemma and the frame are missing. In total, more than 15% of LUs appearing in the gold-standard SemEval annotations are not defined at all within FrameNet. This figure accords with that found by Baker et al. (2007).

5.2 Error analysis of full-text frame assignment

Here we examine the errors made by Shalmaneser for frame assignment on the three data sets. The upper bound on apparent performance is fixed by the number of targets for which Shalmaneser has seen training data, namely the sum of TR-LU and NOTR-LU in Table 2.[6]

Dataset	Correct	Type(i)	Type(ii)	Type(iii)
PB	36.71	5.95	9.56	47.78
NTI	41.22	5.55	7.77	45.46
SE	46.67	4.97	6.86	41.50

Table 3: Shalmaneser performance on texts

We consider three categories of errors: (i) *normal or true errors* are misclassifications when the correct label has been seen in the training data. In this category we also count errors resulting from incorrect lemmatization. (ii) *label-not-seen errors* are misclassifications when the correct label does not appear in the training data and thus is unavailable to the classifier. Finally, (iii) *no-chance errors* occur when the system has no information for either a given target or a given frame. Table 3 shows the prevalence of each error type for each data set, given as the percentage of all frame-assignment targets.

It can be seen that the frame assignment accuracy is relatively low for all three texts (between 37% and 47%). However, only a relatively small proportion of the misclassifications are due to true errors made by the system. Furthermore, a large amount of errors (41% to 48%, with an average of 46.8%) is due to cases where important information is missing from FrameNet (Type (iii) errors). Consequently, improving the semantic role labeller by optimising the feature space or the machine learning framework is going to have very little effect. A much more promising path would be to investigate methods which might enable the SRL system to deal gracefully with unseen data. One possible strategy is discussed in the next section.

[6] By 'apparent performance' we mean the system's own evaluation of its accuracy on frame assignment.

6 Frame and lemma overlap

One potential strategy for improving full-text semantic analysis without performing additional annotation is to take advantage of semantic overlap as it is represented in FrameNet. We can look at two different types of overlap in FrameNet: **lemma overlap** and **frame overlap**.

6.1 Lemma overlap

The approach of treating frame assignment as a word sense disambiguation task (as, e.g., by Shalmaneser) relies on the overlap of LUs with the same lemma and trains lemma-based classifiers on all training instances for all LUs involving that lemma. One way to consider using labeled material in FrameNet to improve performance on targets for which we have no labeled material is to generalize over lemmas associated with the same frame. The idea is to use training instances from related lemmas to build a larger training set for lemmas with little or no annotated data.

Of the 8370 lemmas in FN, 8358 share a single frame with at least one other lemma. 890 overlap on two frames with at least one other lemma, and 111 have 3-frame overlap with at least one other lemma. Only 16 lemmas show an overlap of four or more frames. These groupings are:

```
1. clang.v, clatter.v, click.v, thump.v
2. hit.v, smack.v, swing.v, turn.v
3. drop.v, rise.v
4. remember.v, forget.v
5. examine.v, examination.n
6. withdraw.v, withdrawal.n
```

The first two groupings are sets of words that are closely semantically related, the second two are opposite pairs, and the third two are verb-nominalization pairs.

The lemma overlap groups differ with respect to how much training data they make accessible.

6.2 Frame overlap

Another possibility to be considered is generalization over all instances of a given frame. For the 502 frames with annotated examples, the number of annotated instances ranges from one (SAFE SITUATION, BOARD VEHICLE, and ACTIVITY START to 6233 (SELF MOTION), with an average of 278 training instances per frame.

In future work we will examine the effectiveness of binary frame-based classifiers, abstracting away from individual predicates to predict whether a given lemma belongs to the frame in question (for a related study see Johansson and Nugues (2007)). A potential drawback to this approach is the loss of predicate-specific information. We know, for example, about verbs that they tend to have typical argument structures and typical syntactic realizations of those argument structures.

In addition to this frame-overlap approach, we will consider the impact on coverage of using coarser-grained versions of FrameNet in which frames have been merged according to frame relations defined over the FrameNet hierarchy, using the FrameNet Transformer tool described in (Ruppenhofer et al., 2010).

7 Conclusions

Although it is clear that the capability to do shallow semantic analysis on unrestricted text, and on complete documents or text passages, would help performance on a number of key tasks, currently-available resources seriously limit our potential for achieving this with supervised systems. The analysis in this paper aims for a better understanding of the precise nature of these limitations in order to address them more deliberately and with a principled understanding of the coverage problems faced by current systems.

To this end, we outline a typology of coverage gaps and analyze both coverage of FrameNet and performance of a supervised semantic role labeling system on three different full-text data sets, totaling over 150,000 frame-assignment targets. We find that, on average, 46.8% of targets are not covered under straight supervised-classification approaches to frame assignment.

Acknowledgments

This research has been funded by the German Research Foundation DFG under the MMCI Cluster of Excellence. Thanks to the anonymous reviewers, Josef Ruppenhofer, Ines Rehbein, and Hagen Fürstenau for interesting and helpful comments and discussions, and to Collin Baker for assistance with data.

References

C. Baker, M. Ellsworth, K. Erk. 2007. Semeval-2007 task 19: Frame semantic structure extraction. In *Proceedings of SemEval-2007*.

A. Burchardt, K. Erk, A. Frank. 2005. A WordNet Detour to FrameNet. In *Proceedings of the GLDV-05 Workshop GermaNet II*.

A. Burchardt, M. Pennacchiotti, S. Thater, M. Pinkal. 2009. Assessing the impact of frame semantics on textual entailment. *Journal of Natural Language Engineering, Special Issue on Textual Entailment*, 15(4):527–550.

D. D. Cao, D. Croce, M. Pennacchiotti, R. Basili. 2008. Combining word sense and usage for modeling frame semantics. In *Proceedings of STEP-08*.

K. Deschacht, M.-F. Moens. 2009. Semi-supervised Semantic Role Labeling Using the Latent Words Language Model. In *Proceedings of EMNLP-09*.

M. Ellsworth, K. Erk, P. Kingsbury, S. Padó. 2004. PropBank, SALSA, and FrameNet: How Design Determines Product. In *Proceedings LREC 2004 Workshop on Building Lexical Resources from Semantically Annotated Corpora*.

K. Erk, S. Padó. 2006. Shalmaneser – a toolchain for shallow semantic parsing. In *Proceedings of LREC-06*.

K. Erk. 2005. Frame assignment as word sense disambiguation. In *Proceedings of IWCS 6*.

H. Fürstenau, M. Lapata. 2009a. Graph alignment for semi-supervised semantic role labeling. In *Proceedings of EMNLP 2009*.

H. Fürstenau, M. Lapata. 2009b. Semi-supervised semantic role labeling. In *Proceedings of EACL 2009*.

D. Gildea, D. Jurafsky. 2002. Automatic labeling of semantic roles. *Computational Linguistics*, 28(3):245–288.

A. Gordon, R. Swanson. 2007. Generalizing semantic role annotations across syntactically similar verbs. In *Proceedings of ACL 2007*.

R. Johansson, P. Nugues. 2007. Using WordNet to extend FrameNet coverage. In *Proceedings of the Workshop on Building Frame-semantic Resources for Scandinavian and Baltic Languages, NODALIDA*.

M. Marcus, G. Kim, M. A. Marcinkiewicz, R. MacIntyre, A. Bies, M. Ferguson, K. Katz, B. Schasberger. 1994. The Penn Treebank: Annotating predicate argument structure. In *ARPA Human Language Technology Workshop*.

D. McCarthy. 2009. Word Sense Disambiguation: An Overview. *Language and Linguistics Compass*, 3(2):537–558.

A. Moschitti, P. Morarescu, S. Harabagiu. 2003. Open-domain information extraction via automatic semantic labeling. In *Proceedings of FLAIRS*.

A. Moschitti, D. Pighin, R. Basili. 2008. Tree Kernels for Semantic Role Labeling. *Computational Linguistics*, 34(2).

S. Padó, M. Pennacchiotti, C. Sporleder. 2008. Semantic role assignment for event nominalisations by leveraging verbal data. In *Proceedings of Coling 2008*.

M. Palmer, D. Gildea, P. Kingsbury. 2005. The Proposition Bank: An Annotated Corpus of Semantic Roles. *Computational Linguistics*, 31(1):71–105.

M. Pennacchiotti, D. D. Cao, R. Basili, D. Croce, M. Roth. 2008a. Automatic induction of FrameNet lexical units. In *Proceedings of EMNLP-08*.

M. Pennacchiotti, D. D. Cao, P. Marocco, R. Basili. 2008b. Towards a Vector Space Model for FrameNet-like Resources. In *Proceedings of LREC-08*.

J. Ruppenhofer, M. Ellsworth, M. R. L. Petruck, C. R. Johnson, J. Scheffczyk. 2006. FrameNet II: Extended Theory and Practice.

J. Ruppenhofer, C. Sporleder, R. Morante, C. Baker, M. Palmer. 2009. SemEval-2010 Task 10: Linking Events and Their Participants in Discourse. In *Proceedings of SEW-2009*.

J. Ruppenhofer, M. Pinkal, J. Sunde. 2010. Generating FrameNets of various granularities. In *Proceedings of LREC 2010*.

D. Shen, M. Lapata. 2007. Using semantic roles to improve question answering. In *Proceedings of EMNLP-2007*.

M. Surdeanu, S. Harabagiu, J. Williams, P. Aarseth. 2003. Using predicate-argument structures for information extraction. In *Proceedings of ACL 2003*.

K. Toutanova, A. Haghighi, C. D. Manning. 2008. A Global Joint Model for Semantic Role Labeling. *Computational Linguistics*, 34(2).

Word Space Modeling for Measuring Semantic Specificity in Chinese

Ching-Fen Pan
Department of English
National Taiwan Normal University
debbychingxp@hotmail.com

Shu-Kai Hsieh
Department of English
National Taiwan Normal University
shukai@gmail.com

Abstract

The aim of this study is to use the word-space model to measure the semantic loads of single verbs, profile verbal lexicon acquisition, and explore the semantic information on Chinese resultative verb compounds (RVCs). A distributional model based on Academia Sinica Balanced Corpus (ASBC) with Latent Semantic Analysis (LSA) is built to investigate the semantic space variation depending on the semantic loads/specificity. The between group comparison of age-related changes in verb style is then conducted to suggest the influence of semantic space on verbal acquisition. Finally, it demonstrates how meaning exploring on RVCs is done with semantic space.

1 Introduction

The issue of 'word space' has been gaining attention in the field of distributional semantics, cognitive and computational linguistics. Various methods have been proposed to approximate words' meanings from linguistic distance. One of the most popular models in distributional semantics is Latent Semantic Analysis (LSA) with dimension-reduction technique, Singular Value Decomposition (SVD)(Landauer and Dumais, 1997; Karlgren and Sahlgren, 2001; Sahlgren, 2002; Widdows et al., 2002). The backbone of LSA is the co-occurrence distributional model in which words are conceived as points scattered in a texts-built n-dimensional space(Lenci, 2008). Rather than trying to predict the best performing model from a set of models, this study highlights the extent to which word space or semantic space measured from a vector-based model can access the verbal semantics and has influence on verbal acquisition.

This paper is organized as follows: Section 2 profiles the variation of semantic space affected by the semantic loads of single verbs. Section 3 discusses the correlation between the developing change in verbal lexicon and word space from the experimental data collected by M3[1] project. It will reveal how semantic space facilitates early child verbal learning. Section 4 demonstrates how to assess the meaning of Chinese resultative verb compounds (RVCs) from semantic space. The results of this work are finally concluded in Section 5.

2 The Variation of Semantic Space Between Two Verb Types (G/S) in LSA

The goal of this section is to examine the semantic variation between two verb types, generic versus specific verbs. It first creates a taxonomy for the classification of various verb groups (generic verbs versus specific verbs) based on the semantic distance with Latent Semantic Analysis (LSA) and Cluster Analysis.

2.1 Distributional Model Based on Sinica Corpus

The distributional model built in this survey is based on the Chinese texts collected in Academia

[1] Model and Measurement of Meaning: A Cross-lingual and Multi-disciplinary Approach of French and Mandarin Verbs based on Distance in Paradigmatic Graphs. Project website: http://140.112.147.149:81/m3/

Coling 2008: Poster Volume, pages 937–945,
Beijing, August 2010

Sinica Balanced Corpus (ASBC)[2]. It includes 190 files containing about 96000 word types[3]. The original matrix (M) is further decomposed into the product of three matrices (TSD^T). These matrices are then reduced into *k* dimensions. In the following reconstruction process based on *k* dimensions, it multiplies out the truncated matrices $T_k S_k D'_k$ and then gets a M_k matrix (the approximation of X)(Landauer et al., 1998; Sahlgren, 2005; Widdows and Ferraro, 2008). The following shows an example of finding the nearest neighbors of the word *da* (打 / to hit) via two methods (see Table 1). For the convenience of visualization and cluster analysis, Euclidean distance is applied in the following study.

	qu 'go'	*na* 'take'	*zhao* 'find'
Cosine	0.928	0.926	0.920
Distance	0.377	0.382	0.397

Table 1: Associating words of *da* 'hit'.

2.2 Semantic Clustering

The primary objective of cluster analysis is to examine the formation of a taxonomy: whether G verbs and S verbs form two groups separately. The clusters also help us grasp the semantic space among verbs as well as the potential semantic relation of them. Based on the distance matrix of lexical items generated in the last section, this part applied cluster analysis on the selected 150 verbs/observations[4]. For the convenience of comparison, each verb is coded with its type and a serial number like *zuo* (做/ to do) is G1 and *si* (撕/ to tear) is S27[5].

Once the similarity measure is done, the next procedure is to combine similar verbs into groups. The clustering procedure starts with each verb/observation in its own cluster, and combines two clusters together step by step until all the verbs are in a single cluster[6] The cluster dendrogram is plotted is Figure 1, in which clusters are formed from the bottom to the top.

Figure 1 demonstrates that the highest split separates these verbs into two big groups: the left branch group and right branch group drawn in different squares. The constituents of the two branches are listed in Table 2. It is clear that most of the constituent parts of the left group are G verbs whereas S verbs count as majority in the right group. If the left group is considered as a group formed with G verbs and right group with S verbs, the hit ratio[7] of G verbs (74.6%) is much higher than that of S verbs (57.1%). The clustering algorithm that we applied shows some structure, but there is no accurate separation of these two verb types. A detailed investigation of the relationship between the verb type and the distance is discussed in the next section.

	left group	right group
Generic verbs	59 (64.1%)	18 (33.3%)
Specific verbs	20 (21.7%)	24 (44.5%)
Undetermined verbs	13 (14.1%)	12 (22.2%)
Hit ratio	74.6%	57.1%

Table 2: Distribution of G/S verbs in two big clusters.

[2]ASBC website:
http://dbo.sinica.edu.tw/ftmsbin/kiwi1/mkiwi.sh

[3]The hapax legomena (words occur only once in the whole data) are not included in the matrix. The total word types including hapax amount to 220000 or so. To avoid time and computer consuming, we excluded those hapax from the co-occurrence matrix.

[4]These 150 verbs are single verbs selected from the experimental data. In the previous study of classification, these verbs are divided into two types (G:generic versus S:specific). There are 78 G verbs and 45 S verbs, along with 27 U(undetermined) verbs. It is noticeable that U verbs do not count as one type of verbs. They are floating verbs between G and S. We keep their identity as U and examine their potential characteristics in a binary cluster analysis.

[5]In fact, only 146 of 150 verbs are being classified because four words are missed in Sinica Corpus. To avoid confusion, we still call them 150 verbs in cluster analysis.

[6]Agglomerative method is implemented in the process in which single points are agglomerated into larger groups. This is termed a hierarchical cluster procedure that explores the co-relational structure of these single verbs. In complete linkage, all objects in a cluster are linked to each other with the longest distance. The use of the longest distance in complete linkage makes the least similar pair of objects group together. In other words, the maximum distance of the group results from the linkage of objects with minimum similarity.

[7]The hit ratio is calculated as follows:
hit ratio of G in the left group: $59/(59+20) = 74.6\%$
hit ratio of S in the right group: $24/(18+24) = 57.1\%$
It is noticeable that U verbs are temporarily ignored here.

2.3 Distance Variation in Small-G/S-clusters

Following the line of argumentation, this section demonstrates how distance varies within small-G-clusters and small-S-clusters. In order to examine the distance difference, small-G-cluster (or small-S-cluster) is defined as a cluster formed with the nearest twenty words of the G verb (or S verb) target.[8] In the example of one G verb *yong* (用/use) coded as G5, the closest twenty words are almost G verbs and the only one S verb is the farthest word *xie* (寫/write) (see Figure 2). The distance examination of the small cluster is applied to all of the 150 verbs studied in this survey. Table 3 has illustrated the comparison of verb types and the distance in the small cluster. As expected, the semantic distance is significantly affected by the verb type of the target word in the small cluster. The distances among words in most of the small-G-clusters range between 0.4 and 0.8. In contrast, over eighty percent small-S-clusters obtain a distance from 0.8 to 1.2. As for those U verbs which can not be decided as generic or specific in the manual tagging because of the lacking of agreement, they have distance between 0.6 and 1. Their distance shows an overlap with part of G verbs and part of S verbs. It confirms that U verbs are in a fuzzy zone between G verbs and S verbs.

In summary, G verbs are words with more senses and they appear more frequently in various context. Based on their high frequency distribution, G verbs construct a solid relation with each other in small-G-clusters. In contrast, S verbs are words with restricted meanings and they have relatively limited distributional patterns. Due to their low variety of patterns, S verbs are not easy to have tight relations with other words. It shows that words with generic meaning have high distribution variety and the distances among them are much shorter. The lack of polysemous feature makes the specific verbs be short of various distributional patterns and lose the opportunities to form close semantic relation with others. The semantic space among G verbs is short enough to form a solid cluster whereas S verbs are relatively remote from each other in semantic space. The distance of each verb cluster can help assess the verb category as generic (G) or specific (S). Approximately 75% of generic verbs form small clusters with distance lower than 0.8 while more than 80% of specific verbs acquire a

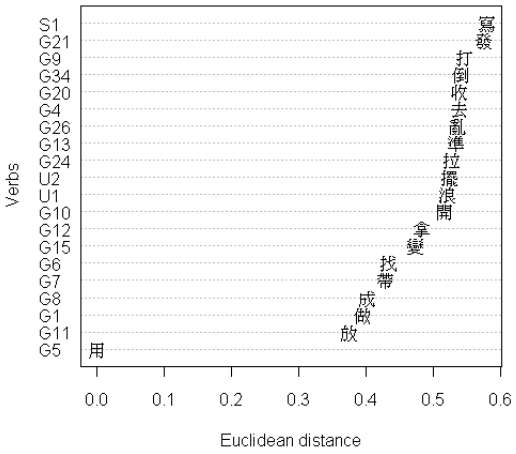


Figure 2: The small-G-cluster of *yong* (用/use).

[8] In order to test the representative power of small-clusters with 20 words, we have examined the clusters with 25 and 30 words as well. In all of the cases, the curves in 20-word cluster don't change significantly when the sample size is set to 25 or 30. The small-G/S-clusters with the sample size (N=20) is justified as representative.

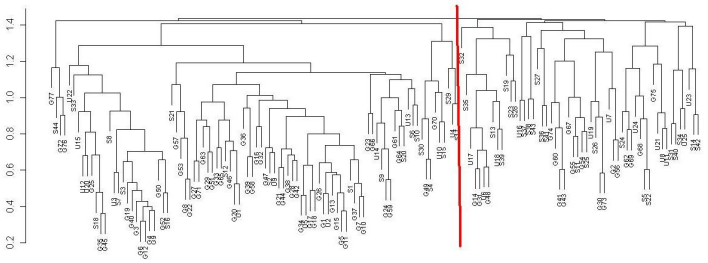


Figure 1: Agglomerative hierarchical cluster analysis of 150 verbs.

distance greater than 0.8 . As to the verbs of indeterminacy, they are averagely scattered in a fuzzy zone between G and S verbs. Over 70% U verbs are centering the distance 0.8, which suggests that words near distance 0.8 are likely to be undetermined verbs. This analysis has proved that semantic space varies in accordance with verb's meaning specificity. The distributions in context represent not only the linguistic behaviors but the semantic contents of lexical items.

3 The Influence of Specificity on Acquisition

This section assesses the influence of semantic space on the acquisition of the verbal lexicon. With the examination of Specific verb (S verb) progress, this study proposes that Generic verbs (G verbs) are acquired earlier than S verbs due to the closer semantic space. It also testifies whether the S verb development is a developing trend parallel with the acquisition of conventional verbs(Chen et al., 2008; Hsieh et al., 2009)[9] from the experimental data collected by M3 project. Based on the developing trend of conventional lexical items, the following parts analyze the relation of meaning specificity and the acquisition of lexical items.

3.1 Decreasing in Lexical Variation

The section is concerned with lexical variation among participants within the same age group. It measures type-token ratios of each group and profiles the lexical variation[10] in verbal acquisition. Data analyzed in this part include five groups of respondents' usages of verbs to four different films, each of which pictures one event. Respondents are assigned into five groups according to their age: 3-year-old, 5-year-old, 7-year-old, and 9-year-old groups have 20 respondents separately while 60 respondents are in the Adult group composed of people in their twenties. In respondents' answers, only one single verb is extracted from each respondent in this study. The number of verbs in each group is equal to the amount of participants. The first analysis begins with the lexical variation or lexical flexibility in these five groups. It is done with the ratio of lexical variation: the amount of word type is divided by the amount of word token, as shown in Table 4. The greater number of the ratio means the lexical variation is more abundant and the smaller ratio means a low diversity of word types. The ratio of lexical variation in these four films all show a decreasing trend from 3-year-old groups to adult groups. The quantity of different verbs is higher in children group (3y, 5y,7y, 9y) than that in adult group. That is, children appear more creative in event description tasks while adults are confined in the conventional usage. With the decreasing trend of lexical variety, the next step is to propose an increasing trend of specific verb

[9]They rearranged the five groups of participants into three units and then investigated the learning trend by Replacing Rate (Frequency of $V2_{freq}$ / Frequency of $V1_{freq}$). By defining adults' usages as the conventional one called V1, children's second highest frequency verb is counted as V2. Along with the increase of age, the number of V2 drops slowly whereas the amount of V1 increases gradually.

[10]Lexical diversity or sometimes called lexical variation is used to mean a combination of lexical variation and lexical sophistication. It is also referred to an indication of a combination of vocabulary size and the ability to use it effectively(Malvern et al., 2004). However, lexical variation or lexical diversity doesn't mean lexical richness in this study. In other kinds of experiment like writing tests, adults should perform better than children in lexical diversity. But the experimental data applied in this study is action-naming task. The trend of lexical variation may perform in an opposite way.

Distance	0.4-0.6	0.6-0.8	0.8-1.0	1.0-1.2
Small-G-cluster	24 (31.2%)	32 (41.6%)	17 (22.0%)	4 (5.2%)
		Total:72.8%		Total:27.2%
Small-S-cluster	0 (0)	6 (13.6%)	19 (43.2%)	19 (43.2%)
		Total:13.6%		Total:86.4%
Small-U-cluster	1 (4%)	8 (32%)	11 (44%)	5 (20%)

Table 3: Comparison of verb types (G/S) and semantic distance within small cluster.

usage when the age raises. It will show that the change is from various generic verbs to one or two specific verbs rather than various specific verbs.

Films	carrot-peel	paper-crumple	plank-saw	glass-break
3y	0.35	0.55	0.2	0.33
5y	0.25	0.47	0.2	0.2
7y	0.3	0.2	0.25	0.1
9y	0.21	0.105	0.157	0.157
Adult	0.016	0.083	0.066	0.066

Table 4: The ratio of lexical variation ($ratio$ = word type/word token).

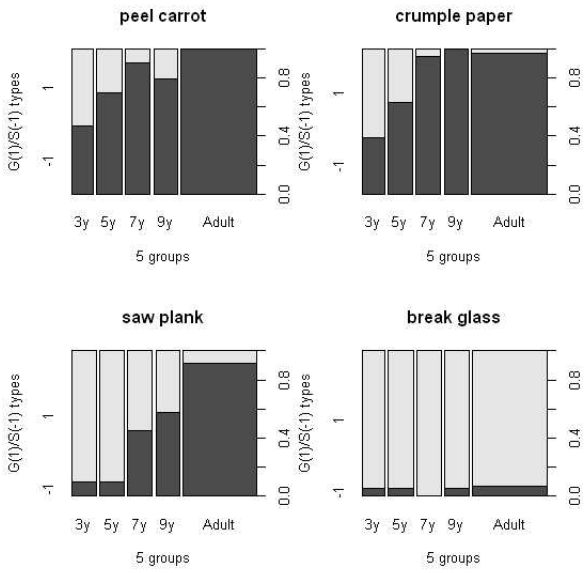


Figure 3: The proportion of S (-1) verbs to G (1) verbs from 5 groups of respondents to four events.

3.2 Increasing in Specific Verbs

With regard to the aim of the investigation, the findings reported above provide evidence of the changing trend of lexical variety in action-naming tasks. The next step is to discover the developing trend of verb type (G/S) usage. According to the annotation result of verb category, each verb in the data is now transferred into either generic (label as G or 1) or specific (S or -1) and the proportions of S verbs is plotted as Figure 3.

3.2.1 The Non-proportionality of S Verb among Age Groups

A closer investigation is then implemented for non-proportionalities by chi-squared test(Baayen, 2008). Although the proportion of S verb changes more or less in different groups, it is still need to confirm that whether S verbs are more frequently used by adults than children. The hypothesis is formulated as follows:

H_0: `The proportions of the two verb types (G verb vs. S verb ) do NOT vary in five age groups.`

With Pearson's chi-square test for four sets of data. It is reported that the small p-values (9.779e-07, 1.324e-09, and 1.191e-13) in the first three sets of data (carrot-peel (f_6), paper-crumple (f_2), and plank-saw (f_16)) suggest a non-proportionality of S verb in different age groups. However, the p-value (0.8467) obtained in the last data set (glass-break (f_3)) is too large to suggest a significant variation of S verb proportion in different age groups. It proves that the proportions of S verb change with the participant's age in the three event-naming tasks but that doesn't happen in the glass-break (f_3) event. Except for the data in glass-break (f_3) event, the null hypothesis doesn't hold in the analysis.

3.2.2 The Relationship between S Verb and Age

In order to test the correlation of S verb proportion and age variation, four groups (3y, 5y, 7y, 9y) are merged into one group called Child versus Adult group. The data are now represented by two by two contingency tables with one categorical dependent variable (verb types) and one categorical independent variable (age). Here summarizes the hypothesis:

H_0: `The frequency of the two verb types (G verb vs. S verb, the dependent variable) do NOT vary depending on participants' age (Child vs. Adult, the independent variable).`

The result has shown that the small p-values (2.803e-05, 0.001225, 1.754e-12) verify the significant difference of S verb in Child group

and Adult group with regard to the three data sets in carrot-peel (f_6), paper-crumple (f_2), and plank-saw (f_16). Along with the correlation examination, the effect size is revealed with correlation coefficient from 0 (no correlation) to 1 (perfect correlation)(Gries, 2009). According to the Phi value in this table, only the data in plank-saw (f_16) has a correlation coefficient (0.612) greater than 0.5. That is, the correlation between S verb usage and age group is considered as significantly correlated in the one data set (plank-saw (f_16)). As for the other two data sets (carrot-peel (f_6) with phi:0.379, paper-crumple (f_2) with phi: 0.297), the correlation is not particularly strong but it is still highly significant. Over half of the data sets exhibit a significant non-proportionality of S verb usage in different age groups but the correlation of S verb and participants' age requires.

In relation to the aim of this study, it has shown that meaning specificity functions as a factor in the development of verbal lexicon. The results of the analysis also show a significant variety of S verb between children and adults. It is plausible to suppose that verbs with specific meaning are acquired later than those with generic meanings. This developing trend suggests that a closer semantic space among G verbs facilitates the acquisition of verb meanings whereas a distant space among S verbs causes difficulties in meaning acquiring. Once those verbs with specific meanings are picked up, most of them will become the so-called conventional verbs. When the conventional use to an action is a specific verb, the progress of S verb usage is more obvious. The usage of verbs with specificity meaning is a developing trend of language acquisition.

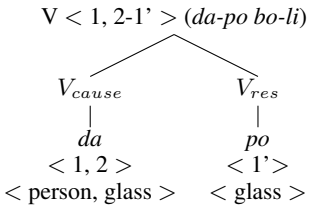
4 Meaning Exploring on Chinese Resultative Verb Compounds (RVCs)

In the verb-event co-occurrence matrix, verbs elicited from the same event are considered to be verbs have the same object in a verb-object co-occurrence matrix. With the distributional model, it then shows how meaning specificity affects the linguistic behavior and semantic content of Chinese resultative verb compounds (RVCs). Those RVCs with similar distributional patterns will present a high semantic relation. This semantic relation could result from the meaning of the first verbal morpheme (V_{caus}) or the second one (V_{res}). It is further proposed that the verb type (generic or specific) of V_{caus} would affect the whole meaning content of V-V compounds.

4.1 The RVC Structure in the Data

A Chinese resultative verb compound (RVC) consists two main elements: the first element (V_{caus}) expresses a causing event or a state while the second element (V_{res}) denotes a resulting event or the aspectual properties of the object. According to the Aspectual Interface Hypothesis(Tenny, 1989), the property of an internal argument can measure out the event. In the Chinese example, *da-po bo-li* (打破玻璃 / hit-break glass), the state of the object *bo-li* (玻璃 / glass) is changed into smashed and this change points out an end point of the event. The resultative *po* (破 / broken) is an delimiting expression which refers to the property of the object. In addition to defining the second element of an RVC as a delimiting expression, other surveys label it as V_{res} which requires the saturation of arguments. Four possible V-V compound argument structures are proposed in Li's (1990) works. In the following studies, most of RVCs require an argument structure like (1). The first verbal morpheme (V_{caus}) has a theta-grid $<$1, 2$>$ and the second morpheme (V_{res}) has $<$1'$>$. V_{caus} requires an external argument (a person) and an internal argument (a glass). The internal argument (a glass) is identified with the argument of V_{res}. Since the internal argument of V_{caus} has to be identified with the argument of V_{res}, it raises the issue that which one functions more prominent in choosing the object of a V-V compound. From the study of RVCs' distributional pattern, it examines which one (V_{caus} or V_{res}) is more salient and also dominates the argument selection of a V-V compound.

(1)



4.2 Semantic Assessment

The semantic links among words are built by measuring the linguistic distances among them. In order to examine the semantic information of RVCs, a sub-sample with thirty-six verbs is selected to do cluster tasks. The semantic relationships of word in the sub-sample is visualized as a clustering tree, as shown in Figure 4. The figure shows that an RVC with a G verb as its V_{caus} ($GV_{caus} - V_{res}$) build a close relation with other RVCs which have the same V_{res} with it. Take the most extreme G verb *da* (打/hit)as an example, *da-lan* (打爛/hit-ruin) is closer to *pai-lan* (拍爛/hit with palm and ruin) than *da* (打/hit). On the other hand, an RVC with an S verb as its V_{caus} ($SV_{caus} - V_{res}$), are grouped with those having the same V_{caus}. The RVC, *ju-kai* (鋸開/saw-open), with a S verb *ju* (鋸/saw) as its head, forms a cluster with *ju* (鋸/saw) and *ju-duan* (鋸斷/saw-crack).

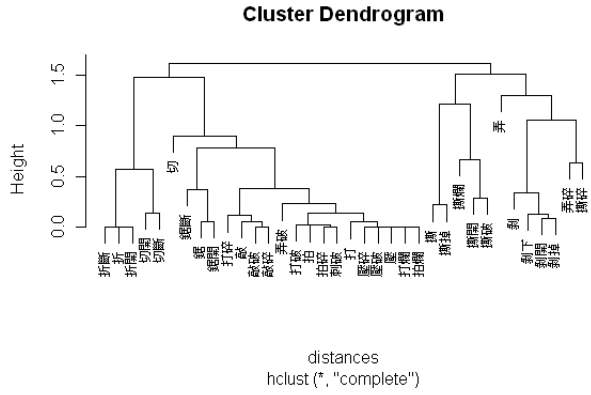


Figure 4: Semantic clustering of selected verbs.

With regard to the semantic relation of RVCs shown in the cluster plot, the next step is to justify the proportion of RVCs with the structure $GV_{caus} - V_{res}$ in which V_{res} selects a G verb as its V_{cause}. As Table 5 shows, the proportion of $GV_{caus} - V_{res}$ and $SV_{caus} - V_{res}$ is 50% respectively. That is, half of the selected seven V_{res} pick up a G verbs as its head while the other half words go with S verbs. Those V_{res} preferring a G head to a S head are *sui*, *po*, *lan*, *duan*; those preferring a S verb to a G verb head are *kai*, *diao*, *xia*. According to the semantic content these resultative verbs, *kai*, *diao*, *xia* describes the direction of the action and the motion of objects and they are defined as 'path' V_{res} in Ma and Lu's (1997) work. As for *sui*, *po*, *lan*, *duan* called as 'result' V_{res}, they mainly express the result of the object affected by the action. The outcome reported here suggests that 'result' V_{res} is apt to have a G verb as its head verb whereas 'path' V_{res} tends to pick up a S head verb. The proposal in literatures that V_{res} tends to choose a G head verb is justified as valid when the V_{res} expresses the meaning of 'result' rather than 'path.'

	GV_{caus}	**S**V_{caus}
'result' V_{res}		
sui (碎/smash)	da, nong, pai, ya, qiao	si
po (破/break)	da, nong, ya, qiao	si, ci
lan (爛/ruin)	da, pai	si
duan (斷/crack)	qie	
Proportion	47%	15%
'path' V_{res}		
kai (開/open)	qie	zhe, ju, si, bo
diao (掉/fall)		zhe, ju, si, bo
xia (下/down)		bo
Proportion	3%	35%

Table 5: $GV_{caus} - V_{res}$ versus $SV_{caus} - V_{res}$.

In summary, words with small distance resulting from their similar distributional patterns can be interpreted to be semantically similar in a semantic cluster. The result of semantic clustering has suggested that the meaning of RVCs depend on either the V_{caus} or the V_{res}. The meaning of $GV_{caus} - V_{res}$ is more determined by V_{res} because GV_{caus} is more polysemous and the V_{res} becomes a prominent role to dominate the meaning of $GV_{caus} - V_{res}$. In contrast, $SV_{caus} - V_{res}$ focuses on the part of SV_{caus} since

SV_{caus} expresses its meaning specific enough. In addition, the property of V_{res} also affects the category of its head verb. When V_{res} like *sui* belong to the 'result' V_{res}, it tend to choose a G verb as its V_{caus}. On the other hand, the 'path' V_{res} like *xia*, its head verb is apt to be a S verb. It is suggested that 'path' V_{res} is more likely to have a G verb than 'path' V_{res}. As the empirical study illustrates the semantic information on Chinese RVCs are affected by the semantic space of words.

5 Conclusion

In this paper, we argue the following points: firstly, the distributional model shows that the semantic space differ clearly in accordance with the specificity of verbs. The G verbs form tight relations with each other and become a larger cluster whereas the semantic space among S verbs is too distant to become a solid group. Secondly, semantic space has influence on the acquiring of words' meanings. Generic verbs are earlier and easier acquired due to the closer semantic space among words. The developing trend of specific verb lexicon parallel with conventional usage suggests a language acquisition phenomenon. Finally, the G/S verbs play an influential role in Chinese resultative compounds. The resultative verb becomes more prominent when the first verb is with a generic meaning. The 'result' V_{res} is apt to have a G verb as its head verb whereas 'path' V_{res} tends to pick up a S head verb. We believe that results of our analysis will shed light on semantic assessment and make predictions for lexical acquisition.

References

Baayen, R. H. 2008. *Analyzing Linguistic Data: A Practical Introduction to Statistics using R.* Cambridge University Press.

Chen, P., M.-A. Parente, K. Duvignau, L. Tonietto, and B. Gaume. 2008. Semantic approximations in the early verbal lexicon acquisition of chinese: Flexibility against error. *The 7th Workshop on Chinese Lexical Semantics*.

Gries, Stefan Thomas. 2009. *Quantitative Corpus Linguistics with R: A Practical Intriduction*. Routledge.

Hsieh, Shu-Kai, Chun-Han Chang, Ivy Kuo, Hintat Cheung, Chu-Ren Huang, and Bruno Gaume. 2009. Bridging the gap between graph modeling and developmental psycholinguistics: An experiment on measuring lexical proximity in chinese semantic space. Presented at The 23rd Pacific Asia Conference on Language, Information and Computation (PACLIC 23). Hong Kong: City University of Hong Kong., December 3-5.

Karlgren, J. and M. Sahlgren. 2001. From words to understanding. In Uesaka, Y., Kanerva P. and H. Asoh, editors, *Foundations of Real-World Intelligence*, pages 294–308.

Landauer, T. K. and S. T. Dumais. 1997. A solution to plato's problem: The latent semantic analysis theory of the acquisition, induction, and representation of knowledge. *Psychological Review*, 104:211–240.

Landauer, T. K., P. W. Foltz, and D. Laham. 1998. Introduction to latent semantic analysis. *Discourse Processes*, 25:259–284.

Lenci, A. 2008. Distributional semantics in linguistic and cognitive research. *From context to meaning: Distributional models of the lexicon in linguistics and cognitive science, special issue of the Italian Journal of Linguistics*, 20/1:1–31.

Li, Yafei. 1990. On v-v compounds in chinese. *Natural Language and Linguistic Theory*, 8:177–207.

Ma, Zhen and Jian-Ming Lu. 1997. Xingrongci zuo jieguobuyu qingkuang kaocha yi (形容詞作結果補語情況考察(一)). *Hanyuxuexi (漢語學習)*, 1:3–7.

Malvern, David D., Brian J. Richards, Ngono Chipere, and Pilar Duran. 2004. *Lexical diversity and language development : quantification and assessment*. New York : Palgrave Macmillan.

Sahlgren, M. 2002. Random indexing of linguistic units for vector-based semantic analysis. *ERCIM News*, 50.

Sahlgren, Magnus. 2005. An introduction to random indexing. In *Proceedings of the Methods and Applications of Semantic Indexing Workshop at the 7th International Conference on Terminology and Knowledge Engineering (TKE)*. Copenhagen, Denmark.

Tenny, Carol. 1989. The aspectual interface hypothesis. In *Proceedings of NELS 18*. University of Massachusetts at Amherst.

Widdows, Dominic and Kathleen Ferraro. 2008. Semantic vectors: a scalable open source package and online technology management application. In Nicoletta Calzolari (Conference Chair),

Khalid Choukri, Bente Maegaard Joseph Mariani Jan Odjik Stelios Piperidis Daniel Tapias, editor, *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. European Language Resources Association (ELRA).

Widdows, Dominic, Scott Cederberg, and Beate Dorow. 2002. Visualisation techniques for analysing meaning. In *Fifth International Conference on Text, Speech and Dialogue (TSD 5)*, pages 107–115. Brno, Czech Republic.

MT Error Detection for Cross-Lingual Question Answering

Kristen Parton
Columbia University
New York, NY, USA
kristen@cs.columbia.edu

Kathleen McKeown
Columbia University
New York, NY, USA
kathy@cs.columbia.edu

Abstract

We present a novel algorithm for detecting errors in MT, specifically focusing on content words that are deleted during MT. We evaluate it in the context of cross-lingual question answering (CLQA), where we try to correct the detected errors by using a better (but slower) MT system to retranslate a limited number of sentences at query time. Using a query-dependent ranking heuristic enabled the system to direct scarce MT resources towards retranslating the sentences that were most likely to benefit CLQA. The error detection algorithm identified spuriously deleted content words with high precision. However, retranslation was not an effective approach for correcting them, which indicates the need for a more targeted approach to error correction in the future.

1 Introduction

Cross-lingual systems allow users to find information in languages they do not know, an increasingly important need in the modern global economy. In this paper, we focus on the special case of cross-lingual tasks with result translation, where system output must be translated back into the user's language. We refer to tasks such as these as *task-embedded machine translation*, since the performance of the system as a whole depends on both task performance and the quality of the machine translation (MT).

Consider the case of cross-lingual question answering (CLQA) with result translation: a user enters an English question, the corpus is Arabic, and the system must return answers in English. If the corpus is translated into English before answer extraction, an MT error may cause the system to miss a relevant sentence, leading to decreased recall. Boschee et al. (2010) describe six queries from a formal CLQA evaluation where none of the competing systems returned correct responses, due to poor translation. In one example, the answer extractor missed a relevant sentence because the name "Abu Hamza al-Muhajir" was translated as "Zarqawi's successor Issa." However, even if answer extraction is done in Arabic, errorful translations of the correct answer can affect precision: if the user cannot understand the translated English sentence, the result will be perceived irrelevant. For instance, the user may not realize that the mistranslation "Alry$Awy" refers to Al-Rishawi.

Our goal was not to improve a specific CLQA system, but rather to find MT errors that are likely to impact CLQA and correct them. We introduce an error detection algorithm that focuses on several common types of MT errors that are likely to impact translation adequacy:

- content word deletion
- out-of-vocabulary (OOV) words
- named entity missed translations

The algorithm is language-independent and MT-system-independent, and generalizes prior work by detecting errors at the word level and detecting errors across multiple parts of speech.

We demonstrate the utility of our algorithm by applying it to CLQA at query time, and investigate using a higher-quality MT system to correct the errors. The CLQA system translates the full corpus, containing 119,879 text documents and 150 hours of speech, offline using a production MT system, which is able to translate quickly (5,000 words per minute) at the cost of lower quality translations. A research MT system has higher quality but is too slow to be practical for a large amount of data (at 2 words per minute,

Coling 2008: Poster Volume, pages 946–954,
Beijing, August 2010

it would take 170 days on 50 machines to translate the corpus). At query-time, we can call the research MT system to retranslate sentences, but due to time constraints, we can only retranslate k sentences (we set k=25). In order to choose the sentences to best improve CLQA performance, we rank potential sentences using a relevance model and a model of error importance.

Our results touch on three areas:

- Evaluation of our algorithm for detecting content word deletion shows that it is effective, accurately pinpointing errors 89% of the time (excluding annotator disagreements).
- Evaluation of the impact of re-ranking shows that it is crucial for directing scarce MT resources wisely as the higher-ranked sentences were more relevant.
- Although the research MT system was perceived to be significantly better than the production system, evaluation shows that it corrected the detected errors only 39% of the time. Furthermore, retranslation seems to have a negligible effect on relevance. These unexpected results indicate that, while we can identify errors, retranslation is not a good approach for correcting them. We discuss this finding and its implications in our conclusion.

2 Task-Embedded MT

A variety of cross-lingual applications use MT to enable users to find information in other languages: e.g., CLQA, cross-lingual information retrieval (CLIR), and cross-lingual image retrieval. However, cross-lingual applications such as these typically do not do result translation – for instance, an English-French CLIR system would take an English query and return French documents, assuming that result translation is a separate MT problem. Part of the reason for the separation between cross-lingual tasks and MT is that evaluating task performance on MT is often difficult. For example, for a multilingual summarization task combining English and machine translated English, Daumé and Marcu (2006) found that doing a pyramid annotation on MT was difficult due to the poor MT quality.

Assessing cross-lingual task performance without result translation is problematic, because in a real-world application, result translation would affect task performance. For instance, in English-Arabic CLIR, a poorly translated relevant Arabic document may appear to be irrelevant to an English speaker. Decoupling the cross-lingual application from the MT system also limits the opportunity for feedback between the application and the MT system. Ji and Grishman (2007) exploited a feedback loop between Chinese and English named entity (NE) tagging and Chinese-English NE translation to improve both NE extraction and NE translation.

In this paper, error detection is done at query time so that query context can be taken into account when determining which sentences to retranslate. We also use the task context to detect errors in translating NEs present in the query.

3 Related Work

There is extensive prior work in describing MT errors, but they usually involve post-hoc error analysis of specific MT systems (e.g., (Kirchhoff et al., 2007), (Vilar et al., 2006)) rather than online error detection. One exception is Hermjakob et al. (2008), who studied NE translation errors, and integrated an improved on-the-fly NE transliterator into an SMT system.

Content word deletion in MT has been studied from different perspectives. Li et al. (2008) and Menezes and Quirk (2008) explored ways of modeling (intentional) source-word deletion in MT and showed that it can improve BLEU score. Zhang et al. (2009) described how errors made during the word-alignment and phrase-extraction phases in training phrase-based SMT often lead to spurious insertions and deletions during translation decoding. This is a common error – Vilar et al. (2006) found that 22% of errors produced by their Chinese-English MT system were due to missing content words. Parton et al. (2009) did a post-hoc analysis on the cross-lingual 5W task and found that content word deletion accounted for 17-22% of the errors on that task.

Some work has been done in addressing MT errors for different cross-lingual tasks. Ji and

1) Source	kmA tHdv wzyr AldfAE AlAsrA}yly Ayhwd bArAk Al*y zAr mwqE Altfjyr AlAntHAry fy dymwnp fy wqt sAbq En Altfjyr AlAntHAry ...
ProdMT	There also the Israeli Defense Minister Ehud Barak, who visited the site of the suicide bombing in Dimona earlier, the suicide bombing ...
Ref.	Moreover, Israeli Defense Minister Ehud Barak, who visited the scene of the suicide bombing in Dimona earlier, spoke about the suicide bombing ...
2) Source	... Akd Ely rgbp hrAry AlAstfAdp mn AltjArb AlAyrAnyp fy mwAjhp Alqwy AlmEtdyp.
ProdMT	... stressed the desire to test the Iranian Harare in the face of the invading forces.
Ref.	... stressed Harare's desire to benefit from the Iranian experience in the face of the forces of aggressors.

Table 1: Two examples of content word deletion during MT.

Grishman (2007) detected NE translation errors in the context of cross-lingual entity extraction, and used the task context to improve NE translation. Ma and McKeown (2009) investigated verb deletion in Chinese-English MT in the context of CLQA. They tested two SMT systems, and found deleted verbs in 4-7% of the translations. After using post-editing to correct the verb deletion, QA relevance increased for 7% of the sentences, showing that an error that may have little impact on translation metrics such as BLEU (Papineni et al., 2002) can have a significant impact on cross-lingual applications.

Our work generalizes Ma and McKeown (2009) by detecting content-word deletions and other MT errors rather than just verb deletions. We also relax the assumption that translation preserves part of speech (i.e., that verbs must translate into verbs), assuming only that a phrase containing a content word should be translated into a phrase containing a content word. Instead of post-editing, we use an improved MT system to retranslate sentences with detected errors.

Using retranslation to correct errors exploits the fact that some sentences are harder to translate than others. In a resource-constrained setting, it makes sense to apply a better MT system only to sentences for which the fast MT system has lower confidence. We do not know of other systems that do multi-pass translation, but it is an interesting area for further work.

4 MT Error Detection

Most MT systems try to balance translation fluency with adequacy, which refers to the amount of meaning expressed in the original that is also expressed in the translation. For task-embedded MT, errors in adequacy are more likely to have an impact on performance than errors in fluency. Many MT metrics (such as BLEU) treat all tokens equally, so deleting a verb is penalized the same as deleting a comma. In contrast, we focus on errors in translating **content words**, which are words with open-class parts of speech (POS), as they are more likely to impact adequacy. First we describe how MT deletion errors arise and how we can detect them, and finally we describe detection of other types of errors.

4.1 Deletion in MT

The simplest case of content word deletion is a complete deletion by the translation model – in other words, a token was not translated. We assume the MT system produces word or phrase alignments, so this case can be detected by checking for a null alignment. However, it is necessary to distinguish correct deletion from spurious deletion. Some content words do not need to be translated – for example the Arabic copular verb "kAn" ("to be") is often correctly deleted when translating into English.

A more subtle form of content word deletion occurs when a content word is translated as a non-content word. This can be detected by comparing the parts of speech of aligned words. Consider the production MT System (Prod. MT) example in Table 1: the verb "tHdv"[1] ("spoke") has been translated as the expletive "there."

Finally, another case of content word deletion occurs when a content word is translated as part of a larger MT phrase, but the content word is not translated. In the second example in Table 1, an Arabic phrase consisting of a noun and preposition is translated as just the preposition "to."

[1] Arabic examples in this paper are shown in Buckwalter transliteration (Buckwalter, 2002).

The latter two kinds of content word deletion are considered mistranslations rather than deletions by the translation model, since the deleted source-language token does produce one or more target-language tokens. However, from the perspective of a cross-lingual application, there was a deletion, since some content that was present in the original is not present in the translation.

4.2 Detecting Deleted Content Words

The deletion detection algorithm is motivated by the assumption that a source-language phrase containing one or more meaning-bearing words should produce a phrase with one or more meaning-bearing words in the translation. (Phrase refers to an n-gram rather than a syntactic phrase.) Note that this does not assume a one-to-one correspondence between content words – for example, translating the phrase "spoke loudly" as the single word "yelled" satisfies the assumption. This hypothesis favors precision over recall, since it may miss cases where two content words are incorrectly translated as a single content word (for instance, if "coffee table" is translated as "coffee").

The algorithm takes as input POS tags in both languages and word alignments produced by the MT system during translation. The exact definition of "content word" will depend upon the language and POS tagset. The system iterates over all content words in the source sentence, and, for each word, checks whether it is aligned to one or more content words in the target sentence. If it has no alignment, or is aligned to only function words, the system reports an error. This rule-based approach has poor precision because of content words that are correctly deleted. For example, in the sentence "I am going to watch TV," "am" and "going" are tagged as verbs, but may be translated as function words. To address this, frequent content words were heuristically filtered using source-language IDF (inverse-document frequency) over the QA corpus. The cut-off was tuned on a development set.

This algorithm is a lightweight, language-independent and MT-system-independent way to find errors in MT. The only requirement is that the MT system produce word or phrase alignments. This algorithm generalizes Ma and McKeown (2009) in several ways. First, it detects any deleted content words, rather than just verbs. The previous work only addresses complete deletions, where the deleted token has a null alignment, whereas this approach finds cases where content words are mistranslated as non-content words. Finally, this error detection algorithm is more fine-grained, since it is at the word level rather than the phrase level.

4.3 Additional Error Detection Heuristics

For the CLQA task, we extended our MT error detection algorithm to handle two additional types of MT errors, OOV words and NE mistranslations, and to rank the errors. The production MT system was explicitly set to not delete OOV words, so they were easy to detect as source-language words left in the target language. The CLIR system was used to find occurrences of query NEs in the corpus, and then word alignments were used to extract the corresponding translations. If the translations were not a fuzzy match to the query, then it was flagged as a possible NE translation error. For instance, in a query about al-Rishawi, the CLIR would return Arabic-language matches to the Arabic word Alry$Awy. If the aligned English translation was al-Ryshoui instead of al-Rishawi, it would be flagged as an error.

Even if the retranslation corrects the errors in MT, if the sentences are not relevant, they will have no impact on CLQA. To account for relevance, we implemented a bilingual bag-of-words matching model, and ranked sentences with more keyword matches to the query higher. Sentences with the same estimated relevance were further sorted by potential impact of the MT error on the task. Errors affecting NEs (either via source-language POS tagging or source-language NE recognition) were ranked highest, since our particular CLQA task is focused on NEs. The final output of the algorithm is a list of sentences with MT errors, ranked by relevance to the query and importance of the error.

5 Experimental Setup

We begin by describing the MT systems, which motivate the need for time-constrained MT. Then we describe the CLQA task and the baseline CLQA system, and finally how the error detection algorithm is used by the CLQA system.

5.1 MT Systems

Both the research and production MT systems used in our evaluation were based on Direct Translation Model 2 (Ittycheriah and Roukos, 2007), which uses a maximum entropy approach to extract minimal translation blocks (one-to-M phrases with optional variable slots) and train system parameters over a large number of source- and target-language features. The research system incorporates many additional syntactic features and does a deeper (and slower) beam search, both of which cause it to be much slower than the production system. In addition, the research MT system filters the training data to match the test data, as is customary in MT evaluations, whereas the production system must be able to handle a wide range of input data. Part of the reason for the slower running time is that the research system has to retrain; the advantage is that more test-specific training data can be used to tailor the MT system to the input.

Overall, the research MT system performs 4 BLEU points better than the production MT system on a standard MT evaluation test corpus, but at a great cost: the production MT handles 5,000 words per minute, while the research MT system handles 2 words per minute. Using 50 machines, the production MT system could translate the corpus in under 2 hours, whereas the research MT system would take 170 days. This vast difference succinctly captures the motivation behind the time-constrained retranslation step.

5.2 CLQA Task

The CLQA task was designed for the DARPA GALE (Global Autonomous Language Exploitation) project. The questions found are open-ended, non-factoid information needs. There are 22 question types, and each type has its own relevance guidelines. For instance, one type is "Describe the election campaign of [PERSON]," and a question could be about Barack Obama. Queries are in English, the corpus is in Arabic, and the system must output comprehensible English sentences that are relevant to the question.

The Arabic corpus was created for the evaluation and consists of four genres: formal text (72,677 documents), informal text (47,202 documents), formal speech (50 hours), and informal speech (80 hours). The speech data was story segmented and run through a speech recognition system before translation. We used 31 text queries developed by the Linguistic Data Consortium (LDC), and 39 speech queries developed by other researchers working on the CLQA task.

5.3 CLQA System

The baseline CLQA system translates the full corpus offline before running further processing on the translated sentences (parsing, NE recognition, information extraction, etc.) and indexing the corpus. At query-time, CLIR (implemented with Apache Lucene) returns documents relevant to the query, and the CLQA answer extraction system is run over the translated documents. The answer extraction system relies on target-language annotations, but any MT errors will propagate to target-language processing, and therefore affect answer extraction.

5.4 CLQA System with MT Error Detection

The error detection and retranslation module was added to the baseline system after CLIR, but before answer extraction. The inputs to the detection algorithm are the query and a list of ranked documents returned by CLIR. The detection algorithm has access to the indexed (bilingual) corpus, source- and target-language annotations (POS tagging and NE recognition), and MT word alignments. The error detection algorithm has two stages: first it runs over sentences in documents related to the query, and after it finds $2k$ sentences with errors (or exhausts the document list), it reranks the errors as described in section 4.3 and retranslates the top k=25 sentences. Then the merged set of original and retranslated relevant sentences are passed to the

answer extraction module.

By doing retranslation before answer extraction, the algorithm has the potential to improve precision and recall. An improved translation of a relevant Arabic sentence is more likely to be selected by the answer extraction system and increase recall, as in Boschee et al. (2010), where answers were missed due to mistranslation. A better translation of a relevant sentence is also more likely to be perceived as relevant, as shown by Ma and McKeown (2009).

6 Evaluation

Amazon Mechanical Turk (AMT) was used to conduct a large-scale evaluation of the impact of error detection and retranslation on relevance. An intrinsic evaluation of the error detection was run on a subset of the sentences, since it required bilingual annotators.

6.1 Task-Based Evaluation

Each sentence was annotated in the production MT version and the research MT version. The annotators were first presented with template relevance guidelines and an example question, along with 3 – 4 example sentences and expected judgments. Then the actual question was presented to the annotator, along with 5 sentences (all from a single MT system). For each sentence, the annotators were first asked to judge perceived adequacy and then relevance.

The *perceived adequacy* rating was loosely based upon MT adequacy evaluations – in other words, annotators were told to ignore grammatical errors and focus on perceived meaning. However, since there were no reference translations, annotators were asked to rate how much of the sentence they believed they understood by selecting one of (All, More than half, About half, Less than half, and None).

The *relevance* rating was based on the template relevance guidelines, and annotators could select one of (Relevant, Maybe relevant, Not relevant, Can't tell due to bad translation and Can't tell due to other reason).

Genre	# detected errors per sentence	# detected errors per 1,000 tokens
Newswire	0.16	56
Broadcast news	0.23	105
Broadcast conversation	0.14	84

Table 2: Number of errors detected across different genres.

6.2 Amazon Mechanical Turk (AMT)

The evaluation was run on AMT, which has been extensively used in NLP and has been shown to have high correlation with expert annotators on many NLP tasks at a lower cost (Snow et al., 2008). It has also been used in MT evaluation (Callison-Burch, 2009), though that evaluation used reference translations.

For 70 queries, the top 25 ranked sentences in both the production and research MT versions were evaluated. Each sentence was judged for both relevance and perceived adequacy by 5 annotators, for a total of 35,000 individual judgments. As is standard, some of the judgments were filtered due to noise by using the percent of time that an annotator disagreed with all other annotators, and the relative time spent on a given annotation. The percent of sentences with majority agreement was 91% for relevance and 72% for perceived adequacy.

6.3 Intrinsic Evaluation

Annotators were presented with an Arabic sentence with a single token highlighted, and asked whether the token was a "content word" or not. Then annotators were asked to decide which of two translations (in random order) translated the highlighted Arabic word best, or whether they were equal. In total, 150 sentences were judged by annotators with knowledge of Arabic. For both questions, kappa agreement was moderate.

7 Results

Table 2 shows how many errors were found by the error detection algorithm for each genre. Not surprisingly, more errors are detected in the speech genres (84 and 105 errors per 1,000 tokens) than in formal text (56 errors per 1,000 tokens). We attribute the large difference between broadcast news and broadcast conversa-

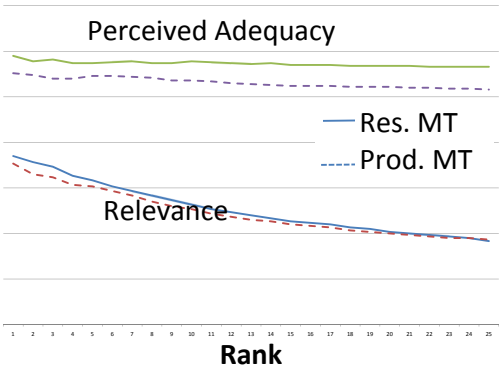


Figure 1: Average normalized cumulative sentence perceived adequacy and relevance versus rank of the sentence, by the ranking heuristic.

tion to the large number of short sentences without content words in informal speech (such as "hello", "thank you", etc.).

7.1 Perceived MT Adequacy

The research MT significantly outperformed the production MT in perceived adequacy (according to ANOVA with p=0.001). Of the production MT translations, 58% were considered "more than half" or "all" understandable, whereas 69% of the research MT were. Overall, retranslation increased perceived adequacy in 17% of the sentences, and decreased it in only 5% of sentences.

7.2 Ranking Algorithm

Figure 1 show the average cumulative sentence relevance and perceived adequacy, as ranked by the error detection algorithm. In other words, at each rank i, the average relevance (or perceived adequacy) of sentences $(1 - i)$ was calculated. On the perceived adequacy chart, the research MT system consistently outperforms the production MT system by a statistically significant margin. For relevance, the research MT curve is only marginally higher than the production MT curve.

The shape of the relevance curves shows that ranking sentences by a simple bilingual bag-of-words model did affect sentence relevance, since sentences that are higher ranked have higher cumulative average relevance. By ranking sentences with a basic relevance model, we were able to focus the scarce MT resources on sentences that are most likely to help the CLQA task. This underscores the importance of using the task context to guide MT error detection, especially in the case of time-constrained MT.

	Relevance				
	⇑	Same	⇓	No maj./ Don't know	
MT ⇑	20	201	9	56	17%
MT same	93	919	72	212	78%
MT ⇓	2	56	4	28	5%
	7%	70%	5%	18%	

Table 3: The relationship between changes in perceived adequacy and changes in relevance.

7.3 CLQA Relevance

Annotators judged 14.5% of the production MT sentences relevant. After retranslation, the overall number of sentences considered relevant increased to 14.7%. Although the overall numbers are similar, the relevance of many individual sentences did change. Table 3 shows the results of comparing annotations on the original MT with annotations on the retranslated MT. Relevance was classified as ⇑ or ⇓ by comparing the majority judgment of the production MT to the research MT. Changes in MT were based on comparing the average rating of both versions, with a tolerance of 1.0.

Of the sentences with better perceived MT, 7% increased in relevance, and 3% decreased in relevance. When the retranslated sentence was considered worse, there was a 2% increased in relevance and a 4% decrease. In other words, when retranslation had a positive effect, it more often led to increased relevance. However, the impact of retranslation was mixed, and none of the changes was statistically significant.

7.4 Intrinsic Evaluation

While the extrinsic evaluation focused on the impact on CLQA relevance, the goal of the intrinsic evaluation was to measure the precision of the error detection algorithm, and whether retranslation addressed the detected errors.

Of the 82% of sentences where both judges agreed, 89% of the detected errors were considered content words. All of the OOV tokens were content words (except for one disagree-

ment). Surprisingly, for the errors involving content words, 60% of the time both systems were judged the same with regard to the highlighted error. The research system was better 39% of the time, and the original was better only 1% of the time (excluding 26% disagreements).

8 Discussion

The CLQA evaluation was based on three hypotheses:

- That we could detect errors in MT with high precision.
- That retranslating errorful sentences with a much better MT system would correct the errors we detected.
- That correcting errors would cause some sentences to become relevant which were not previously relevant, as in (Ma and McKeown, 2009).

The intrinsic evaluation confirmed that we can identify content word deletions in MT with high precision, thus validating the first hypothesis. However, detecting the errors and retranslating them did not lead to large improvements in CLQA relevance – the impact of increased perceived adequacy on relevance was mixed and not significant. The intrinsic evaluation explains this negative result: even though the retranslated sentences were judged significantly better, the retranslation only corrected the detected error 39% of the time. In other words, the better research MT system was making many of the same mistakes as the production MT system, despite using syntactic features and a much deeper search space during decoding. Since the second hypothesis did not hold, we need to improve our error correction algorithm before we can tell whether the third hypothesis holds.

This result directly motivates the need for targeted error correction of MT. Automatic MT post-editing has been successfully used for selecting determiners (Knight and Chander, 1994), reinserting deleted verbs (Ma and McKeown, 2009), correcting NE translations (Parton et al., 2008), and lexical substitutions (Elming, 2006). Since Arabic and English word order differ significantly, straightforward re-insertion of the deleted words is not sufficient for error correction, so we are currently working on more sophisticated post-editing techniques.

9 Conclusions

We presented a novel online algorithm for detecting MT errors in the context of a question, and a heuristic for ranking MT errors by their potential impact on the CLQA task. The error detection algorithm focused on content word deletion, which has previously been shown to be a significant problem in SMT. The algorithm is generally applicable to any MT system that produces word or phrase alignments for its output and any language pair that can be POS-tagged, and it is more fine-grained and covers more types of errors than previous work. It was able to detect errors in Arabic-English MT across multiple text and speech genres, and the intrinsic evaluation showed that the large majority of tokens flagged as errors were indeed content words.

The large-scale CLQA evaluation confirmed that the slower research MT system was significantly better than the production MT system. Relevance judgments showed that the ranking component was crucial for directing scarce MT resources wisely, as the higher-ranked sentences were most likely to be relevant to the query, and therefore most likely to benefit the CLQA system by being retranslated.

Although we correctly identified MT errors, retranslating the sentences with the errors had a negligible effect on CLQA relevance. This unexpected result may be explained by the fact that only 39% of the errors were actually corrected by the research MT system, so re-translation was not a good approach for error correction. We are currently working on correcting content word deletion in MT via post-editing.

Acknowledgments The authors are grateful to Radu Florian, Salim Roukos, Vittorio Castelli, Dan Bikel and the whole GALE IBM team for providing the experimental testbed, including the CLQA and MT systems. This research was partially supported by DARPA grant HR0011-08-C-0110.

References

Boschee, Elizabeth, Marjorie Freedman, Roger Bock, John Graettinger, and Ralph Weischedel. 2010. Error analysis and future directions for distillation. In *GALE book (in preparation)*.

Buckwalter, Tim. 2002. Buckwalter arabic morphological analyzer. *Linguistic Data Consortium. (LDC2002L49)*.

Callison-Burch, Chris. 2009. Fast, cheap, and creative: evaluating translation quality using amazon's mechanical turk. In *EMNLP '09*, pages 286–295, Morristown, NJ, USA. Association for Computational Linguistics.

Daumé, III, Hal and Daniel Marcu. 2006. Bayesian query-focused summarization. In *ACL*, pages 305–312, Morristown, NJ, USA. Association for Computational Linguistics.

Elming, Jakob. 2006. Transformation-based corrections of rule-based mt. In *EAMT-2006: 11th Annual Conference of the European Association for Machine Translation*, pages 219–226.

Hermjakob, Ulf, Kevin Knight, and Hal Daumé III. 2008. Name translation in statistical machine translation - learning when to transliterate. In *Proceedings of ACL-08: HLT*, pages 389–397, Columbus, Ohio, June. Association for Computational Linguistics.

Ittycheriah, Abraham and Salim Roukos. 2007. Direct translation model 2. In Sidner, Candace L., Tanja Schultz, Matthew Stone, and ChengXiang Zhai, editors, *HLT-NAACL*, pages 57–64. The Association for Computational Linguistics.

Ji, Heng and Ralph Grishman. 2007. Collaborative entity extraction and translation. In *International Conference on Recent Advances in Natural Language Processing*.

Kirchhoff, Katrin, Owen Rambow, Nizar Habash, and Mona. Diab. 2007. Semi-automatic error analysis for large-scale statistical machine translation systems. In *Proceedings of the Machine Translation Summit IX (MT-Summit IX)*.

Knight, Kevin and Ishwar Chander. 1994. Automated postediting of documents. In *AAAI*, pages 779–784.

Li, Chi-Ho, Dongdong Zhang, Mu Li, Ming Zhou, and Hailei Zhang. 2008. An empirical study in source word deletion for phrase-based statistical machine translation. In *StatMT '08: Proceedings of the Third Workshop on Statistical Machine Translation*, pages 1–8, Morristown, NJ, USA. Association for Computational Linguistics.

Ma, Wei-Yun and Kathleen McKeown. 2009. Where's the verb?: correcting machine translation during question answering. In *ACL-IJCNLP '09: Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 333–336, Morristown, NJ, USA. Association for Computational Linguistics.

Menezes, Arul and Chris Quirk. 2008. Syntactic models for structural word insertion and deletion. In *EMNLP '08*, pages 735–744, Morristown, NJ, USA. Association for Computational Linguistics.

Papineni, Kishore, Salim Roukos, Todd Ward, and Wei jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL*, pages 311–318.

Parton, Kristen, Kathleen R. McKeown, James Allan, and Enrique Henestroza. 2008. Simultaneous multilingual search for translingual information retrieval. In *CIKM 08*, pages 719–728, New York, NY, USA. ACM.

Parton, Kristen, Kathleen R. McKeown, Bob Coyne, Mona T. Diab, Ralph Grishman, Dilek Hakkani-Tür, Mary Harper, Heng Ji, Wei Yun Ma, Adam Meyers, Sara Stolbach, Ang Sun, Gokhan Tur, Wei Xu, and Sibel Yaman. 2009. Who, what, when, where, why?: comparing multiple approaches to the cross-lingual 5w task. In *ACL-IJCNLP '09*, pages 423–431, Morristown, NJ, USA. Association for Computational Linguistics.

Snow, Rion, Brendan O'Connor, Daniel Jurafsky, and Andrew Y. Ng. 2008. Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks. In *EMNLP '08*, pages 254–263, Morristown, NJ, USA. Association for Computational Linguistics.

Vilar, David, Jia Xu, Luis Fernando D'Haro, and Hermann Ney. 2006. Error analysis of machine translation output. In *International Conference on Language Resources and Evaluation*, pages 697–702, Genoa, Italy, May.

Zhang, Yuqi, Evgeny Matusov, and Hermann Ney. 2009. Are unaligned words important for machine translation ? In *Conference of the European Association for Machine Translation*, pages 226–233, Barcelona, March.

The Role of Queries in Ranking Labeled Instances Extracted from Text

Marius Paşca
Google Inc.
mars@google.com

Abstract

A weakly supervised method uses anonymized search queries to induce a ranking among class labels extracted from unstructured text for various instances. The accuracy of the extracted class labels exceeds that of previous methods, over evaluation sets of instances associated with Web search queries.

1 Introduction

Classes pertaining to unrestricted domains (e.g., *west african countries*, *science fiction films*, *slr cameras*) and their instances (*cape verde*, *avatar*, *canon eos 7d*) play a disproportionately important role in Web search. They occur prominently in Web documents and among search queries submitted most frequently by Web users (Jansen et al., 2000). They also serve as building blocks in formal representation of human knowledge, and are useful in a variety of text processing tasks.

Recent work on offline acquisition of fine-grained, labeled classes of instances applies manually-created (Banko et al., 2007; Talukdar et al., 2008) or automatically-learned (Snow et al., 2006) extraction patterns to large document collections. Although various methods exploit additional textual resources to increase accuracy (Van Durme and Paşca, 2008) and coverage (Talukdar et al., 2008), some of the extracted class labels are inevitably less useful (*works*) or spurious (*car makers*) for an associated instance (*avatar*). In Web search, the relative ranking of documents returned for a query directly affects the outcome of the search. Similarly, the relative ranking among class labels extracted for a given instance influences any applications using the labels.

Our paper proposes the use of features other than those computed over the underlying document collection, such as the frequency of co-occurrence or diversity of extraction patterns producing a given pair (Etzioni et al., 2005), to determine the relative ranking of various class labels, given a class instance. Concretely, the method takes advantage of the co-occurrence of a class label and an instance within search queries from anonymized query logs. It re-ranks lists of class labels produced for an instance by standard extraction patterns, to promote class labels that co-occur with the instance. This corresponds to a soft ranking approach, focusing on the ranking of candidate extractions such as the less relevant ones are ranked lower, as opposed to removed when deemed unreliable based on various clues.

By using queries in ranking, the ranked lists of class labels available for various instances are instrumental in determining the classes to which given sets of instances belong. The accuracy of the class labels exceeds that of previous work, over evaluation sets of instances associated with Web search queries. The results confirm the usefulness of the extracted IsA repository, which remains general-purpose and is not tailored to any particular task.

2 Instance Class Ranking

2.1 Extraction of Instances and Classes

The initial extraction of labeled instances relies on hand-written patterns from (Hearst, 1992), widely used in work on extracting hierarchies from text (Snow et al., 2006; Ponzetto and Strube,

Coling 2008: Poster Volume, pages 955–962,
Beijing, August 2010

2007):

⟨[..] $\mathcal{C}$ [such as|including] $\mathcal{I}$ [and|,|.]⟩,

where $\mathcal{I}$ is a potential instance (e.g., *diderot*) and $\mathcal{C}$ is a potential class label (e.g., *writers*).

Following (Van Durme and Paşca, 2008), the boundaries of potential class labels $\mathcal{C}$ are approximated from the part-of-speech tags of the sentence words, whereas the boundaries of instances $\mathcal{I}$ are identified by checking that $\mathcal{I}$ occurs as an entire query in query logs. Since users type many queries in lower case, the collected data is converted to lower case.

When applied to inherently-noisy Web documents, the extraction patterns may produce irrelevant extractions (Kozareva et al., 2008). Causes of errors include incorrect detection of possible enumerations, as in *companies such as Procter and Gamble* (Downey et al., 2007); incorrect estimation of the boundaries of class labels, due to incorrect attachment as in *years* from *on a limited number of vehicles over the past few years, including the Chevrolet Corvette*; subjective (*famous actors*) (Hovy et al., 2009), relational (*competitors*, *nearby landmarks*) and otherwise less useful (*others*, *topics*) class labels; or questionable source sentences, as in *Large mammals such as deer and wild turkeys can be [..]* (Van Durme and Paşca, 2008).

As a solution, recent work uses additional evidence, as a means to filter the pairs extracted by patterns, thus trading off coverage for higher precision. The repository extracted from a similarly-sized Web document collection using the same initial extraction patterns as here, after a weighted intersection of pairs extracted with patterns and clusters of distributionally similar phrases, contains a total of 9,080 class labels associated with 263,000 instances in (Van Durme and Paşca, 2008). Subsequent extensions of the repository, using data derived from tables within Web documents, increase instance coverage and induce a ranking among class labels of each instance, but do not increase the number of class labels (Talukdar et al., 2008). Due to aggressive filtering, the resulting number of class labels is higher than the often-small sets of entity types studied previously, but may still be insufficient given the diversity of Web search queries.

2.2 Ranking of Classes per Instance

As an alternative, the soft ranking approach proposed here attempts to rank better class labels higher, without necessarily removing class labels deemed incorrect according to various criteria. For each instance $\mathcal{I}$, the associated class labels are ranked in the following stages:

1) Apply the scoring formula below, resulting in a ranked list of class labels $\mathrm{L}_1(\mathcal{I})$:

$$Score(\mathcal{I},\mathcal{C}) = Size(\{Pattern(\mathcal{I},\mathcal{C})\})^2 \times Freq(\mathcal{I},\mathcal{C})$$

Thus, a class label $\mathcal{C}$ is deemed more relevant for an instance $\mathcal{I}$ if $\mathcal{C}$ is extracted by multiple extraction patterns and its original frequency-based score is higher.

2) For each term within any class label from $\mathrm{L}_1(\mathcal{I})$, compute a score equal to the frequency sum of the term within anonymized queries containing the instance $\mathcal{I}$ as a prefix, and the term anywhere else in the queries. Each class label is assigned the geometric mean of the scores of its terms, after ignoring stop words. The class labels are ranked according to the means, resulting in a ranked list $\mathrm{L}_2(\mathcal{I})$. In case of ties, $\mathrm{L}_2(\mathcal{I})$ preserves the relative ranking from $\mathrm{L}_1(\mathcal{I})$. Thus, a class label is deemed more relevant if its individual terms occur in popular queries containing the instance.

3) Compute a merged ranked list of class labels out of the ranked lists $\mathrm{L}_1(\mathcal{I})$ and $\mathrm{L}_2(\mathcal{I})$, by sorting the class labels in decreasing order of the inverse of the average rank, computed with the following formula:

$$MergedScore(\mathcal{C}) = \frac{2}{Rank(\mathcal{C},L_1) + Rank(\mathcal{C},L_2)}$$

where 2 is the number of input lists of class labels, and Rank($\mathcal{C}$, L_i) is the rank of $\mathcal{C}$ in the list L_i of class labels computed for the corresponding input instance. The rank is set to 1000, if $\mathcal{C}$ is not present in the list L_i. By using only the relative ranks of the class labels within the input lists, and not on their scores, the outcome of the merging is less sensitive to how class labels of a given instance are scored within the IsA repository. In case of ties, the scores of the class labels from $\mathrm{L}_1(\mathcal{I})$ serve as a secondary ranking criterion.

Note that the third stage is introduced because relying on query logs to estimate the relevance of

class labels exposes the ranking method to significant noise. On one hand, arguably useful class labels (e.g., *authors*) may not occur in queries along with the respective instances (*diderot*). On the other hand, for each query containing an instance and (part of) useful class labels, there are many other queries containing, e.g., attributes (*diderot biography* or *diderot beliefs*) or the name of a book in the query *diderot the nun*. Therefore, the ranked lists $L_2(\mathcal{I})$ may be too noisy to be used directly as rankings of the class labels for $\mathcal{I}$.

3 Experimental Setting

3.1 Textual Data Sources

The acquisition of the IsA repository relies on unstructured text available within Web documents and search queries. The collection of queries is a sample of 50 million unique, fully-anonymized queries in English submitted by Web users in 2009. Each query is accompanied by its frequency of occurrence in the logs. The document collection consists of a sample of 100 million documents in English. The textual portion of the documents is cleaned of HTML, tokenized, split into sentences and part-of-speech tagged using the TnT tagger (Brants, 2000).

3.2 Experimental Runs

The experimental runs correspond to different methods for extracting and ranking pairs of an instance and a class:

- as available in the repository from (Talukdar et al., 2008), which is collected from a document collection similar in size to the one used here plus a collection of Web tables, in a run denoted R_g;
- from the repository extracted here, with class labels of an instance ranked based on the frequency and the number of extraction patterns (see $Score(\mathcal{I}, \mathcal{C})$ in Section 2), in run R_s;
- from the repository extracted here, with class labels of an instance ranked based on the $MergedScore$ from Section 2, in run R_u.

3.3 Evaluation Procedure

The manual evaluation of open-domain information extraction output is time consuming (Banko et al., 2007). Fortunately, it is possible to implement an automatic evaluation procedure for ranked lists of class labels, based on existing resources and systems. Assume that a gold standard is available, containing gold class labels that are each associated with a gold set of their instances. The creation of such gold standards is discussed later. Based on the gold standard, the ranked lists of class labels available within an IsA repository can be automatically evaluated as follows. First, for each gold label, the ranked lists of class labels of individual gold instances are retrieved from the IsA repository. Second, the individual retrieved lists are merged into a ranked list of class labels, associated with the gold label. The merged list is computed using an extension of the $MergedScore$ formula described earlier in Section 2. Third, the merged list is compared against the gold label, to estimate the accuracy of the merged list. Intuitively, a ranked list of class labels is a better approximation of a gold label, if class labels situated at better ranks in the list are closer in meaning to the gold label.

3.4 Evaluation Metric

Given a gold label and a list of class labels, if any, derived from the IsA repository, the rank of the highest class label that matches the gold label determines the score assigned to the gold label, in the form of the reciprocal rank, max(1/rank$_{match}$). Thus, if the gold label matches a class label at rank 1, 2, 3, 4 or 5 in the computed list, the gold label receives a score of 1, 0.5, 0.33, 0.25 or 0.2 respectively. The score is 0 if the gold label does not match any of the top 20 class labels. The overall score over the entire set of gold labels is the mean reciprocal rank (MRR) score over all gold labels from the set. Two types of MRR scores are automatically computed:

- MRR$_f$ considers a gold label and a class label to match if they are identical;
- MRR$_p$ considers a gold label and a class label to match if one or more of their tokens that are not stop words are identical.

During matching, all string comparisons are case-insensitive, and all tokens are first converted to their singular form (e.g., *european countries* to *european country*) when available, by using WordNet's morphological routines. Thus, *insurance carriers* and *insurance companies* are considered to not match in MRR_f scores, but match in MRR_p scores, whereas *insurance companies* and *insurance company* match in both MRR_f and MRR_p scores. Note that both MRR_f and MRR_p scores fail to give any credit to arguably valid and useful class labels, such as *insurers* for the gold label *insurance carriers*, or *asian nations* for the gold label *asia countries*. On the other hand, MRR_p scores may give credit to less relevant class labels, such as *insurance policies* for the gold label *insurance carriers*. Therefore, MRR_p is an approximate, and MRR_f is a conservative, lower-bound estimate of the actual usefulness of the computed ranked lists of class labels as approximations of the semantics of the gold labels.

4 Evaluation Results

4.1 Evaluation Sets of Queries

A random sample of anonymized, class-seeking queries (e.g., *video game characters* or *smartphone*) submitted by Web users to Google Squared [1] over a 30-day interval is filtered, to remove queries for which Google Squared returns fewer than 10 instances at the time of the evaluation. The resulting evaluation set of queries, denoted Q_e, contains 807 queries, each associated with a ranked list of between 10 and 100 instances automatically extracted by Google Squared.

Since the instances available as input for each query as part of Q_e are automatically extracted, they may (e.g., *acorn a7000*) or may not (e.g., *konrad zuse*) be true instances of the respective queries (e.g., *computers*). A second evaluation set Q_m is assembled as a subset of 40 queries from Q_e, such that the instances available for each query in Q_m are correct. For this purpose, each instance returned by Google Squared for the 40 queries from Q_m is reviewed by at least three human annotators. Instances deemed highly relevant (out of 5 possible grades) with high inter-annotator agreement are retained. As a result, the 40 queries from Q_m are associated with between 8 and 33 human-validated instances.

Query Set: Sample of Queries
Q_e (807 queries): 2009 movies, amino acids, asian countries, bank, board games, buildings, capitals, chemical functional groups, clothes, computer language, dairy farms near modesto ca, disease, egyptian pharaohs, eu countries, french presidents, german islands, hawaiian islands, illegal drugs, irc clients, lakes, macintosh models, mobile operator india, nba players, nobel prize winners, orchids, photo editors, programming languages, renaissance artists, roller costers, science fiction tv series, slr cameras, soul singers, states of india, taliban members, thomas edison inventions, u.s. presidents, us president, water slides
Q_m (40 queries): actors, airlines, birds, cars, celebrities, computer languages, digital camera, dog breeds, drugs, endangered animals, european countries, fruits, greek gods, horror movies, ipods, names, netbooks, operating systems, park slope restaurants, presidents, ps3 games, religions, renaissance artists, rock bands, universities, university, vitamins

Table 1: Size and composition of evaluation sets of queries associated with non-filtered (Q_e) or manually-filtered (Q_m) instances

Table 1 shows a sample of the queries from Q_e and queries from Q_m. A small number of queries are slight lexical variations of one another, such as *u.s. presidents* and *us presidents* in Q_e, or *universities* and *university* in Q_m. In general, however, the sets cover a wide range of domains of interest, including entertainment for *2009 movies* and *rock bands*; biology for *endangered animals* and *amino acids*; geography for *asian countries* and *hawaiian islands*; food for *fruits*; history for *egyptian pharaohs* and *greek gods*; health for *drugs* and *vitamins*; and technology for *photo editors* and *ipods*. Some of the queries from Table 1 are specific enough that computing them exactly,

[1] Google Squared (http://www.google.com/squared) is a Web search tool taking as input class-seeking queries (e.g., *insurance companies*) and returning lists of instances (e.g., *allstate*, *state farm insurance*), along with attributes (e.g., *industry*, *headquarters*) and values for each instance.

	Accuracy											
I_Q	3			5			10			15		
C_I	5	10	20	5	10	20	5	10	20	5	10	20
MRR_f computed over Q_e:												
R_g	0.106	0.112	0.112	0.121	0.122	0.123	0.131	0.135	0.127	0.134	0.132	0.127
R_s	0.186	0.195	0.198	0.198	0.207	0.210	0.204	0.214	0.218	0.206	0.216	0.221
R_u	0.202	0.211	0.216	0.232	0.238	0.244	0.245	0.255	0.257	0.245	0.252	0.254
MRR_p computed over Q_e:												
R_g	0.390	0.399	0.394	0.420	0.420	0.413	0.443	0.443	0.435	0.439	0.431	0.425
R_s	0.489	0.495	0.495	0.517	0.528	0.529	0.541	0.553	0.557	0.551	0.557	0.557
R_u	0.520	0.531	0.533	0.564	0.573	0.578	0.590	0.601	0.602	0.598	0.603	0.601
MRR_f computed over Q_m:												
R_g	0.284	0.289	0.295	0.305	0.327	0.322	0.320	0.335	0.335	0.334	0.328	0.337
R_s	0.406	0.436	0.442	0.431	0.447	0.466	0.467	0.470	0.501	0.484	0.501	0.554
R_u	0.423	0.426	0.429	0.436	0.483	0.508	0.500	0.526	0.530	0.520	0.540	0.524
MRR_p computed over Q_m:												
R_g	0.507	0.517	0.531	0.495	0.509	0.518	0.555	0.553	0.550	0.563	0.561	0.572
R_s	0.667	0.662	0.660	0.675	0.677	0.699	0.702	0.695	0.716	0.756	0.765	0.787
R_u	0.711	0.703	0.680	0.734	0.731	0.748	0.733	0.797	0.782	0.799	0.834	0.819

Table 2: Accuracy of instance set labeling, as full-match (MRR_f) or partial-match (MRR_p) scores over the evaluation sets of queries associated with non-filtered instances (Q_e) or manually-filtered instances (Q_m), for various experimental runs (I_Q=number of instances available in the input evaluation sets that are used for retrieving class labels; C_I=number of class labels retrieved from IsA repository per input instance)

even from a comprehensive, perfect list of extracted instance, would be very difficult whether done automatically or manually. Examples of such queries are *dairy farms near modesto ca* and *science fiction tv series*, but also *mobile operator india* (phrase expressed as keywords) in Q_e, or *park slope restaurants* (specific location) in Q_m.

Access to a system such as Google Squared is useful, but not necessary to conduct the evaluation. Given other sets of queries, it is straightforward, albeit time consuming, to create evaluation sets similar to Q_m, by manually compiling correct instances, for each selected query or concept.

Following the general evaluation procedure, each query from the sets Q_e and Q_m acts as a gold class label associated with its set of instances. Given a query and its instances $\mathcal{I}$ from the evaluation sets Q_e or Q_m, we compute merged, ranked lists of class labels, by merging the ranked lists of class labels available in the underlying IsA repository for each instance $\mathcal{I}$. The evaluation compares the merged lists of class labels, on one hand, and the corresponding queries from Q_e or Q_m, on the other hand.

4.2 Accuracy of Class Labels

Table 2 summarizes results from comparative experiments, quantifying a) horizontally, the impact of alternative parameter settings on the computed lists of class labels; and b) vertically, the comparative accuracy of the experimental runs over the query sets. The experimental parameters are the number of input instances from the evaluation sets that are used for retrieving class labels, I_Q, set to 3, 5, 10 and 15; and the number of class labels retrieved per input instance, C_I, set to 5, 10 and 20.

The scores over Q_m are higher than those over Q_e, confirming the intuition that the higher-quality input set of instances available in Q_m relative to Q_e should lead to higher-quality class labels for the corresponding queries. When I_Q is fixed, increasing C_I leads to small, if any, score improvements. Conversely, when C_I is fixed,

even small values of I_Q, such as 3 or 5 (that is, very small sets of instances provided as input) produce scores that are competitive with those obtained with a higher value like. This suggests that useful class labels can be generated even in extreme scenarios, where the number of instances available as input is as small as 3 or 5.

For most combinations of parameter settings and on both query sets, run R_u produces the highest scores. In particular, when I_Q is set to 10 and C_I to 20, run R_u identifies the original query as an exact match among the top four class labels returned; and as a partial match among the top two class labels returned, as an average over the Q_e set. In this case, the original query is identified at ranks 1, 2, 3, 4 and 5 for 16.8%, 8.7%, 6.1%, 3.7% and 1.7% of the queries, as an exact match; and for 48.8%, 14.2%, 6.1%, 3.6% and 1.9% respectively, as a partial match. The corresponding MRR_f score of 0.257 over the Q_e set obtained with run R_u is higher than with run R_s, and much higher than with run R_g. In all experiments, the higher scores of R_u can be attributed to higher coverage of class labels, relative to R_g; and higher-quality lists of class labels, relative to R_s but also to R_g, despite the fact that R_g combines high-precision seed data with using both unstructured and structured text as sources of class labels (cf. (Talukdar et al., 2008)). Among combinations of parameter settings described in Table 2, values around 15 for I_Q and 20 for C_I give the highest scores over both Q_e and Q_m.

5 Related Work

5.1 Extraction of IsA Repositories

Knowledge including instances and classes can be manually compiled by experts (Fellbaum, 1998) or collaboratively by non-experts (Singh et al., 2002). Alternatively, classes of instances acquired automatically from text are potentially less expensive to acquire, maintain and grow, and their coverage and scope are theoretically bound only by the size of the underlying data source. Existing methods for extracting classes of instances acquire sets of instances that are each either unlabeled (Wang and Cohen, 2008; Pennacchiotti and Pantel, 2009; Lin and Wu, 2009), or associated with a class label (Pantel and Pennacchiotti, 2006; Banko et al., 2007; Wang and Cohen, 2009). When associated with a class label, the sets of instances may be organized as flat sets or hierarchically, relative to existing hierarchies such as WordNet (Snow et al., 2006) or the category network within Wikipedia (Wu and Weld, 2008; Ponzetto and Navigli, 2009). Semi-structured text was shown to be a complementary resource to unstructured text, for the purpose of extracting relations from Web documents (Cafarella et al., 2008).

The role of anonymized query logs in Web-based information extraction has been explored in the tasks of class attribute extraction (Paşca and Van Durme, 2007) and instance set expansion (Pennacchiotti and Pantel, 2009). Our method illustrates the usefulness of queries considered in isolation from one another, in ranking class labels in extracted IsA repositories.

5.2 Labeling of Instance Sets

Previous work on generating relevant labels, given sets or clusters of items, focuses on scenarios where the items within the clusters are descriptions of, or full-length documents within document collections. The documents are available as a flat set (Cutting et al., 1993; Carmel et al., 2009) or are hierarchically organized (Treeratpituk and Callan, 2006). Relying on semi-structured content assembled manually as part of the structure of Wikipedia articles, such as article titles or categories, the method introduced in (Carmel et al., 2009) derives labels for clusters containing 100 full-length documents each. In contrast, our method relies on IsA relations automatically extracted from unstructured text within arbitrary Web documents, and computes labels given textual input that is orders of magnitude smaller, i.e., around 10 phrases (instances). The experiments described in (Carmel et al., 2009) assign labels to one of 20 sets of newsgroup documents from a standard benchmark. Each set of documents is associated with a higher-level, coarse-grained label used as a gold label against which the generated labels are compared. In comparison, our experiments compute text-derived class labels for finer-grained, often highly-specific gold labels.

Reducing the granularity of the items to be labeled from full documents to condensed document descriptions, (Geraci et al., 2006) submits arbitrary search queries to external Web search engines. It organizes the top 200 returned Web documents into clusters, by analyzing the text snippets associated with each document in the output from the search engines. Any words and phrases from the snippets may be selected as labels for the clusters, which in general leads to labels that are not intended to capture any classes that may be associated to the query. For example, labels of clusters generated in (Geraci et al., 2006) include *armstrong ceilings*, *italia*, *armstrong sul sito* and *louis jazz* for the query *armstrong*; and *madonnaweb*, *music*, *madonna online* and *madonna* itself for the query *madonna*. The amount of text available as input for the purpose of labeling is at least two orders of magnitude larger than in our method, and the task of selecting any phrases as labels, as opposed to selecting only labels that correspond to classes, is more relaxed and likely easier.

Another approach specifically addresses the problem of generating labels for sets of instances, where the labels are extracted from unstructured text. In (Pantel and Ravichandran, 2004), given a collection of news articles that is both cleaner and smaller than Web document collections, a syntactic parser is applied to document sentences in order to identify and exploit syntactic dependencies for the purpose of selecting candidate class labels. Such methods are comparatively less applicable to Web document collections, due to scalability issues associated with parsing a large set of Web documents of variable quality. Moreover, the class labels generated in (Pantel and Ravichandran, 2004) tend to be rather coarse-grained. For example, the top labels generated for a set of Chinese universities (*qinghua university*, *fudan university*, *beijing university*) are *university*, *institution*, *stock-holder*, *college* and *school*.

6 Conclusion

The method presented in this paper produces an IsA repository whose class labels have higher coverage and accuracy than with recent methods operating on document collections. This is done by injecting useful ranking signals from inherently-noisy queries, rather than making binary, coverage-reducing quality decisions on the extracted data. Current work investigates the usefulness of the extracted class labels in the generation of flat or hierarchical query refinements for class-seeking queries.

Acknowledgments

The author thanks Randolph Brown for assistance in assembling the evaluation sets of class-seeking queries.

References

Banko, M., Michael J Cafarella, S. Soderland, M. Broadhead, and O. Etzioni. 2007. Open information extraction from the Web. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI-07)*, pages 2670–2676, Hyderabad, India.

Brants, T. 2000. TnT - a statistical part of speech tagger. In *Proceedings of the 6th Conference on Applied Natural Language Processing (ANLP-00)*, pages 224–231, Seattle, Washington.

Cafarella, M., A. Halevy, D. Wang, E. Wu, and Y. Zhang. 2008. WebTables: Exploring the power of tables on the Web. In *Proceedings of the 34th Conference on Very Large Data Bases (VLDB-08)*, pages 538–549, Auckland, New Zealand.

Carmel, D., H. Roitman, and N. Zwerding. 2009. Enhancing cluster labeling using Wikipedia. In *Proceedings of the 32nd ACM Conference on Research and Development in Information Retrieval (SIGIR-09)*, pages 139–146, Boston, Massachusetts.

Cutting, D., D. Karger, and J. Pedersen. 1993. Constant interaction-time scatter/gather browsing of very large document collections. In *Proceedings of the 16th ACM Conference on Research and Development in Information Retrieval (SIGIR-93)*, pages 126–134, Pittsburgh, Pennsylvania.

Downey, D., M. Broadhead, and O. Etzioni. 2007. Locating complex named entities in Web text. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI-07)*, pages 2733–2739, Hyderabad, India.

Etzioni, O., M. Cafarella, D. Downey, A. Popescu, T. Shaked, S. Soderland, D. Weld, and A. Yates. 2005. Unsupervised named-entity extraction from the Web: an experimental study. *Artificial Intelligence*, 165(1):91–134.

Fellbaum, C., editor. 1998. *WordNet: An Electronic Lexical Database and Some of its Applications*. MIT Press.

Geraci, F., M. Pellegrini, M. Maggini, and F. Sebastiani. 2006. Cluster generation and cluster labelling for Web snippets: A fast and accurate hierarchical solution. In *Proceedings of the 13th Conference on String Processing and Information Retrieval (SPIRE-06)*, pages 25–36, Glasgow, Scotland.

Hearst, M. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th International Conference on Computational Linguistics (COLING-92)*, pages 539–545, Nantes, France.

Hovy, E., Z. Kozareva, and E. Riloff. 2009. Toward completeness in concept extraction and classification. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP-09)*, pages 948–957, Singapore.

Jansen, B., A. Spink, and T. Saracevic. 2000. Real life, real users, and real needs: a study and analysis of user queries on the Web. *Information Processing and Management*, 36(2):207–227.

Kozareva, Z., E. Riloff, and E. Hovy. 2008. Semantic class learning from the web with hyponym pattern linkage graphs. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics (ACL-08)*, pages 1048–1056, Columbus, Ohio.

Lin, D. and X. Wu. 2009. Phrase clustering for discriminative learning. In *Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics (ACL-IJCNLP-09)*, pages 1030–1038, Singapore.

Paşca, M. and B. Van Durme. 2007. What you seek is what you get: Extraction of class attributes from query logs. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI-07)*, pages 2832–2837, Hyderabad, India.

Pantel, P. and M. Pennacchiotti. 2006. Espresso: Leveraging generic patterns for automatically harvesting semantic relations. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (COLING-ACL-06)*, pages 113–120, Sydney, Australia.

Pantel, P. and D. Ravichandran. 2004. Automatically labeling semantic classes. In *Proceedings of the 2004 Human Language Technology Conference (HLT-NAACL-04)*, pages 321–328, Boston, Massachusetts.

Pennacchiotti, M. and P. Pantel. 2009. Entity extraction via ensemble semantics. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP-09)*, pages 238–247, Singapore.

Ponzetto, S. and R. Navigli. 2009. Large-scale taxonomy mapping for restructuring and integrating Wikipedia. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI-09)*, pages 2083–2088, Pasadena, California.

Ponzetto, S. and M. Strube. 2007. Deriving a large scale taxonomy from Wikipedia. In *Proceedings of the 22nd National Conference on Artificial Intelligence (AAAI-07)*, pages 1440–1447, Vancouver, British Columbia.

Singh, P., T. Lin, E. Mueller, G. Lim, T. Perkins, and W. Zhu. 2002. Open Mind Common Sense: Knowledge acquisition from the general public. In *Proceedings of the ODBASE Conference (ODBASE-02)*, pages 1223–1237.

Snow, R., D. Jurafsky, and A. Ng. 2006. Semantic taxonomy induction from heterogenous evidence. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (COLING-ACL-06)*, pages 801–808, Sydney, Australia.

Talukdar, P., J. Reisinger, M. Paşca, D. Ravichandran, R. Bhagat, and F. Pereira. 2008. Weakly-supervised acquisition of labeled class instances using graph random walks. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing (EMNLP-08)*, pages 582–590, Honolulu, Hawaii.

Treeratpituk, P. and J. Callan. 2006. Automatically labeling hierarchical clusters. In *Proceedings of the 7th Annual Conference on Digital Government Research (DGO-06)*, pages 167–176, San Diego, California.

Van Durme, B. and M. Paşca. 2008. Finding cars, goddesses and enzymes: Parametrizable acquisition of labeled instances for open-domain information extraction. In *Proceedings of the 23rd National Conference on Artificial Intelligence (AAAI-08)*, pages 1243–1248, Chicago, Illinois.

Wang, R. and W. Cohen. 2008. Iterative set expansion of named entities using the web. In *Proceedings of the International Conference on Data Mining (ICDM-08)*, pages 1091–1096, Pisa, Italy.

Wang, R. and W. Cohen. 2009. Automatic set instance extraction using the Web. In *Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics (ACL-IJCNLP-09)*, pages 441–449, Singapore.

Wu, F. and D. Weld. 2008. Automatically refining the Wikipedia infobox ontology. In *Proceedings of the 17th World Wide Web Conference (WWW-08)*, pages 635–644, Beijing, China.

Incremental Chinese Lexicon Extraction with Minimal Resources on a Domain-Specific Corpus

Gaël Patin
(1) Texts, Computer Science and Multilingualism Research Center (Ertim)
National Institute of Oriental Languages and Civilizations (Inalco)
(2) Arisem, Thales Company
gael.patin@arisem.com

Abstract

This article presents an original lexical unit extraction system for Chinese. The method is based on an incremental process driven by an association score featuring a minimal resources statistically aided linguistic approach. We also introduce a linguistics-based lexical unit definition and use it to describe an evaluation protocol dedicated to the task. The experimental results on a domain specific corpus show that the method performs better than other approaches. The extraction results, evaluated on a random sample of the working corpus, show a recall of 68.4 % and precision of 37.1 %.

1 Introduction

Lexical resources are all the more fundamental to NLP systems since domain specific corpora are multiple and various. The performance of common tasks, such as Information Retrieval or Information Extraction, can be improved by comprehensive and updated domain specific lexicon (i.e. terminology). However the constitution of lexicons raises pragmatic issues, such as development cost or re-usability, which have a great importance in an industrial context ; and also theoretical issues, such as the definition of the lexical unit or evaluation protocol, which are crucial for the relevance of the results. In Chinese text processing context, lexicons are particularly important for dictionary-based word segmentation techniques in which out-of-vocabulary words are an important cause of errors (Sproat and Emerson, 2003).

In this paper we consider the lexicon extraction task independent of the word segmentation, this position differs from Zhao and Kit's (2004) point of view. Generally speaking, word segmentation aims at delimiting units in a sequence of characters. The delimited units are usually morphological lexical units (i.e. words) and internal composition of the unit is not considered. The evaluation process checks whether each word occurrence is well delimited. On the opposite, lexicon extraction aims at extracting lexicon entries from a corpus. The extracted units are morphological or syntactic units and the internal components are also considered. The evaluation process checks the extracted candidates list considering the corpus global scope.

Many approaches for Chinese lexicon extraction rely on a supervised word segmenter (Wu and Jiang, 2003; Li et al., 2004) or a morpho-syntactic tagger (Piao et al., 2006) to extract unknown words. These techniques perform well but suffer from a major drawback, they cannot be applied efficiently to corpora that cover different domains than the calibration corpus. Some approaches are nested in an unsupervised word segmentation process and aim at improving its effectiveness. Fung and Wu (1994) try to select segments using mutual information on bigram. Chang and Su (1997) present an iterative unsupervised lexicon extraction system driven by the quality of segmentation obtained with the discovered lexicon. This approach, although efficient, imposes an arbitrarily 4-character length restriction on candidates. Other works, like this approach, focus on the lexicon or terminology extraction as standalone task. Feng et al. (2004) introduce a lexicon extraction unsuper-

Coling 2008: Poster Volume, pages 963–971,
Beijing, August 2010

vised method based on context variation with very convincing results. Yang et al. (2008) focus on terminology extraction using delimiters extracted from a training corpus with good results.

This study proposes an original answer to the Chinese lexicon extraction task using an incremental minimal resources method to extract and rank lexical unit candidates. An annotated reference corpus is required to extract a common-word dictionary and to prepare the data. The method has the advantage of proposing structured candidates, which allow interactive candidate filtering. In addition the candidate maximum length is determined by the number of associations that allow the detection of the longer lexical units. We extend the association measure method introduced by Sun et al. (1998) for word segmentation without lexical resources. This paper starts with a linguistic definition of the lexical unit which drives the method. We also build on it to propose an improvement of the evaluation protocol for the Chinese lexicon extraction task.

2 Lexical Unit Definition

Although defining the Chinese lexical unit is not a trivial task, we think that it is absolutely necessary for the understanding of the kind of linguistic phenomena we are dealing with. Without this knowledge we may miss important features and may not be able to efficiently evaluate the extraction process. We introduce two linguistic concepts to define the lexical units focusing on contemporary written Chinese: the *morpho-syntactic unit* and the *lexical content*. These definitions use concepts introduced by Polguère (2003) applied to the Chinese case by Nguyen (2008).

2.1 Morpho-syntactic Unit

A *graphy* is the Chinese minimal autonomous orthographic unit and it approximatively matches the glyph concept in computer science. The following glyphs are different Chinese graphies: 猫, 貓, 寿, 葡, 萄. A *morph* (noted | m |) is the smallest meaningful unit representable by a sequence of graphies. Morphs are atomic so that they cannot be representable by a smaller sequence of morphs. The following sequences of graphies are different morphs : |longevity 寿|, |grape 葡萄|, |aspirin 阿司匹林|, |buy 买|. Note that the graphy 萄 does not carry any meaning and is not a morph. A *morpheme* (noted ‖M‖[1]) is a set of morphs sharing the same lexical content ignoring grammatical inflection or variants (Table 1). Chinese morphs cannot be inflected, unlike European languages, but some graphies have variants.

Morpheme	***Morph***
‖protect 保‖	|protect 保|
‖aspirin 阿司匹林‖	|aspirin 阿司匹林|
‖cat 猫‖	|cat 猫| |cat 貓|

Table 1: *Morphemes and related morphs*

A *word-form* (noted (w)) is an autonomous and inseparable sequence of morphs. Autonomy means that it can be enunciated individually and can take place in a syntactic paradigm. Inseparability means that breaking the sequence causes the loss of the relationship between elements. A *lexeme* (noted ⟪ w ⟫) is a set of word-forms sharing the same lexical content ignoring inflection or variants (Table 2).

Lexeme	***Word-form***
⟪aspirin 阿司匹林⟫	(|aspirin 阿司匹林|)
⟪take 拿⟫	(|take 拿|) (|take /prefect/ 拿|了|) (|take /progressive/ 拿|着|) (|take /experience/ 拿|过|)
⟪insurance 保险⟫	(|insurance 保|险|)
⟪|panda 熊|猫|⟫	(|panda 熊|猫|) (|panda 熊|貓|)

Table 2: *Lexemes and associated word-forms*

A *phrase* (noted [s]) is a syntactic combination of word-forms. The syntactic nature of the combination implies that the phrase components are relatively free. A *locution* (noted ⟦ s ⟧) is a set of lexicalized phrases sharing the same lexical content ignoring inflection or variants (Table 3).

Locution	***Phrase***
⟦shoot 开枪⟧	[(shoot 开)(枪)] [(shoot /prefect/ 开了)(枪)] ...
⟦be jealous 吃醋⟧	[(be jealous 吃)(醋)]
⟦insurance company 保险公司⟧	[(insurance company 保险)(公司)]

Table 3: *Locutions and associated phrases*

[1]The standard simplified form is used to represent morphemes.

The morphs, word-forms and phrases are the morpho-syntactic units, they describe the composition of lexemes and locutions.

2.2 Lexical Content

The lexical units we look for are lexemes and locutions. Finding lexical units means identifying words-forms and phrases having a lexical content. We use two criteria to define the lexical content: the *compositionality criterion* and the *referentiality criterion* (Table 4). Units which fulfill at least one of these criteria are said to have a lexical content.

The compositionality criterion (or lexicalization criterion) is relative to the relationship between the sense of the unit and the sense of its components. The question is whether or not the sense of the unit can be deduced from the combination of its components. The referentiality criterion is related to the relationship between the unit and the referent concept or object. The question is whether or not the referent has specific properties for the speakers. This criterion is strongly dependent on human judgment and the working domain.

	Referential	***No-Referential***
Compositional	《Chinese food 中餐》	(anticization 古代化)
	⟦insurance company 保险公司⟧	[African car 非洲汽车]
Lexicalized	《disinfect 消毒》	⟦everyone 大家⟧
	⟦dividend product 分红产品⟧	⟦selling vinegar as wine 挂羊头卖狗肉⟧

Table 4: *Referential and Compositional units*

The Table 4 presents examples of four criterion combinations. Referentiality and compositionality criteria are always applied at the highest association level, thus ⟦insurance company 保险公司⟧) is compositional, although 《insurance 保险》 and 《company 公司》 are not compositional. Word-forms are not necessarily compositional or referential, thus the unit (anticization 古代化) does not refer to any concept and we can use the combination of its components to interpret it: 《antiquity 古代》 + ‖-ation 化‖. Referentiality does not imply lexicalization, thus the compositional unit ⟦German car 德国汽车⟧ is referential because it refers to the German car brands or characteristics in the automobile context.

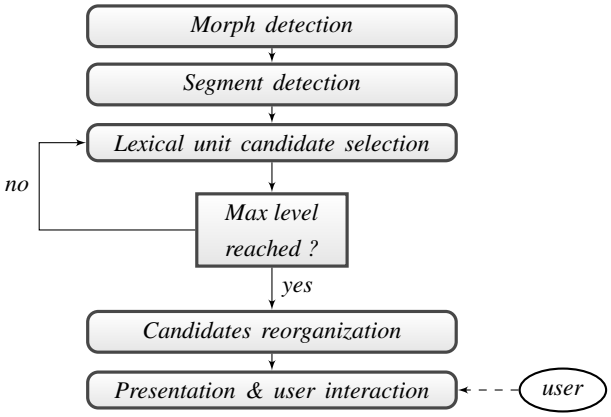


Figure 1: *Method overview*

3 Methodology

The method (Figure 1) follows the linguistic intuitions developed in the previous section. We identify morpho-syntactic units and select those that are likely to have a lexical content to obtain *lexical unit candidates* (LUCs). The word-forms and phrases are respectively generated by associations of morphs or word-forms and association of word-forms or phrases. We consequently use an incremental process, which associates LUCs as they are selected. The incremental process is initiated by detecting every morph and spliting the corpus into segments. Then we enumerate all the morpho-syntactic unit couples and use lexical content criteria to select the couples to associate. This process is repeated until the maximum number of associations is reached. At the end, the LUCs are reorganized and submitted to the user. The user's answers are used to filter the remaining LUCs.

3.1 Morphs Detection

As stated in Section 2.1, we consider that the morph is the minimal morpho-syntactic unit. Every glyph is considered as a morph unless it can be included in an ancient loanword morph (《butterfly 蝴蝶》 《garbage 垃圾》) or a foreign transcription morph (《Italy 意大利》, 《microphone 麦克风》). In an ambiguous case the longest possibility is accepted. Foreign transcriptions are phonetic interpretations of foreign words using the pronunciation of the Chinese graphies. The set of graphies used for transcription is well-

known and closed. We trained a CRF tagger[2] using simple features based on current, next and previous graphies to extract foreign transcriptions (the training corpus is described in Section 4.1). Ancient loanwords importation process is not productive anymore, thus they are detected using a loanword list.

3.2 Segment Detection

The aim of the segment detection step is to split the corpus into segments (i.e. a succession of Chinese graphies). Chinese texts contain two kinds of delimiters which are not likely to be components of a lexical unit, delimiter-words and delimiter-expressions. Delimiter-words are enumerable with a common word dictionary[3] and include prepositions (对, 使), adverbs (很, 也, 都), pronouns (我, 其他, 那儿), interrogative pronouns (哪里, 谁), conjunctions (而且, 但, 因此), discourse structure words (目前, 按照, 由), tonal particles (啊, 吧) and tool-words (的). Delimiter-expressions include numerical expressions (六万美元, 三个), temporal expressions (今天晚上, 八点左右), circumstantial expressions (从...开始, 在...中) , which are easily describable using shallow context-free grammars. Delimiters are removed from the corpus and used to delimit the segments. The inflexions (了, 过, 着), which introduce inflectional variations, are also removed from the corpus but do not delimit the segments. The delimiters identification is controlled by rules. For instance tonal particles are removed only if they are the end of a segment, discourse structure words are removed only if they are the beginning of a segment. Delimiters and inflexions are not removed if they are inside a sequence of graphies which is present in a common-word dictionary.

3.3 Selection of Lexical Unit Candidates

In this step, *lexical unit candidates* (LUCs) are extracted by selecting morpho-syntactic unit couples, which are likely to have a lexical content. The first assumption is that lexical units can always be decomposed into binary trees. Only a small number of lexical units do not satisfy this assumption (e.g. 乌漆墨黑), in such case it is possible to select a non-linguistically motivated way to decompose the unit into binary associations. Thus, every couples of contiguous morpho-syntactic units are iteratively enumerated for each segment. The second assumption is that *association measures* are good statistical evidence to detect lexical content. Thereby, the association strength of morpho-syntactic couples is used as a main criterion to identify relevant candidates.

Sentence with delimiters noted {delimiter}:

公司银代主力产品“新红A”、“新红C”两款分红产品适应{了}今年资本市场{的}现状，产品设计、分红水平、特殊红利分配{等}方面{都}得到合作银行{和}客户{的}认同，充分满足{了}客户{的}预期利益，{在}市场{上}得到{了}{很}高{的}美誉度。

Obtained segments noted [segment]:

[公司银代主力产品] [新红] [新红] [两款分红产品适应今年资本市场] [现状] [产品设计] [分红水平] [特殊红利分配] [方面] [得到合作银行] [客户] [认同] [充分满足客户] [预期利益] [市场] [得到] [高] [美誉度]

Figure 2: *Segment detection example*

Consider G the alphabet of all Chinese graphies, $M = G^+$ the language describing the morpho-syntactic units, S_n a set of segments at step n, $s_n^i = m_1, m_2, ..., m_n$ the i^{th} segment of S_n where $\forall m \in s_n^i \mid m \in M$ and S_n^* the set of all morpho-syntactic unit couples in S_n segments. Given the morpho-syntactic unit couple m_i, $m_{i+1} \in S_n^*$ (denoted as $m_{i,i+1}$), the *lexical content criteria* ($LCC(m_{i,i+1})$) matches if the following conditions are fulfilled:

1. Neither m_i nor m_{i+1} has not been associated at the current step n.
2. $Nb(m_i) \neq 1$ or $Nb(m_{i+1}) \neq 1$.
3. $AS(m_{i,i+1}) > T$.
4. $AS(m_{i,i+1}) > AS(m_{i-1,i})$ or not $LCC(m_{i-1,i})$
5. $AS(m_{i,i+1}) > AS(m_{i+1,i+2})$ or not $LCC(m_{i+1,i+2})$

where $Nb(x)$ is the number of occurrences of x, $AS(x, y)$ returns the association score of the cou-

[2] CRF++ implementation of Conditional Random Fields

[3] We assert that this kind of dictionary is easily available

ple x, y computed with a given association measure, and T is the association threshold relative to the association measure (cf. 4.1).

Let S_0 the initial set of segments where $\forall s_0^i \in S_0$, s_0^i is a segment (cf. 3.2) such that $\forall m \in s_0^i$, m is a morph (cf. 3.1). The LUC list is composed of morpho-syntactic couples produced by the association operator $\oplus$ to compute S_{max} (algorithm 1) with max the maximum number of iteration.

$$
\begin{aligned}
&S \leftarrow S_{n-1} \\
&\textbf{while} \quad \exists\, m_{i,i+1} \in S^* |\ LCC(m_{i,i+1}) \\
&\quad S \leftarrow S[m_i \oplus m_{i+1}] \qquad (1) \\
&\textbf{end} \\
&S_n \leftarrow S
\end{aligned}
$$

with $\oplus$ the association operator whose result is a morpho-syntactic unit, $S_n[m_1 \oplus m_2]$ the replacement of m_1 and m_2 by the morpho-syntactic unit $m_1 \oplus m_2$ in the corresponding segment. See the Section 5 for more details about the maximum number of iteration setting.

3.4 Candidates Reorganization

Once LUCs are extracted, we map every LUC to the couple of morpho-syntactic units it is composed of. These units are called *components*. Some LUCs are generated from two different couples at the candidate selection step. For instance, 旅游业者 is discovered in two ways: 旅游⊕业者 or 旅游业⊕者. We always choose the most frequent option. When the "LUC/couple" map is created, we sort the LUCs by their corresponding couple association scores. Finally, if a LUC is ranked in the list before its components we move the components to the position just before it in the list and use the same rule to recursively check the moved components. The candidates list is expected to be ordered by likelihood deduced by an association measure and compositional order.

3.5 Presentation and User Interaction

The lexicon extraction task aims at submitting a ranked list of candidates to the user in order to help him produce lexical resources. The user is expected to check the list in this order and the method uses the user answers to discard not yet verified candidates. To do so, the user is asked to answer the following questions for each LUC according to the definition given in the Section 2:

1. Does the unit have a lexical content ?
2. Is the unit a part of a lexical unit ?

If answers to both these questions are 'no' then all the candidates having this component are removed from the remaining list.

4 Evaluation

Since the submitted candidates are progressively modified according to the user answers, the evaluated candidates are only the ones submitted to the user. We used three measures to evaluate the method: recall, precision and precision at rank n. Since producing large annotated corpora is costly, we perform the evaluation using a sample of texts from the evaluation corpus. Therefore the scores obtained are an estimation of the true scores. The inter-human variation is not considered here and should be integrated in further works.

4.1 Evaluation parameters

The morphs and the segment detection step use data from a *reference corpus*: *The Lancaster Corpus of Mandarin Chinese* (McEnery and Xiao, 2004). The corpus is composed of text samples choose in various domain and genre corpora, it contains two millions of glyphs and it is annotated according to the Beijing University annotation guideline[4]. This corpus is mainly used to extract delimiter-words, to produce the grammar for delimiter-expressions and to extract a common-word dictionary. All foreign transcriptions are also annotated for the CRF tagger training (cf. 3.1).

The lexical unit detection step is evaluated using four well-known association measures: Pointwise Mutual Information (PMI), Poisson-Striling (PS) (Quasthoff and Wolff, 2002), Log-likelihood (LL), Pointwise Mutual Information Cube (PMI^3) (Daille, 1994). These measures are detailed in table 5. The significant association threshold is intuitively given by the statistical interpretation of

[4] *http://icl.pku.edu.cn/icl_groups/corpus/coprus-annotation.htm*

AM	*Formulas*	*Variables*
PMI	$\log \frac{p_{xy}}{p_{x*}p_{*y}}$	x, y : words $\overline{x}$: all words but x $*$: all words p_x : x probability f_x : x frequency N : nb. of bigram $\lambda = N \cdot p_x \cdot p_y$ $k = f_{xy}$ $\hat{f}_{xy} = \frac{f_{x*}f_{*y}}{N}$
LL	$2 \sum_{i,j}^{\{x,\overline{x}\},\{y,\overline{y}\}} f_{ij} \log \frac{f_{ij}}{\hat{f}_{ij}}$	
PS	$\frac{k(\log k - \log \lambda - 1)}{\log N}$	
PMI^3	$\log \frac{N {f_{xy}}^3}{f_{x*}f_{*y}}$	

Table 5: *Association score calculation*

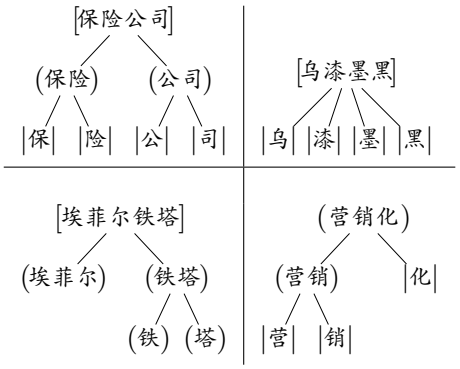


Figure 3: *Lexical unit trees*

the formulas for MI and PS. Thus, these measures are used for LCC's *selection criterion 2* and T is set to 0 (cf. 3.3). A threshold can not be deduced from PS and PMI^3, therefore they are only used for LCC's *comparison criteria 3 & 4*.

4.2 Evaluation Process

To prepare the evaluation we randomly selected twenty texts in an evaluation corpus and annotated lexical units according to the linguistic description given in Section 2. For each sample text, we obtained a set of lexical unit trees (Table 3) corresponding to all the encountered lexical units. N-trees are used for units which can not be transformed into binary tree. Two evaluation sets are defined, the *shallow set* which contains the root nodes of the lexical unit trees and the *deep set* which contains the inner nodes[5] of the lexical unit trees. Given the four examples of Figure 3, the shallow set contains [保险公司], [乌漆墨黑], [埃菲尔铁塔] and (营销化) ; and the deep set contains [保险公司], (保险), (公司), [乌漆墨黑], [埃菲尔铁塔], (铁塔), (营销化) and (营销).

Experiments with different parameters produce different candidate lists and an expert intervention is required to evaluate each candidate list. To avoid this problem, all the repeated sequences of non-inflectional graphies are generated from the annotated sample texts and intersected with the LUC list. The obtained list is a projection of the candidate list on the sample texts. This trick allows us to extract all LUCs appearing in the sample texts and evaluate them automatically.

[5] All nodes excluding leaves.

5 Experiments

The experiments are conducted on insurance domain corpus containing ten million graphies. This evaluation corpus is composed of news and articles collected automatically from Chinese insurance companies websites. The text fields are extracted with an xhtml parser. Several text fields, such as menus or buttons, are repeated and duplicates are removed to avoid noise. The presented method, referred as ILex (Incremental Lexicon Extractor), is applied using the previously mentioned 4 measures (cf. 4.1). The evaluation is based on couple of measures, the first measure is dedicated to candidates selection (LCC 2.) and the second to candidates comparison (LCC 3. & 4.). The comparison measure is also used to sort the candidates (cf. 3.4). The maximal number of iterations is set to 3 (for a maximal depth of 4), which is the maximum number of associations required to compose the majority of lexical units in the reference corpus. The precision and recall are computed on the deep set in order to consider all valid lexical units, the recall on the shallow set is given to see the results on wider lexical units (Table 6). The results show that PMI-LL couple performs better overall than the other measures. It can be noticed that the scores are relatively close ($\pm$1.8% for precision and $\pm$7.0% for deep recall) meaning that the choice of the association measure has a low influence over the results. For the further experiments are conducted with PMI-LL, which achieves the best recall score.

Selection	PMI		PS	
Comparison	LL	PMI^3	LL	PMI^3
Precision	37.1	38.9	37.3	38.1
Deep recall	68.4	65.6	62.3	61.4
Shallow recall	75.1	74.2	70.6	70.6

Table 6: *Measure combinations results*

The method extracted 585,794 LUCs from the whole corpus using the PMI-LL couple before applying the user interaction step. The *candidate list projection* (cf. 4.2) contains 4,539 LUCs. The user decisions are simulated with the lexical unit trees obtained from sample texts. In total 312 LUCs were removed in consequence of the user interaction (cf. 3.5), without this step the precision decreases to 33.7%. The 1,246 LUCs present in the common-word dictionary are ignored. Finally 1,886 invalid candidates and 1,105 valid lexical units are submitted to the user, the evaluation is based on these 3,059 LUCs.

Lexical unit	**Rank**	**Nb.**
⟦政策性农业保险⟧ (policy agricultural insurance)	155	1798
⟦泰康人寿⟧ (Tai Kang Life Insurance)	453	1,854
⟪保险人⟫ (insurer)	1,048	4,999
⟦南开大学⟧ (Nan Kai University)	2,828	111
⟦洛杉矶旅游业者⟧ (Los Angeles tourism professionals)	9,647	3
⟦人寿保险⟧ (life insurance)	11,647	871
⟪王恩韶⟫ (Wang Enshao (person))	14,617	2
⟦有偿使用⟧ (compensated use)	34,596	8
⟪太湖流域⟫ (Taihu Lake Basin)	102,612	2
⟪择机⟫ (wait an opportunity)	126,044	31
⟦中华人民共和国劳动合同法⟧ (The People's Republic of China labor contract law)	387,235	1

Table 7: *Sample of extracted lexical units*

A sample of extracted lexical units is presented in Table 7. In this list, the lower number of occurrences is 1 and the longest unit length is 12 graphies. Most of the extracted lexical unit are terms, a significant number of people names, common words and larger named entities are extracted too. The most part of the very frequent lexical units are ranked at the top of the list but some low frequency LUCs are ranked over the high frequency candidates. The Figure 4 presents

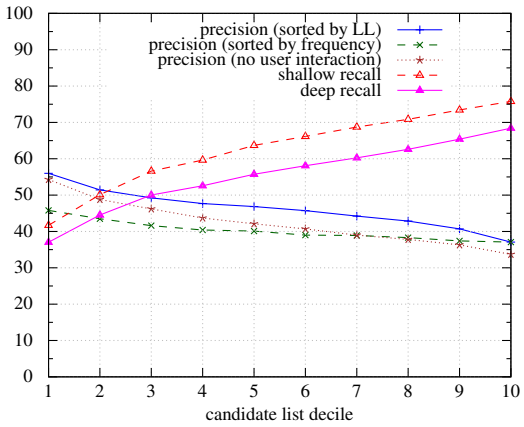


Figure 4: *ILex results using PMI-LL*

the results as a function of the LUC list deciles. The LL sorting is compared to frequency sorting for the precision at rank n. The LL sorting curve is above the frequency sorting curve, this fact shows that LL is more efficient at sorting valid LUCs. The majority of the missed candidates have a low number of occurrences ($\leq$3) and 57.8% of the longest lexical unit ($>$7) are also missed. Most of extraction errors have a low number of occurrences, 40.1% of the errors are caused by lexical unit composition errors (e.g. $*(\text{保险})\oplus|\overset{\text{study}}{\text{学}}|$ in $[(\text{保险})(\overset{\text{insurance institute}}{\text{学院}})]$ or $*(\overset{\text{reform}}{\text{改革}})|\overset{\text{commission}}{\text{委}}|$ in $[(\overset{\text{reform \& development commission}}{\text{发展}})(\text{改革})|\text{委}|]$) and 59.9% by association errors (e.g. $(\overset{\text{extend}}{\text{扩大}})\oplus[\overset{\text{agricultural Insurance}}{\text{农业保险}}]$ or $(\overset{\text{standard}}{\text{规范}})\oplus(\overset{\text{development}}{\text{发展}})$).

The AccessVar method (Feng et al., 2004), an unsupervised lexicon extraction method having the best performance, was reimplemented and used as a reference. This method uses the corpus substrings' number of distinct contexts, noted AV (*accessor variety*), to extract candidates. AccessVar is configured by an *accessors variety threshold* (AVT), which is the minimal AV required to hold a candidate, the number of occurrences of candidates is consequently greater or equal to the AVT. For the experiments, the candidate maximal length is set to 7 graphies[6] and AVT to 3. Similarly, ILex candidates appearing less than three times and having a length greater than 7 are discarded. The ILex user interaction is not applied for this comparison. In order unify the input data, AccessVar handles the segments detected by ILex

[6]Higher values cause space complexity issues.

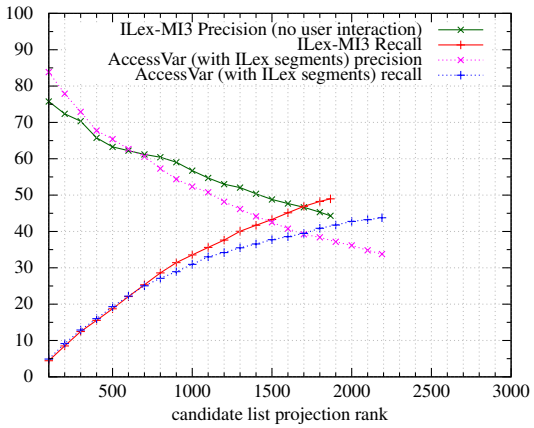


Figure 5: *ILex & AccessVar results*

instead of the corpus full text.

AccessVar and ILex extract respectively 125,467 and 116,412 LUCs and the candidate list projection contains 2,190 and 1,876 LUCs. The results are computed on the deep set (figure 5). AccessVar and ILex achieve respectively recall of 43.7% and 49.0%. A total of 667 of the lexical units extracted are common to both methods, 161 lexical units are extracted exclusively by ILex and 74 lexical units are extracted exclusively by AccessVar. This means that both methods have close covering capacities. From rank 100 to rank 700, the results are close but the curves begin to diverge after this rank, this trend means that the performance are similar for the 700 best candidates. However, ILex achieves 44.4% precision which is 10.6% higher than AccessVar (33.8%), this difference, in view of the close recall score, shows that ILex generates less invalid candidates. The errors specific to AccessVar are due to context adhesion errors (e.g. $*(\overset{\text{company}}{\text{保险}})\oplus|\overset{\text{specialty}}{\text{产}}|$ in $[(\overset{\text{insurance industry}}{\text{保险}})(\text{产业})]$, $[(\overset{\text{insurance product}}{\text{保险}})(\text{产品})]$, $[(\overset{\text{insurance produce}}{\text{保险}})(\text{产生})]$ etc.), or association errors (e.g. $*|\overset{\text{country}}{\text{国}}|\oplus|\overset{\text{East}}{\text{东}}|$, $*(\overset{\text{industry}}{\text{工业}})\oplus(\overset{\text{group}}{\text{集团}})$). ILex avoids these errors because of three mechanisms. First, the statistical likelihood between the couple components is tested (e.g. $*|\overset{\text{country}}{\text{国}}|\oplus|\overset{\text{East}}{\text{东}}|$ PMI score is negative). Second, the method checks association likelihood of the contexts before associating two morpho-syntactic units, (e.g. $(\overset{\text{aeronautic}}{\text{航空}})(\overset{\text{industry}}{\text{工业}})$ score is over $*(\overset{\text{industry}}{\text{工业}})\oplus(\overset{\text{group}}{\text{集团}})$ score in $[\overset{\text{Aviation Industries Corporation of China}}{\text{中国航空工业集团公司}}]$). Third, the incremental association process determine smaller unit before trying associating bigger couples (e.g. $(\overset{\text{insurance}}{\text{保险}})$ and $(\overset{\text{industry}}{\text{产业}})$ are associated before $[\overset{\text{insurance industry}}{\text{保险产业}}]$).

6 Conclusion and Further Works

The presented method features incremental lexical unit extraction with interactive candidate filtering capability. The maximal candidate length is not imposed directly, but instead is determined by the maximal number of associations. The lexical resources required are re-usable and non-domain specific, which significantly reduce their cost for long-term deployment. The method achieves decent performance and improves the reference method's precision for this task. Furthermore, the extracted results include low-frequency and long candidates which are known to be difficult to extract. Finally, the binary association process allows us to sort the candidates by association measure, which is more relevant than frequency.

This paper also introduced the beginning of a linguistically consistent lexical unit definition. This definition draws the outlines of a corpus annotation guide dedicated to the lexicon extraction task. The evaluation process is improved by the lexical unit trees annotations and a candidate list projection technique, which allows full-automatic estimation of extraction system performance.

The first upcoming objective is the development of a robust evaluation protocol for the lexical extraction task. This is crucial for further improvements and means that the variation between annotators of the evaluation corpus, and the stability of the method over different corpora need to be considered. Finally we will try to solve the not yet managed lexicon extraction issues, Latin characters tokens which cause the method miss some extractions (e.g. $(\!(\overset{\{\textit{product name}\}}{\text{新红A}})\!)$), and the discontinuous locutions (e.g. $[\![\overset{\text{reach on the phone}}{\text{打通电话}}]\!]$ in $(\overset{\text{call / through}}{\text{打通了}})\text{常总的}(\overset{\text{phone}}{\text{电话}})$ or $[\![\overset{\text{bear responsibility}}{\text{负责任}}]\!]$ in $\text{本公司}(\overset{\text{bear}}{\text{负}})\text{给付保险金}(\overset{\text{responsibility}}{\text{责任}})$).

Acknowledgements

Our sincere thanks to the anonymous reviewers. Special thanks to Pierre Zweigenbaum, to all my colleagues from Arisem and Ertim and to the corpus annotators without which this work would not be possible.

References

Chang, Jing-Shin and Keh-Yih Su. 1997. An unsupervised iterative method for Chinese new lexicon extraction. *International Journal of Computational Linguistics & Chinese Language Processing*, vol. 1(1), pp. 101–157.

Daille, Béatrice. 1994. *Approche mixte pour l'extraction automatique de terminologie: statistiques lexicales et filtres linguistiques*. PhD thesis, Université Paris 7.

Feng, Haodi, Kang Chen, Xiaotie Deng and Weimin Zheng. 2004. Accessor variety criteria for Chinese word extraction. *Computational Linguistics*, vol. 30:1, pp. 75-93.

Fung, Pascale and Dekai Wu. 1994. Statistical augmentation of a Chinese machine-readable dictionary. In *WVLC-2, Second Annual Workshop on Very Large Corpora (COLING-94)*, Kyoto, Japan, pp. 69-85.

Hai, Zhao and Chunyu Kit. 2008. An Empirical Comparison of Goodness Measures for Unsupervised Chinese Word Segmentation with a Unified Framework. *In Proceedings of the Third International Joint Conference on Natural Language Processing (IJCNLP-08)*, Hyderabad, India, Vol. 1, pp. 9-16.

McEnery, Tony and Richard Xiao. 2004. The Lancaster Corpus of Mandarin Chinese: A corpus for monolingual and contrastive language study. *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC-04)*, Lisbon, Portugal, pp. 1175-1178.

Li, Hongqiao, Changning Huang, Jiangfen Gao and Xiaozhong Fan. 2004. The use of SVM for Chinese new word identification. *First International Joint Conference on Natural Language Processing (IJCNLP-04)*, Sanya, China, pp. 497-504.

Nguyen, Etienne Van Tien. 2008. *Unité lexicale et morphologie en chinois mandarin – vers l'élaboration d'un DEC du chinois*. Phd thesis, Montréal University.

Piao, Scott S. L., Guangfan Sun, Paul Rayson and Qi Yuan. 2006. Automatic extraction of Chinese multiword expressions with a statistical tool. *Workshop on Multi-word-expressions in a Multilingual Context held in conjunction with the 11th EACL*, Trento, Italy, pp. 17-24.

Polguère, Alain. 2003. *Lexicologie et sémantique lexicale. Notions fondamentales*. Presses de l'Université de Montréal, Coll. Paramètres.

Quasthoff, Uwe and Christian Wolff. 2003. *The Poisson collocation measure and its application*. In *Second International Workshop on Computational Approaches to Collocations*, Vienna, Austria.

Sproat, Richard and Tom Emerson. 2003. The first international Chinese word segmentation bakeoff. *Proceedings of the Second SIGHAN Workshop on Chinese Language Processing*, Japan, vol. 17, pp. 133-143.

Sun, Maosong, Danyang Shen and Benjamin K Tsou. 1998. Chinese Word segmentation without lexicon and hand-crafted training data. *Proceedings of the 17th international conference on Computational linguistics*, Montreal, Canada, Vol. 2, pp. 1265-1271.

Wu, Andi and Zixin Jiang. 2000. Statistically-enhanced new word identification in a rule-based Chinese system. *Proceedings of the 2nd Chinese Language Processing Workshop*, Hong-Kong, vol. 12, pp. 45-51.

Yang, Yuhang, Qin Lu and Tiejun Zhao. 2008. Chinese Term Extraction Using Minimal Resources. *Proceedings of the 22nd International Conference on Computational Linguistics*, Manchester, United Kingdom, Vol. 1, pp.1033-1040.

Improving Name Origin Recognition with Context Features and Unlabelled Data

Vladimir Pervouchine, Min Zhang, Ming Liu and Haizhou Li
Institute for Infocomm Research, A-STAR
vpervouchine@gmail.com,{mzhang,mliu,hli}@i2r.a-star.edu.sg

Abstract

We demonstrate the use of context features, namely, names of places, and unlabelled data for the detection of personal name language of origin.

While some early work used either rule-based methods or n-gram statistical models to determine the name language of origin, we use the discriminative classification maximum entropy model and view the task as a classification task. We perform bootstrapping of the learning using list of names out of context but with known origin and then using expectation-maximisation algorithm to further train the model on a large corpus of names of unknown origin but with context features. Using a relatively small unlabelled corpus we improve the accuracy of name origin recognition for names written in Chinese from 82.7% to 85.8%, a significant reduction in the error rate. The improvement in F-score for infrequent Japanese names is even greater: from 77.4% without context features to 82.8% with context features.

1 Introduction

Transliteration is a process of rewriting a word from a source language to a target language in a different writing system using the word's phonological equivalent. Many technical terms and proper nouns, such as personal names, names of places and organisations are transliterated during translation of a text from one language to another. A process reverse to the transliteration, which is recovering a word in its native language from its transliteration in a foreign language, is called back-transliteration (Knight and Graehl, 1998). In many natural language processing (NLP) tasks such as machine translation and cross-lingual information retrieval, transliteration is an important component.

Name origin refers to the language of origin of a name. For example, the origin of English name "Smith" and its Chinese transliteration "史密斯 (Shi-Mi-Si)" is English, while both "Tokyo" and "东京 (Dong-Jing)" are of Japanese origin.

For machine transliteration the name origins dictate the way we re-write a foreign name. For example, given a name written in Chinese for which we do not have a translation in an English-Chinese dictionary, we first have to decide whether the name is of Chinese, Japanese, Korean, English or another origin. Then we follow the transliteration rules implied by the origin of the name. Although all English personal names are rendered in 26 letters, they may come from different romanization systems. Each romanisation sys-

Coling 2008: Poster Volume, pages 972–978,
Beijing, August 2010

tem has its own rewriting rules. English name "Smith" could be directly transliterated into Chinese as "史密斯 (Shi-Mi-Si)" since it follows the English phonetic rules, while the Chinese translation of Japanese name "Koizumi" becomes "小泉 (Xiao-Quan)" following the Japanese phonetic rules. The name origins are equally important in back-transliteration. Li et al. (2007b) demonstrated that incorporating name origin recognition (NOR) into a transliteration system greatly improves the performance of personal name transliteration. Besides multilingual processing, the name origin also provides useful semantic information (regional and language information) for common NLP tasks, such as co-reference resolution and name entity recognition.

Unfortunately, not much attention has been given to name origin recognition (NOR) so far in the literature. In this paper, we are interested in recognition of the origins of names written in Chinese, which names can be of three origins: Chinese, Japanese or English, where "English" is a rather broad category that includes other West European and American names written natively in Latin script.

Unlike previous work (Qu and Grefenstette, 2004; Li et al., 2007a; Li et al., 2007b), where NOR was formulated with a generative model, we follow the approach of Zhang et al. (2008) and regard the NOR task as a classification problem, using a discriminative learning algorithm for classification. Furthermore, in the training data with names labelled with their origin is rather limited, whereas there is vast data from news articles that contains many personal names without any labels of their origins. In this research we propose a method to harness the power of the unlabelled noisy news data by bootstrapping the learning process with labelled data and then using the *personal name context* in the unlabelled data to improve the NOR model. We achieve that by using the maximum entropy model and the expectation-maximisation training, and demonstrate that our method can significantly improve the accuracy of NOR compared to the baseline model trained only from the labelled data.

The rest of the paper is organised as follows: in Section 2 we review the previous research. In Section 3 we present our approach, and in Section 4 we describe our experimental setup, the data used and the evaluation method. We conclude in Section 5.

2 Related research

Most the research up to date focuses primarily on recognition of origin of names written in Latin script, called English NOR (ENOR), although the same methods can be extended to names in Chinese script (CNOR). We notice that there are two informative clues that used in previous work in ENOR. One is the lexical structure of a romanisation system, for example, Hanyu Pinyin, Mandarin Wade-Giles, Japanese Hepbrun or Korean Yale, each has a finite set of syllable inventory (Li et al., 2007a). Another is the phonetic and phonotactic structure of a language, such as phonetic composition, syllable structure. For example, English has unique consonant clusters such as *"str"* and *"ks"* which Chinese, Japanese and Korean (CJK) do not have. Considering the NOR solutions by the use of these two clues, we can roughly group them into two categories: rule-based methods (for solutions based on lexical structures) and statistical methods (for solutions based on phonotactic structures).

Rule-based method Kuo et al. (2007) proposed using a rule-based method to recognise different romanisation system for Chinese only. The left-to-right longest match-based lexical segmentation was used to parse a test word. The romanisation system is confirmed

if it gives rise to a successful parse of the test word. This kind of approach (Qu and Grefenstette, 2004) is suitable for romanisation systems that have a finite set of discriminative syllable inventory, such as Pinyin for Chinese Mandarin. For the general tasks of identifying the language origin and romanisation system, rule based approach sounds less attractive because not all languages have a finite set of discriminative syllable inventory.

N-gram statistics methods

N-gram sum method Qu and Grefenstette (2004) proposed a NOR identifier using a trigram language model (Cavnar and Trenkle, 1994) to distinguish personal names of three language origins, namely Chinese, Japanese and English. In their work the training set includes 11,416 Chinese, 83,295 Japanese and 88,000 English name entries. However, the trigram is defined as the joint probability $p(c_i c_{i-1} c_{i-2})$ rather than the commonly used conditional probability $p(c_i|c_{i-1}c_{i-2})$. Therefore it is basically a substring unigram probability. For origin recognition of Japanese names, this method works well with an accuracy of 92%. However, for English and Chinese, the results are far behind with a reported accuracy of 87% and 70% respectively.

N-gram perplexity method Li et al. (2007a) proposed a method of NOR using n-gram character perplexity PP_c to identify the origin of names written in Latin script. Using bigrams, the perplexity is defined as

$$PP_c = 2^{\frac{-1}{N_c}\sum_{i=1}^{N_c} \log p(c_i|c_{i-1})}$$

where N_c is the total number of characters in a given name, c_i is the i-th character in the name and $p(c_i|c_{i-1})$ is the bigram probability learned from a list of names of the same origin. Therefore, PP_c can be used to measure how well a new name fits the model learned from the training set of names. The origin is assigned according to the model that gives the lowest perplexity value. Li et al. (2007a) demonstrated that using PP_c gives much better performance than with the substring unigram method.

Classification method Zhang et al. (2008) proposed using a discriminative classification approach and extract features from the names. They use Maximum Entropy (MaxEnt) model and a number of features based on n-grams, character positions, word length as well as some rule-based phonetic features. They performed both ENOR and CNOR and demonstrated that their method indeed leads to better performance in name origin recognition then the n-gram statistics method. They attribute that to the fact their model incorporates more robust features than the n-gram statistics based models.

In this paper we too follow the discriminating classification approach, but we add features based on the context of a personal name. These features require the original text with the names to be available. Our approach closely models the real-life situation when large corpora of articles with personal names is readily available in the Web, yet the origins of the names are unknown.

3 Model and training methods

3.1 Maximum entropy model for NOR

The principle of maximum entropy is that given a collection of facts we should choose a model that is consistent with all the facts but otherwise as uniform as possible (Berger et al., 1996). maximum entropy model (MaxEnt) is known to easily combine diverse features and

has been used widely in natural language processing research. Given an observation x the probability of outcome label c_i, $i = 1 \ldots N$ given x is given by

$$p(c_i|x) = \frac{1}{Z} \exp\left(\sum_{j=1}^{n} \lambda_j f_j\,(x, c_i)\right) \quad (1)$$

where N is the number of the outcome labels, which is the number of name origins in our case, n is the number of features, f_j are the feature functions and λ_j are the model parameters. Each parameter corresponds to exactly one feature and can be viewed as a "weight" for the corresponding feature. Z is the normalisation factor given by

$$Z = \sum_{i=1}^{N} p(c_i|x) \quad (2)$$

In the problem at hand x is a personal name and all the features are binary. The features, also known as contextual predicates, are in the form

$$f_i(x, c) = \begin{cases} 1 & \text{if } c = c_i \text{ and } cp(x) = true \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

where cp is the contextual predicate that maps a pair (c_i, x) to $\{true, false\}$.

In our experiments we use Zhang's maximum entropy library[1].

3.2 Initial training with labelled data and n-gram features

For the initial training of MaxEnt model we use labelled data: personal names of Chinese, Japanese or English origin written in Chinese. The origin of each name is known. Following paper by Zhang et al. (2008) and their findings regarding the contribution value of each feature that they studied, we extract unigram, positional unigram and word length features. For example, Chinese name "温家宝" has the following features:

温 家 宝 (温,0) (家,1) (宝,2) 3

We restrict the n-gram features to unigram only to avoid the data sparseness, because our data contains a number of Chinese surnames and given names, which have a length of one or two characters.

3.3 Further training with unlabelled data and context features

For further training of MaxEnt model we use unlabelled data collected from news articles. The name origin is not known but each personal name is in a context and is often surrounded by names of places that may give a hint about the personal name origin. For each personal name we extract all names of places in the same paragraph and use them as features. If a place name is repeated many times in the same paragraph we only include it once in the feature list.

For example, paragraph containing passage *"The U.S. President Barack Obama ..."* will result in two personal names "Barack" and "Obama" having "U.S." as their context feature. Due to the diversity of place names we also attempt to map the names of the places into the country names. In this case, features like "U.S.", "USA", "America" are manually substituted with "USA". In our experiments we also try to narrow the place name extraction to windows of different sizes surrounding the personal name. The rationale here is that the closer a place name is to the personal name, the more likely it has a connection to the origin of the personal name.

In summary, our algorithm includes two steps.

[1] http://homepages.inf.ed.ac.uk/lzhang10/maxent_toolkit.html

First, we use the boostrap data and n-gram, positional n-gram and name length features to do the initial training (the 0-th iteration) of MaxEnt model with L-BFGS method (Byrd et al., 1995). After that we use the model to assign origin labels to names of the training set of the unlabelled data.

Next, we use both the bootstrap data and the training set of the unlabelled data, labelled in the previous step, and add the context features to the already used n-gram, positional n-gram and name length features. Since there is no context available for the bootstrap data, the context features for it are missing, which can be handled by the MaxEnt model. We perform the Expectation-Maximisation (EM) iterations by using the mixed data to train the i-th iteration of the MaxEnt model, then use the model to re-label the training set of the unlabelled data and repeat the training of the model for the $(i + 1)$-st iteration. We stop the iterations when the ratio of patterns that change the origin labels becomes less than 0.01%.

4 Experiments

4.1 Corpora

The corpora consists of two datasets. One dataset, called the "bootstrap data", is a set of Chinese, Japanese and English names written in Chinese following the respective transliteration rules according to the name origins. The names are a mixture of full names, first (given) names and surnames. Table 1 shows the number of names of each origin. This is the labelled data; the origin of each name is known. The data is used to start the MaxEnt model training.

The second dataset, called the "unlabelled data", is Chinese, Japanese and English personal names written in Chinese, which have been extracted from the news articles collected over 6 months from Xinhua news website. The articles have been processed by an automatic part-of-speech (POS) tagger, after which personal names and names of places have been manually identified (the latter for extracting the context features). Normally the first (given) name and surnames are identified as two separate personal names. The data is split into a training set of 27,882 names with unknown origin and a testing set of 1,476 names whose origin was manually assigned. We split data in such a way that there is no overlap between patterns in the training and testing sets, although there may be overlap between names. For example, if a name may be present in both training and testing sets but in a different context, making the two names two distinct patterns. The number of names of each origin in the testing set is shown in Table 2. As seen from the table, the number of Chinese names exceeds the number of English or Japanese names. This is an expected consequence of using articles from a Chinese news agency because many of the articles are reporting on local affairs. We have manually removed a number of Chinese name patterns from the testing set, since the original percentage of Chinese names in the articles is about 83%.

Origin	Number of names
Chinese	52,342
Japanese	26,171
English	26,171

Table 1: Number of names of each origin in the bootstrap dataset.

Origin	Number of names
Chinese	738
Japanese	369
English	422

Table 2: Number of names of each origin in the testing dataset.

4.2 Evaluation method

Following Zhang et al. (2008) to make our results comparable to theirs, we evaluate our system using precision P_o, recall R_o and F-score F_o for each origin $o \in \{\text{“}Chinese\text{”} \text{“}Japanese\text{”} \text{“}English\text{”}\}$. Let the number of correctly recognised names of a given origin o be k_o, and the total number of names recognised as being of origin o be m_o, while the actual number of names of origin o be n_o. Then the precision, recall and F-score are given as:

$$P_o = \frac{k_o}{m_o}$$

$$R_o = \frac{k_o}{n_o}$$

$$F_o = \frac{2 \times P_o \times R_o}{P_o + R_o}$$

We also report the overall accuracy of the system (or, rather the overall recall), which is the ratio of the total number of correctly recognised names to the number of all names:

$$Acc = \frac{k_{Chinese} + k_{Japanese} + k_{English}}{n_{Chinese} + n_{Japanese} + n_{English}}$$

4.3 Results

After each iteration of our MaxEnt-based EM algorithm, we record the number of patterns in the training set that changed their origin labels, as well as calculate the precision, recall and F-score for each origin as well as the overall accuracy. The results are reported in Tables 3 and 4, where for the sake of brevity the origin subscripts are “C”, “J” and “W” for Chinese, Japanese and English name origin respectively.

Compared to the 0-th iteration there is an significant improvement in accuracy, particularly in recognition of Japanese names, which are relatively infrequent compared to Chinese and English ones in the unlabelled training data. This clearly shows the effectiveness of our proposed method.

Iteration	P_C	P_J	P_W	R_C	R_J	R_W
0	0.887	0.724	0.857	0.823	0.911	0.761
1	0.914	0.736	0.875	0.823	0.968	0.775
2	0.910	0.736	0.874	0.823	0.968	0.767
3	0.914	0.737	0.874	0.824	0.973	0.767
4	0.913	0.742	0.875	0.825	0.968	0.778

Table 3: Results of running EM iterations, original names of the places are kept.

Iteration	Acc	F_C	F_J	F_W
0	0.829	0.854	0.807	0.806
1	0.847	0.866	0.836	0.822
2	0.845	0.864	0.836	0.817
3	0.847	0.867	0.839	0.817
4	0.849	0.867	0.840	0.824

Table 4: Results of running EM iterations, original names of the places are kept.

5 Conclusions

We propose extension of MaxEnt model for NOR task by using two types of data for training: origin-labelled names alone and origin-unlabelled names in their context surrounding. We show how to apply a simple EM method to make use of the contextual words as features, and improve the NOR accuracy from 82.9% to 84.9% overall, while for rare names such as Japanese the effect of using unlabelled data with context features is even greater.

The purpose of this research is to demonstrate how the unlabelled data can be used. In the future we hope to investigate the use of other context features, as well as to study the effect of data size on the NOR accuracy improvement.

The feature of names' places normally exhibit great variation: one country name may be spelled in many different ways, and often there are names of cities etc that surround personal names. We will explore to normalise names of places by substituting each name with name of the country where the place is in the future work.

References

[Berger et al.1996] Berger, A., Stephen A. Della Pietra, and V. Della Pietra. 1996. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71.

[Byrd et al.1995] Byrd, R. H., P. Lu, and J. Nocedal. 1995. A limited memory algorithm for bound constrained optimization. *SIAM Journal of Scientific and Statistical Computing*, 16(5):1190–1208.

[Cavnar and Trenkle1994] Cavnar, William B. and John M. Trenkle. 1994. Ngram based text categorization. In *Proc. 3rd Annual Symposium on Document Analysis and Information Retrieval*, pages 275–282.

[Knight and Graehl1998] Knight, Kevin and Jonathan Graehl. 1998. Machine transliteration. *Computational Linguistics*, 24(4).

[Kuo et al.2007] Kuo, Jin-Shea, Haizhou Li, and Ying-Kuei Yang. 2007. A phonetic similarity model for automatic extraction of transliteration pairs. *ACM Transactions on Asian Language Information Processing*, 6(2).

[Li et al.2007a] Li, Haizhou, Shuanhu Bai, and Jin-Shea Kuo. 2007a. Transliteration. In *Advances in Chinese Spoken Language Processing*, chapter 15, pages 341–364. World Scientific.

[Li et al.2007b] Li, Haizhou, Khe Chai Sim, Jin-Shea Kuo, and Minghui Dong. 2007b. Semantic transliteration of personal names. In *Proc. 45th Annual Meeting of the ACL*, pages 120–127.

[Qu and Grefenstette2004] Qu, Yan and Gregory Grefenstette. 2004. Finding ideographic representations of Japanese names written in Latin script via language identification and corpus validation. In *Proc. 42nd ACL Annual Meeting*, pages 183–190, Barcelona, Spain.

[Zhang et al.2008] Zhang, Min, Chengjie Sun, Haizhou Li, Aiti Aw, Chew Lim Tan, and Xiaolong Wang. 2008. Name origin recognition using maximum entropy model and diverse features. In *Proc. 3rd Int'l Conf. NLP*, pages 56–63.

Filling Knowledge Gaps in Text for Machine Reading

Anselmo Peñas
UNED NLP & IR Group
anselmo@lsi.uned.es

Eduard Hovy
USC Information Sciences Institute
hovy@isi.edu

Abstract

Texts are replete with gaps, information omitted since authors assume a certain amount of background knowledge. We define the process of enrichment that fills these gaps. We describe how enrichment can be performed using a Background Knowledge Base built from a large corpus. We evaluate the effectiveness of various openly available background knowledge bases and we identify the kind of information necessary for enrichment.

1 Introduction: Knowledge Gaps

Automated understanding of connected text remains an unsolved challenge in NLP. In contrast to systems that harvest information from large collections of text, or that extract only certain pre-specified kinds of information from single texts, the task of extracting and integrating all information from a single text, and building a coherent and relatively complete representation of its content, is still beyond current capabilities.

A significant obstacle is the fact that text always omits information that is important, but that people recover effortlessly. Authors leave out information that they assume is known to their readers, since its inclusion (under the Gricean maxim of minimality) would carry an additional, often pragmatic, import. The problem is that systems cannot perform the recovery since they lack the requisite background knowledge and inferential machinery to use it.

In this research we address the problem of automatically recovering such omitted information to 'plug the gaps' in text. To do so, we describe the background knowledge required as well as a procedure of *enrichment*, which recognizes where gaps exist and fills them out using appropriate background knowledge as needed. We define *enrichment* as:

Def: Enrichment is the process of adding explicitly to a text's representation the information that is either implicit or missing in the text.

Central to enrichment is the source of the new knowledge. The use of Proposition Stores as Background Knowledge Bases (BKB) have been argued to be useful for improving parsing, coreference resolution, and word sense disambiguation (Clark and Harrison 2009). We argue here that Proposition Stores are also useful for Enrichment and show how in Section 4. However, we show in Section 5 that current BKB resources such as TextRunner (Banko et al. 2007) and DART (Clark and Harrison 2009) are not ideal for enrichment purposes. In some cases there is a lack in normalization. But the most important shortcoming is the lack in answering about instances, their possible classes, how they relate to propositions, and how different propositions are related through them. We propose easy to achieve extensions in this direction. We test this hypothesis building our own Proposition Store with the proposed extensions, and compare it with them for enrichment in the US football domain.

To perform enrichment, we begin with an initial simple text representation and a Proposition Stores as a background knowledge base. We execute a simple formalized procedure to select and attach appropriate elements from the BKB to the entities and implicit relations present in the initial text representation. Surprisingly, we find that some quite simple processing can be effective if we are able to contextualize the text under interpretation.

We describe in Section 2 our textual representations and in Section 3 the process of building the Proposition Store. Enrichment is described in Section 4, and an evaluation and comparison is performed in Section 5.

2 Text Representation

The initial, shallow, text representation must capture the first impression of what is going on in the text, possibly (unfortunately) losing some fragments. After the first impression, in accord with the purpose of the reading, we "contextual-

Coling 2008: Poster Volume, pages 979–987,
Beijing, August 2010

ize" each sentence, expanding its initial representation with the relevant related background knowledge in our base.

During this process of making explicit the implicit semantic relations it will become apparent whether we need to recover some of the missed elements, whether we need to expand some others, etc. So the process is identified with the growing of the context until deeper interpretation is possible. This approach resembles some well-known theories such as the Theory of Relevance (Sperber and Wilson, 1995). The particular method we envisage is related to Interpretation as Abduction (Hobbs et al. 1993).

How can the initial information be represented so as to enable the context to grow into an interpretation? We hypothesize that:

1. Behind certain syntactic dependencies there are semantic relations.
2. In the case of dependencies between nouns, this semantic relation can be made more explicit using verbs and/or prepositions. The knowledge base (our Proposition Store) must help us find them.

We look for a semantic representation close enough to the syntactic representation we can obtain from the dependency graph. The main syntactic dependencies we want to represent in order to enable enrichment are:

1. Dependencies between nouns such as noun-noun compounds (nn) or possessive (poss).
2. Dependencies between nouns and verbs, such as subject and object relations.
3. Prepositions having two nouns as arguments. Then the preposition becomes the label for the relation, being the object of the preposition the target of the relation.

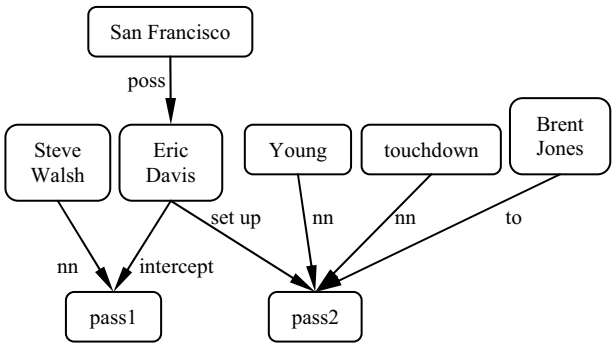


Figure 1. Initial text representation.

We collapse the syntactic dependencies between verb, subject, and object into a single semantic relation. Since we are assuming that the verb is the more explicit expression of a semantic relation, we fix this in the initial representation. The subject will be the source of the relation and the object will be the target of the relation. When the verb has more arguments we consider its expansion as a new node as referred in Section 4.4.

Figure 1 shows the initial minimal representation for the sentence we will use for our discussion: "*San Francisco's Eric Davis intercepted a Steve Walsh pass on the next series to set up a seven-yard Young touchdown pass to Brent Jones"*. Notice that some pieces of the text are missing in the initial representation of the text, as for example "*on the next series"* or *"seven-yard".*

3 Background Knowledge Base

We will use a Proposition Stores as a Background Knowledge Base (BKB). We built it from a collection of 30,826 New York Times articles about US football, similar to the kind of texts we want to interpret. We parsed the collection using a standard dependency parser (Marneffe and Manning, 2008; Klein and Maning, 2003) and, after collapsing some syntactic dependencies, obtained 3,022,305 raw elements in the BKB.

3.1 Types of elements in the BKB

We distinguish three kinds of elements in our Background Knowledge Base: Entities, Propositions, and Lexical relations. All three have associated their frequency in the reference collection.

Entities: We distinguish between entity classes and entity instances:

1. Entity classes: Entity classes are denoted by nouns. We don't restrict classes to any particular predefined set. In addition, we introduce two special classes: Person and Group. These two classes are related to the use of pronouns in text. Pronouns "I", "he" and "she" are linked to class Person. Pronouns "we" and "they" are linked to class Group. For example, the occurrence of the pronoun "he" in "He threw a pass" would produce an additional count of the proposition "person:throw:pass".

2. Entity Instances: Entity instances are indicated by proper nouns. Proper nouns are identified by the part of speech tagging. Some of these instances will participate in the "has-instance" relation (see below). When they participate in a proposition they produce proposition instances.

Propositions: Following Clark and Harrison (2009) we call *propositions* the tuples of words that have some determined pattern of syntactic relations among them. We focus on NVN, NVNPN and NPN proposition types. For example, a NVNPN proposition is a full instantiation of: *Subject:Verb:Object:Prep:Complement*.

The first three elements are the subject, the verb and the direct object. Fourth is the preposition that attaches the PP complement to the verb. For simplicity, indirect objects are considered as a Complement with the preposition "to".

The following are the most frequent NVN propositions in the BKB ordered by frequency.

NVN 2322 'NNP':'beat':'NNP'
NVN 2231 'NNP':'catch':'pass'
NVN 2093 'NNP':'throw':'pass'
NVN 1799 'NNP':'score':'touchdown'
NVN 1792 'NNP':'lead':'NNP'
NVN 1571 'NNP':'play':'NNP'
NVN 1534 'NNP':'win':'game'
NVN 1355 'NNP':'coach':'NNP'
NVN 1330 'NNP':'replace':'NNP'
NVN 1322 'NNP':'kick':'goal'

The 'NNP' tag replaces specific proper nouns (instances) found in the proposition.

When a sentence has more than one complement, a new occurrence is counted for each complement. For example, given the sentence *"Steve_Walsh threw a pass to Brent_Jones in the first quarter",* we would add a count to each of the following propositions:

Steve_Walsh:throw:pass
Steve_Walsh:throw:pass:to:Brent_Jones
Steve_Walsh:throw:pass:in:quarter

Notice that we include only the heads of the noun phrases in the propositions.

We call *proposition classes* the propositions that only involve instance classes (e.g., *"person:throw:pass"*), and *proposition instances* those that involve at least one entity instance (e.g., *"Steve_Walsh:throw:pass*").

Proposition instances are useful for the tracking of a entity instance. For example, *"'Steve_Walsh':'supplant':'John_Fourcade':'as':'quarterback'"*. When a proposition instance is found, it is stored also as a proposition class replacing the proper nouns by a special word (NNP) to indicate the presence of an entity instance. The enrichment of the text is based on the use of most frequent proposition classes.

Lexical Relations: We make use of very general patterns considering appositions and copula verbs (detected by the Stanford parser) in order to extract "is", and "has-instance" relations:

1. **Is**: between two entity classes. They denote a kind of identity between both entity classes, but not in any specific hierarchical relation such as hyponymy. Neither is a relation of synonymy. As a result, it is somehow a kind of underspecified relation that groups those more specific. For example, if we ask the BKB what a "receiver" is, the most frequent relations are:

   290 'person':is:'receiver'
   29 'player':is:'receiver'
   16 'pick':is:'receiver'
   15 'one':is:'receiver'
   14 'receiver':is:'target'
   8 'end':is:'receiver'
   7 'back':is:'receiver'
   6 'position':is:'receiver'

   The number indicates the frequency of the relation in the collection.
2. **Has-instance**: between an entity class and an entity instance. For example, if we ask for instances of team, the top instances with more support in the collection are:

   192 'team':has-instance:'Jets'
   189 'team':has-instance:'Giants'
   43 'team':has-instance:'Eagles'
   40 'team':has-instance:'Bills'
   36 'team':has-instance:'Colts'
   35 'team':has-instance:'Miami'

But we can ask also for the possible classes of an instance. For example, all the entity classes for "*Eric_Davis*" are:

12 'cornerback':has-instance:'Eric_Davis'
1 'hand':has-instance:'Eric_Davis'
1 'back':has-instance:'Eric_Davis'

We still work on other lexical relations such as "part-of" and "is-value-of". For example, the most frequent "is-value-of" relations are:

5178 '[0-9]-[0-9]':is-value-of:'lead'
3996 '[0-9]-[0-9]':is-value-of:'record'
2824 '[0-9]-[0-9]':is-value-of:'loss'
1225 '[0-9]-[0-9]':is-value-of:'season'

4 Enrichment operations

The goal of the following enrichment operations is to make explicit what kind of semantic relations and entity classes are involved in the text.

4.1 Fusion of nodes

Sometimes, the syntactic dependency ties two or more words that form a single concept. This is the case with multiword terms such as "*tight end*", "*field goal*", "*running back*", etc. In these cases, the meaning of the compound is beyond the syntactic dependency. Thus, we shouldn't look for its explicit meaning. Instead, we fuse the nodes into a single one.

The question is whether the fusion of the words into a single expression allows or not the consideration of possible paraphrases. For example, in the case of "*field:nn:goal*", we don't find other ways to express the concept in the BKB. However, in the case of "*touchdown:nn:pass*" we can find, for example, "*pass:for:touchdown*" a significant amount of times, and we want to identify them as equivalent expressions.

4.2 Building context for instances

Suppose we wish to determine what kind of entity "*Steve Walsh*" is in the context of the syntactic dependency "*Steve_Walsh:nn:pass*". First, we look into the BKB for the possible entity classes of *Steve_Walsh* previously found in the collection. In this particular case, the most frequent class is "*quarterback*":

40 'quarterback':has-instance:'Steve_Walsh'
2 'junior':has-instance:'Steve_Walsh'

But what happens if we see "*Steve_Walsh*" for the first time? Then we need to take into account the classes shared by other instances in the same syntactic context. The most frequent are "*Marino*", "*Kelly*", "*Elway*", etc. From them we are able to infer the most plausible class for the new entity. In our example, *quarterback*:

20 'quarterback':has-instance:'Marino'
6 'passer':has-instance:'Marino'
...
17 'quarterback':has-instance:'Kelly'
6 'passer':has-instance:'Kelly'
...
16 'quarterback':has-instance:'Elway'
9 'player':has-instance:'Elway'

4.3 Building context for dependencies

Now we want to determine the meaning behind such syntactic dependencies as:

"*Steve_Walsh:nn:pass*", "*touchdown:nn:pass*", "*Young:nn:pass*" or "*pass:to:Brent_Jones*".

We have two ways for adding more meaning to these syntactic dependencies: find the most appropriate prepositions to describe them, and find the most appropriate verbs. Whether one, the other, or both is useful has to be determined during the reasoning system development.

Finding the prepositions

Several types of propositions in the BKB involve prepositions. The most relevant are NPN and NVNPN. In the case of "*touchdown:nn:pass*", "*for*" is clearly the best interpretation:

NPN 712 'pass':'for':'touchdown'
NPN 24 'pass':'include':'touchdown'

In the case of "*Steve_Walsh:nn:pass*" and "*Young:nn:pass*", since we know they are quarterbacks, we can ask for all the prepositions between "*pass*" and "*quarterback*":

NPN 23 'pass':'from':'quarterback'
NPN 14 'pass':'by':'quarterback'

If we don't have any evidence on the instance class, and we know only that they are instances, the pertinent query to the BKB obtains:

NPN 1305 'pass':'to':'NNP'
NPN 1085 'pass':'from':'NNP'
NPN 147 'pass':'by':'NNP'

In the case of "*Young:nn:pass*" (in "*Young pass to Brent Jones*"), there exists already the preposition "*to*" ("*pass:to:Brent_Jones*"), so the most promising choice becomes the second, "*pass:from:Young*", which has one order of magnitude more occurrences than its successor.

In the case of "*Steve_Walsh:nn:pass*" (in "*Eric Davis intercepted a Steve Walsh pass*") we can use additional information: we know that "*Eric_Davis:intercept:pass"*. So, we can try to find the appropriate preposition using NVNPN propositions in the following way:

"Eric_Davis:intercept:pass:P:Steve_Walsh"

Asking the BKB about the propositions that involve two instances with "*intercept*" and "*pass*", we obtain:

NVNPN 48 'NNP':'intercept':'pass':'by':'NNP'

NVNPN 26 'NNP':'intercept':'pass':'at':'NNP'
NVNPN 12 'NNP':'intercept':'pass':'from':'NNP'

We could also query the BKB with the classes we have already found for "*Eric_Davis*" (*cornerback, player, person*):

NVNPN 11 'person':'intercept':'pass':'by':'NNP'
NVNPN 4 'person':'intercept':'pass':'at':'NNP'
NVNPN 2 'person':'intercept':'pass':'in':'NNP'
NVNPN 2 'person':'intercept':'pass':'against':'NNP'
NVNPN 1 'cornerback':'intercept':'pass':'by':'NNP'

All these queries accumulate evidence over the preposition "*by*" ("*pass:by:Steve_Walsh*").

Finding the verbs

The next exercise is to find a verb able to give meaning to syntactic dependencies such as "*Steve_Walsh:nn:pass*", "*touchdown:nn:pass*", "*Young:nn:pass*" or "*pass:to:Brent_Jones*".

We can ask the BKB what instances (NNP) do with *passes*. The most frequent propositions are:

NVN 2241 'NNP':'catch':'pass'
NVN 2106 'NNP':'throw':'pass'
NVN 844 'NNP':'complete':'pass'
NVN 434 'NNP':'intercept':'pass'
…
NVNPN 758 'NNP':'throw':'pass':'to':'NNP'
NVNPN 562 'NNP':'catch':'pass':'for':'yard'
NVNPN 338 'NNP':'complete':'pass':'to':'NNP'
NVNPN 255 'NNP':'catch':'pass':'from':'NNP'

Considering the evidence of "Brent_Jones" being instance of "*end*" (*tight end*), if we ask the BKB about the most frequent relations between "*end*" and "*pass*" we find:

NVN 28 'end':'catch':'pass'
NVN 6 'end':'drop':'pass'

So, in this case, the BKB suggests that the syntactic dependency "*pass:to:Brent_Jones*" means "*Brent Jones is an end catching a pass*". Or in other words, that "*Brent_Jones*" has a role of "*catch-ER*" with respect to "*pass*".

If we want to accumulate more evidence on this we can consider NVNPN propositions including "*touchdown*". We only find evidence for the most general classes (*NNP* and *person*):

NVNPN 189 NNP:'catch':'pass':'for':'touchdown'
NVNPN 26 NNP:'complete':'pass':'for':'touchdown'
NVNPN 84 person:catch:pass:for:touchdown
NVNPN 18 person:complete:pass:for:touchdown

This means that when we have "*touchdown*", we don't have counts for the second option "*Brent_Jones:drop:pass*", while "*catch*" becomes stronger.

In the case of "*Steve_Walsh:nn:pass*" we hypothesize that "*Steve_Walsh*" is a "*quarterback*". Asking the BKB about the most plausible relation between a *quarterback* and a *pass* we find:

NVN 98 'quarterback':'throw':'pass'
NVN 27 'quarterback':'complete':'pass'

Again, if we take into account that it is a "*touchdown:nn:pass*", then only the second option "*Steve_Walsh:complete:pass*" is consistent with the NVNPN propositions. So, in this case, the BKB suggests that the syntactic dependency "*Steve_Walsh:nn:pass*" means "*Steve_Walsh is a quarterback completing a pass*".

Finally, with respect to "*touchdown:nn:pass*", we can ask about the verbs that relate them:

NVN 14 'pass':'set_up':'touchdown'
NVN 6 'pass':'score':'touchdown'
NVN 5 'pass':'produce':'touchdown'

Figure 2 shows the resulting enrichment after the process described.

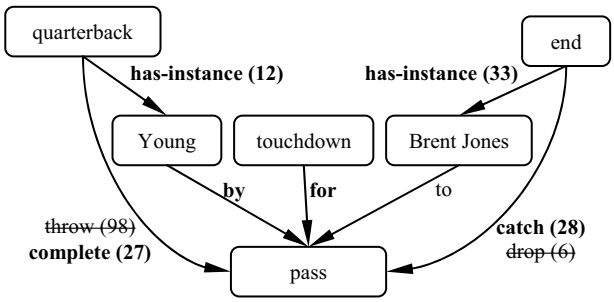


Figure 2. Enrichment of the noun phrase: "*Young touchdown pass to Brent Jones*"

4.4 Expansion of relations

Sometimes, the sentence shows a verb with more than two arguments. In our example, we have "*Eric_David:intercept:pass:on:series*". In these cases, relations can be expanded into new nodes.

Following our example, the new node is the eventuality of "*intercept*" ("*intercept-ION*"), "*Eric_Davis*" is the "*intercept-ER*" and "*pass*" is the "*intercept-ED*". Then, the missing information is attached to the new node (see Figure 3).

In addition, we can proceed with the expansion of the context considering this new node. For example, we are working with the hypothesis that "*Steve_Walsh*" is an instance of "*quarterback*" and thus, its most plausible relations with "*pass*" are "*throw*" and "*complete*". However, now we can ask about the most frequent relation between "quarterback" and a nominalization of

"intercept". The most frequent proposition is "*quarterback:throw:interception*", supported 35 times in the collection. In this way, we have inferred that the nominalization for the eventuality of intercept is interception (in our documents). Two further actions are possible: reinforce the hypothesis of "*throw:pass*" instead of "*complete:pass*" and add the hypothesis that "*Steve_Walsh:throw:interception*".

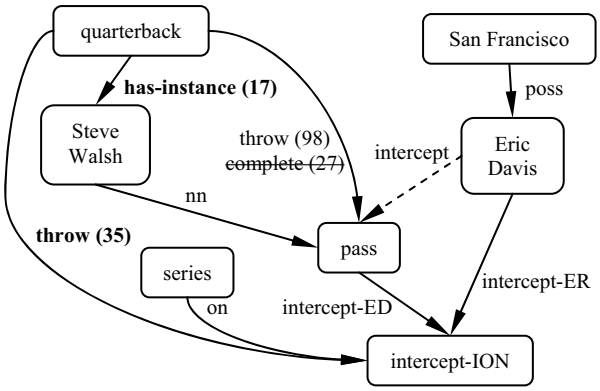


Figure 3. Expansion of "intercept" relation

Finally, notice that since "set_up" doesn't need to accommodate more arguments, we can maintain the collapsed edge.

4.5 Constraining the interpretations

Some of the inferences being performed are local in the sense that they involve only an entity and a relation. However, these local inferences must be coherent both with the sentence and the complete document. To ensure this coherence we can use additional information as a way to constrain different hypotheses. In section 4.3 we showed the use of NVNPN propositions to constrain NVN ones. Another example is the case of "*Eric_Davis:intercept:pass*". We can ask the BKB for the entity classes that participate in such kind of proposition:

NVN 75 'person':'intercept':'pass'
NVN 14 'cornerback':'intercept':'pass'
NVN 11 'defense':'intercept':'pass'
NVN 8 'safety':'intercept':'pass'
NVN 7 'group':'intercept':'pass'

So the local inference for the kind of entity "Eric_Davis" is (*cornerback*) must be coherent with the fact that it intercepted a pass. In this case "*cornerback*" and "*person*" are properly reinforced. In some sense, we are using these additional constrains as selectional preferences.

5 Evaluation

Properly evaluating the enrichment process is very difficult. Ideally, one would compare the output of an enrichment engine—a text graph fully fleshed out with additional knowledge—to a gold-standard graph containing all relevant information explicitly, and measure Recall and Precision of the links added by enrichment. But since we have no gold standard examples, and it is unclear how much knowledge should be included manually if one were to try to build some, two options remain: extrinsic evaluations and measuring the utility of the BKB in providing knowledge. We are in the process of performing an extrinsic evaluation, by measuring how much QA about the text read improves using the enriched representation. We report here the results of comparing the utility, for enrichment purposes, of two other publicly available background knowledge bases: DART (Clark and Harrison, 2009) and TextRunner (Banko et al. 2007).

5.1 Ability to answer about instances

As shown in our examples, BKBs need the ability to answer about instances and their classes. The BKBs don't need to be completely populated, but at least have enough instance-class attachments in order to allow analogy.

Neither DART nor TextRunner allow asking about possible classes for a particular instance. This is out of the scope of TextRunner. In DART, instances are replaced by one of three basic categories (person, place, organization). Although storing the original proper nouns attached to the assigned class would be straightforward, these three general classes are not enough to support inference. This leads us to the next ability.

5.2 Ability to discover new classes and relations

While *quarterbacks throw passes*, *ends* usually *catch or drop* them. As we have shown in our examples, classifying them as "*person"* or even "*player"* is not specific enough for enrichment.

Using a predefined set of entity classes doesn't seem a good approach for midterm goals. First, human abstraction is not correlated with the appropriate granularity level that enable recovering

of relevant background knowledge. Second, annotation will be needed for training.

In our Proposition Stores, we count simply what is explicitly said in the texts about our instances. This seems correlated to an appropriate level of granularity. Furthermore, an instance can be attached to several classes that can be compatible (quarterback, player, person, leader, veteran, etc.). Frequencies tell us the classes we have to consider in the first place in order to find a coherent interpretation of the text.

5.3 Ability to constrain interpretation and accumulate evidence

Enrichment must be guided by the coherence of the ensuing interpretation. For this reason BKBs must allow different types of queries over the same elements. The aim is to constrain as much as possible the relations we recover to the ones that give a coherent interpretation of the text.

As shown in our example, we require the ability to ask different syntactic contexts/structures (NN, NVNPN, etc.), not only NVN (subject-verb-object). Achieving this is very difficult for approaches that don't use parsing.

5.4 Ability to digest enough knowledge adapted to the domain

None of the abilities discussed above are relevant if the BKB doesn't contain enough knowledge about the domain in which we want to enrich documents. To evaluate, we ran three simple queries related to the US football domain in order to assess the suitability of the BKBs for enrichment: *What do quarterbacks do with passes? What do persons do with passes? Who intercepts passes?* Table 1 shows the results obtained with DART, TextRunner and our BKB.

Although DART is a general domain BKB built using parsing, its approach doesn't allow one to process enough information to answer the first question (first row in Table 1). A web scale resource such as TextRunner is better suited for this purpose. However, results show its lack of normalization. On the other hand, our BKB is able to return a clean and relevant answer.

The second question (second row) shows the ability of the three BKBs to deal with a basic abstraction needed for inference. Since TextRunner doesn't perform any kind of processing over entities or pronouns, it doesn't recover relevant knowledge for this question in the football domain. In addition, the table shows the need for domain adaptation: most of the TextRunner relations, such as "person:gets:pass" or "person:bought:pass", refer to different domains. DART shows the same effect: the first two entries ("person:make:pass", "person:take:pass") belong to different domains.

DART[1]	TextRunner[2]	BKB (Football)
(no results)	(~200) threw (~100) completed (36) to throw (26) has thrown (19) makes (19) has (18) fires	(99) throw (25) complete (7) have (5) attempt (5) not-throw (4) toss (3) release
(47) make (45) take (36) complete (30) throw (25) let (23) catch (1) make (1) expect	(22) gets (17) makes (10) has (10) receives (7) who has (7) must have (6) acting on (6) to catch (6) who buys (5) bought (5) admits (5) gives	(824) catch (546) throw (256) complete (136) have (59) intercept (56) drop (39) not-catch (37) not-throw (36) snare (27) toss (23) pick off (20) run
(13) person (6) person/ place/ organization (2) full-back (1) place	(30) Early (26) Two plays (24) fumble (20) game (20) ball (17) Defensively	(75) person (14) cornerback (11) defense (8) safety (7) group (5) linebacker

Table 1. Comparison of DART, TextRunner and our BKB for the following queries (rows): (1) *quarterback:X:pass*, (2) *person:X:pass*, (3) *X:intercept:pass*. Frequencies are in parentheses.

Finally, the third question is aimed at recovering possible agents (those that *intercept passes* in our case). Again, as shown in DART, the reduced set of classes given by the entity recognizer is not enough for the football domain. But having no classes (TextRunner) is even worse, showing its orientation to discovering relations rather than to generalizing and answering about their possible arguments. Our approach is able to discover plausible agent-classes for the query.

Other queries related to the football domain show the same behavior.

[1] Available at http://userweb.cs.utexas.edu/users/pclark/dart/

[2] After aggregating partial results for each cluster using http://www.cs.washington.edu/research/textrunner/

6 Related Work

Our approach lies between macro-reading and Open Information Extraction (OIE). Macro-reading (Mitchell et al. 2009) is a different task from ours; it seeks to populate ontologies. Here concepts and relations are predefined by the ontology.

OIE (Banko et al. 2007) does not use a predefined set of semantic classes and relations and is aimed at web scale. For this reason the framework does not include a complete NLP pipeline. The resulting lack of term normalization and absence of domain adaptation (e.g., the query *person:X:pass* return *throw* but also *buy*) makes the results less relevant to single-document reading.

When, as with DART, the complete NLP pipeline is applied over a general corpus, the amount of information to be processed has to be limited due to computational cost. Ultimately, too little knowledge remains for working in a specific domain. For example, asking DART about "*quarterback:X:pass*" produces no results.

Our approach takes advantage of both worlds, ensuring that enough amounts of documents related to the domain will be processed with a complete NLP pipeline. Doing so provides cleaner and canonical representations (our propositions) and even higher counts than TextRunner for our domain. This level of processing will be scalable in the midterm; various people including (Huang and Sagae, 2010) are working in linear time parsers with state-of-the-art performance.

Another intermediate point between a collection of domain documents and the general web, reached by restricting processing to the results of a web query, is explored in IE-on-demand (Sekine 2006; Shinyama and Sekine 2006). However, they use a predefined set of entity classes, preventing from discovering the appropriate granularity level that enables retrieval of relevant background knowledge. We do not predefine the concepts/classes and relations, but discover them from what it is explicitly said in the collection.

The process of building the BKB described here is closely related to DART (Clark and Harrison, 2009) which in turn is related to KNEXT (Van Durme and Schubert, 2008). Perhaps the most important extension we performed is the inclusion of lexical relations (like "*has-instance*") that activate more powerful uses of the Proposition Stores.

7 Conclusions and Future Work

In building a BKB, limiting oneself to a specific domain provides some powerful benefits. Ambiguity is reduced inside the domain, making counts in propositions more accurate. Also, frequency distributions of propositions differ from one domain to another. For example, the list of the most frequent NVN propositions in our BKB (see Section 3.1) is, by itself, an indication of the most salient and important events specifically in the US football domain. Furthermore, the amount of text required to build the BKB is reduced significantly allowing processing such as parsing.

The task of inferring omitted but necessary information is a significant part of automated text interpretation. In this paper we show that even simple kinds of information, gleaned relatively straightforwardly from a parsed corpus, can be quite useful. Though they are still lexical and not even starting to be semantic, propositions consisting of verbs as relations between nouns seem to provide a surprising amount of utility. It remains a research problem to determine what kinds and levels of knowledge are most useful in the long run.

In the paper, we discuss only the propositions that are grounded in instantial statements about players and events. But for true learning by reading, a system has to be able to recognize when the input expresses general rules, and to formulate such input as axioms or inferences. In addition is the significant challenge of generalizing certain kinds of instantial propositions to produce inferences. At which point, for example, should the system decide that "all football players have teams", and how should it do so? This remains a topic for future work.

A further topic of investigation is the time at which expansion should occur. Doing so at question time, in the manner of traditional task-oriented back-chaining inference, is the obvious choice, but some limited amount of forward chaining at reading time seems appropriate too, especially if it can significantly assist with text processing tasks, in the manner of expectation-driven understanding.

Finally, as discussed above, the evaluation of intrinsic evaluation procedures remains to be developed.

Acknowledgments

We are grateful to Hans Chalupsky and David Farwell for their comments and input for this work. We acknowledge the builders of TextRunner and DART for their willingness to make their resources openly available.

This work has been partially supported by the Spanish Government through the "Programa Nacional de Movilidad de Recursos Humanos del Plan Nacional de I+D+i 2008-2011" (Grant PR2009-0020). Research supported in part by Air Force Contract FA8750-09-C-0172 under the DARPA Machine Reading Program.

References

Banko, M., Cafarella, M., Soderland, S., Broadhead, M., Etzioni, O. 2007. Open Information Extraction from the Web. IJCAI 2007.

Barker, K. 2007. Building Models by Reading Texts. Invited talk at the AAAI 2007 Spring Symposium on Machine Reading, Stanford University.

Clark, P. and Harrison, P. 2009. Large-scale extraction and use of knowledge from text. The Fifth International Conference on Knowledge Capture (K-CAP 2009). http://www.cs.utexas.edu/users/pclark/dart/

Hobbs, J.R., Stickel, M., Appelt, D. and Martin, P., 1993. Interpretation as Abduction. Artificial Intelligence, Vol. 63, Nos. 1-2, pp. 69-142. http://www.isi.edu/~hobbs/interp-abduct-ai.pdf

Huang, L. and Sagae, K. 2010. Dynamic Programming for Linear-Time Shift-Reduce Parsing. ACL 2010.

Klein, D. and Manning, C.D. 2003. Accurate Unlexicalized Parsing. Proceedings of the 41st Meeting of the Association for Computational Linguistics, pp. 423-430

Marneffe, M. and Manning, C.D. 2008. The Stanford typed dependencies representation. In COLING 2008 Workshop on Cross-framework and Cross-domain Parser Evaluation.

Mitchell, T. M., Betteridge, J., Carlson, A., Hruschka, E., and Wang, R. Populating the Semantic Web by Macro-reading Internet Text. The Semantic Web - ISWC 2009. LNCS Volume 5823. Springer-Verlag.

Sekine, S. 2006. On Demand Information Extraction. COLING 2006.

Shinyama, Y. and Sekine, S. 2006. Preemptive Information Extraction using Unrestricted Relation Discovery. HLT-NAACL 2006.

Sperber, D. and Wilson, D. 1995. Relevance: Communication and cognition (2nd ed.) Oxford, Blackwell.

Van Durme, B., Schubert, L. 2008. Open Knowledge Extraction through Compositional Language Processing. Symposium on Semantics in Systems for Text Processing, STEP 2008.

Dynamic Parameters for Cross Document Coreferece

Octavian Popescu
papsi@racai.ro
Racai, Romanina Academy

Abstract

In this paper we present a new algorithm for the Person Cross Document Coreference task. We show that accurate results require a way to adapt the parameters of the similarity function – metrics and threshold – to the ontological constraints obeyed by individuals. The technique we propose dynamically changes the initial weights computed when the context is analyzed. The weight recomputation is necessary in order to resolve clusters borders, which are inevitably blurred by a static approach. The results show a significant gain in accuracy.

1 Introduction

The Person Cross Document Coreference, CDC, task requires that all the personal name mentions, PNMs, in a corpus be clustered together according to the individuals they refer to (Grishman 1994). The coreference between two PNMs is decided on the basis of the local contexts. In this paper we consider a news corpus, and the local context is the piece of news to which a particular PNM belongs. We work on a seven year Italian local newspaper corpus, Adige 500K (Magnini et. al. 2006).

While there are certain similarities between a disambiguation task and the CDC task, we maintain that there is a significant difference which sets the CDC task apart. Unlike in other disambiguation tasks, in the CDC tasks the relevant coreference context depends on the corpus itself. In word sense disambiguation, for instance, the distribution of the relevant context is mainly regulated by strong syntactic and semantic rules. The existence of such rules allows for disambiguation decisions which are made by considering the local context only. On the other hand, the distribution of the PNMs in a corpus is rather random and the relevant coreference context is a dynamic variable which depends on the diversity of the corpus, that is, on how many different persons with the same name share a similar context. Unlike the word senses which are subject to strong linguistic constraints, the name distribution is more or less random. To exemplify, consider the name "John Smith" and an organization, say "U.N.". The extent to which "works for U.N." in "John Smith works for U.N." is a relevant coreference context depends on the diversity of the corpus itself. If in that corpus, among all the "John Smiths" there is only one person who works for "U.N." then "works for U.N." is a relevant coreference context, but if there are many "John Smiths" working for U.N., then "works for U.N." is not a relevant coreference system.

In this paper we present a method to exactly determine the relevance of a piece of context for the coreference. As above, the exactness is understood in relationship with the whole system of clusters. The relevance of a piece of context is computed by means of a weighting procedure. The classic weighting procedures are static, each piece of context receives an initial value that is also a final one and the clustering proceeds on the basis of these values. We demonstrate that this approach has serious drawbacks and we argue that in order to obtain accurate results, a dynamic weighting procedure is necessary, which outputs new values depending on the cluster configuration.

In Section 2 we review the relevant literature. In Section 3 we present the problems related to the classical approach to the CDC task and we present evidence that the data distribution in a news corpus requires a proper treatment of these

Coling 2008: Poster Volume, pages 988–996,
Beijing, August 2010

problems. In Section 4 we present the technique that permits to overcome the problems identified in Section 3. In Section 5 we present the context extraction technique that supports the method developed in Section 4. In Section 6 we present the results of an evaluation experiment. The paper ends with Conclusion and Further Work section.

2 Related Work

In a classical paper (Bagga and Baldwin 1998), a PCDC system based on the vector space model (VSM) is proposed. While there are many advantages in representing the context as vectors on which a similarity function is applied, it has been shown that there are inherent limitations associated with the vectorial model (Popescu 2008). These problems, related to the density in the vectorial space (superposition) and to the discriminative power of the similarity power (masking), become visible as more cases are considered.

Testing the system on many names, (Gooi and Allan, 2004), it has been noted empirically that the accuracy of the results varies significantly from name to name. Indeed, by considering just the sentence level context, which is a strong requirement for establishing coreference, a PCDC system obtains a good score for "John Smith". This happens because the prior probability of coreference of any two "John Smiths" mentions is low, as this is a very common name and none of the "John Smiths" has an overwhelming number of mentions. But for other types of names the same system is not accurate. If it considers, for instance, "Barack Obama", the same system obtains a very low recall, as the probability of any two "Barack Obama" mentions to corefer is very high and the relevant coreference context is very often found beyond the sentence level. Without further adjustments, a vectorial model cannot resolve the problem of considering too much or too little contextual evidence in order to obtain a good precision for "John Smith" and simultaneously a good recall for "Barack Obama". These types of name have different cluster systems

In an experiment using bigrams (Pederson et al. 2005) on a news corpus, it has been observed that the relationship between the amount of information given to a CDC system and the performances is not linear. If the system has received in input the correct number of persons with the same name, the accuracy of the system has dropped. A typical case for this situation is when there is a person that is very often mentioned, and few other persons that have few mentions. When the number of clusters is passed in the input, the clusters representing the persons who are rarely mentioned are wrongly enriched. However, this situation can be avoided if there is a measure of how big the threshold should be. The system of clusters is not developed unrealistically if we are able to handle the fact that individuals obey different constraints which are derived directly from the ontological properties. These constraints are determined directly from the context and adequate weights can be set.

Recently, there has been a major interest in the CDC systems, and, in the last two years, two important evaluation campaigns have been organized: Web People Search-1 (Artiles et al. 2007) and ACE 2008 (www.nist.gov/speech/tests/ace/). It has been noted that the data variance between training and test is very high (Lefever 2007). Rather than being a particularity of those corpora, the problem is general. The performances of a bag of words VSM depends to a very high extent on the corpus diversity (see Section 3.2). For reliable results, a CDC system must have access to global information regarding the coreference space.

Rich biographic facts have been shown to improve the accuracy of CDC (Mann and Yarowsky 2003). Indeed, when available, the birth date, the occupation etc. represent a relevant coreference context because the probability that two different persons have the same name, the same birth date and the same occupation is negligible. However, it is equally unlikely to find this information in a news corpus a sufficient number of times. Even for a web corpus, where the amount of this kind of information is higher than in a news corpus, the extended biographic facts, including e-mail address, phones, etc., contribute only with approximately 3% to the total number of coreferences (Elmacioglu et al. 2007). In order to improve the performances of the CDC systems based on VSM, the special importance of pieces of context has been exploited by implementing a cascade clustering technique (Wei 2006). Other authors have relied on advanced clustering techniques (among others Han et al. 2005, Chen 2006). However, these techniques rely on the precise analysis of the context, which is a time consuming process. It has been also noted that, in spite of deep analysis, the relevant coreference context is hard to find (Vu 2007).

3 Coreference Based on Association Sets

The coreference of two PNMs is realized on the basis of the context. In a news corpus, the context surrounding each PNM, which is relevant for coreference, is extracted into a set, called association set. In Table 1 we present an example of association sets related to the same name.

Name	Associated Sets
Paolo Rossi	TV, comedian, , satire research, conference politics, meeting

Table 1: Associated Sets

A weighting schema, a global metrics and threshold are set, and the distance between two association sets is computed. The decision of coreferencing two PNMs is made on comparing the distance to the threshold and clustering the PNMs representing the same individual into a unique cluster. The accuracy of a CDC system based on association sets depends on two factors: (1) the ability to extract the relevant elements for the association sets from the news context and (2) the adequacy of the similarity formula - metrics and threshold.

Regarding the first factor, the ability to extract the relevant pieces of context, the right heuristics must be found, because the exact syntax-semantics analysis of text is unfortunately very hard or impossible to implement. A strong limitation comes from the fact that even a shallow parsing requires too much time in order to be practical. However, it has been shown that accurate parings of PNMs and co-occurring special words can be found by employing relaxed extraction techniques (Buitelaar&Magnini 2005). The association sets built in this way are effective in solving the CDC task (Sekine 2008, Popescu 2008). We make use of these findings in order to build the association sets, which mainly include named entities and certain special words, which are bound to an ontology. The details of these particular association sets are given in Section 5.

As straightforward as the classical approach based on the distance between association sets may seem, there are actually a series of problems related to the second requirement, namely the adequacy of similarity formula. We make these problems explicit below.

3.1 Masking, Superposition and Border Proximity

In order to introduce the first problem we start with an intuitive example. Suppose that we want to individuate the persons with the name Michael Jackson in a news corpus. A simplistic solution is to cluster together all such PNMs and declare that than there is just one person mentioned in the whole corpus with this name. This solution has the advantage of being very simple and of obtaining a very high score in terms of precision and recall. This is because most of such PNMs refer to only one person indeed – the pop star. However, the above method fails short when it comes to presenting the evidence for its coreference decision. Actually, it turns out that this is a very hard task, because the number of PNMs, which do not refer to the pop star, is extremely small. Thus, the prior chances of correctly finding two PNMs which do not refer to this person are quite small. Unfortunately, the classical metrics are too coarse to capture the difference in such cases, even if the association sets are 100% correct. To support this statement, let us consider three classes under the same name, with each class corresponding to a different individual. Let us further suppose that two classes contain the great majority of the PNMs, and the third class only has a small number of PNMs. A linear decision is likely to confound the elements of the third class to the ones of the first two[1]. This happens because the elements of the third class are transparent to the hyper plane that separates the two well-represented classes. This situation is called masking, and is a direct effect of applying an inaccurate weighting schema and metrics (Hastie&Tibshirani 2001). The effects of masking on the CDC task have been empirically noticed in (Pederson 2005). The main obstacle in dealing with masking is the correct treatment of the border elements. δ_{ij}, the discrimant function between two classes, i and j respectively, must assign zero to all border elements. In Section 4, we directly address this problem.

The second problem that needs to be solved by the CDC systems based on associated sets may be regarded as the negative effect of counter balancing the sparseness problem. In general, the association sets are too sparse to permit pair to pair comparison. Rather, the information must be interpolated from a set of corefered association sets. For example, in Figure 1, any two association sets chosen from the three ones on the left, AS_1, AS_2 and AS_3 respectively, are similar

[1] In fact any decision functions that can be bijectively transformed into a linear function, like most exponential kernel functions for example, are similarly prone to masking.

enough to one another to corefer. However, none of these association sets is similar enough to the one on the right – AS_4. But accepting the coreference of any initial pair, in this particular case, we implicitly accept the coreference with the fourth one.

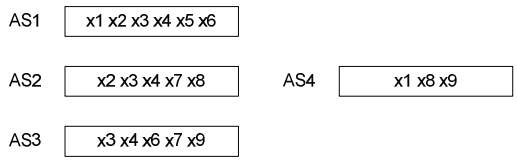


Figure 1. Interpolating

By interpolating the information in the set of the initial three association sets, the coreference becomes possible between all four association sets. In general, by interpolating from a set of the association sets, one wants to find the right coreferences and to avoid the false ones accurately. In a vector space, the interpolation is safe if the initial vectors are orthogonal to each other, because the sum of orthogonal vectors is also orthogonal to any other vector that is not part of the sum. Therefore the right coreferences have a big dot product with the sum, while the false ones have a dot product with the sum close to zero. This property of the sum of the orthogonal vectors is called superposition (Gallant 1993). By representing the association sets as vectors, where each set of vectors is associated exclusively with a certain individual, the sum of these vectors has the superposition property.

However, if the vectors representing the association sets are not orthogonal, then the interpolated vectors are prone to false coreferences. In this case, the accidental coincidences – which are responsible for the original vectors not being orthogonal – biases the dot product and introduces false coreferences. Consequently the superposition affects negatively the overall accuracy. The aggravating effect of superposition in conjunction with an agglomerative clustering procedure has been empirically noted in Gooi&Allan.

The third problem is directly related to the fact that in the most ambiguous cases the association sets lead to high-dimensional, very sparse vectors. The basic fact is that inside a cluster of correctly corefered PNMs that refer to the same individual, the distance from most of these PNMS to the center of the cluster is smaller than the distance from these PNMs to the border. Let us consider that all the *m* PNMs representing the same individual are points in an *n* dimensional vector space and their cluster is normalized to the unit sphere. The distance from the center of the sphere to the closest point is an exponentially growing formula both in *1/n* and *1/m*. Even for small values, the distance from the center to the closest point is larger than ½. The points representing the PNMs in the same cluster are closer to the border, and not to the center of the sphere. This is a secondary effect of the curse of dimensionality problem in the vector space[2].

3.2 Data Distribution

Let us consider the corpus, focusing on the distribution of PNMs. Many PNMs are the mentions of the same name, considered as a string. We are interested in the frequency with which a certain name appears. We have noticed that there is a strict relationship between the names, their frequencies and the number of mentions; see Table 2.

Freq	PNM	# PNM
1	317,245	317,245
2 – 5	166,029	467,560
6 – 20	61,570	634,309
21 – 100	25,651	1,090,836
101 – 1000	7,750	2,053,994
1001 – 2000	4,25	569,627
2001 – 4000	157	422,585
4001 – 5000	17	73,860
5001 – 31091	22	190,373

Table 2 Frequency of Names and PNMs in Adige500k

The names have a very unbalanced distribution. A name which has a frequency over 20 and is ambiguous represents a difficult case. The measure we use in order to evaluate the difficulty is the Gini's mean difference. Let X_1, X_2, …, X_n be the individuals that are named with the same name and let S be the set of the PNMs of this name PNMS, S_1, S_2, … S_n. The Gini's mean difference is a measure of the spread of the information in the set S:

$$\sum_{j=2}^{n} \sum_{i} \binom{n}{2} |Si - Sj| = G \qquad (1)$$

The uniform distribution makes Gini's factor null. A value of this factor close to 1 shows a skewed distribution. In the first case, $G \approx 0$, the superposition effect is likely to be responsible for false coreferences, while in the latter case, $G \approx 1$,

[2] The curse of dimensionality refers to the fact that the number of sample points required to state confident values for a statistics grows exponentially with the dimension of vector space.

the masking effect is predominant. However, there is a close relationship between all the three problems above. As the most ambiguous cases are near the border, it is likely that the vectors are not orthogonal and consequently the false coreferences are introduced in the system, which ultimately leads to masking.

4 Resolving the Border Condition

We are going to present a technique developed to deal with the problems identified in the previous section. The bottom line is that the weights and the threshold required by the similarity function of two association sets should be dynamically computed. In this way the border between any pair of clusters can be accurately set.

We present the procedure of adjusting the weights and the threshold for a given group of clusters in order to maximize the probability of the correct coreferences. The first step is to present the construction of the association sets, with initial weight values. The second step is to show how these initial weight values are recomputed for a set of given clusters.

Initialization

As mentioned in the first paragraph of Section 3, the association sets are built out of the surrounding context by considering the named entities, and special words. The named entities are clearly marked in the input, the corpus having being tagged by a Named Entities Recognition tool. The words considered special are identified using an ontology and the procedure is given in Section 5. The construction of the association set is a search procedure starting from the PNM. The first search space is the longest nominal group which is headed by a PNM:

uno dei falchi dell' amministrazione di Stati Uniti guidata dal presidente George W.Bush
one of the falcons of the U.S. administration lead by the president Georg W. Bush

All the special words that are present in this nominal group are included in the association set of this PNM. In this example, these special words are "president" and "administration" respectively. The named entity "U.S." is also included. These elements receive the highest weights. The search space is extended to the sentence level and new named entities/special words are included. However, unlike in the first phase, the weight of these words is determined on the basis of a second parameter, namely the number of different names interfering between the PNM and these words. We take into consideration three values 0, 1 and 2 or more. After the sentence, the next search domain is the whole news. Basically, the significance of an element decreases linearly with the distance and the number of other interfering PNMs. In Table 3 we present the linear kernel weighting schema described above. The series α_{ij} is decreasing linearly over both indexes.

	Interfering PNMs		
Domain	0	1	≥2
PNM Group	α_{11}	α_{12}	α_{13}
IN Sentence	α_{21}	α_{22}	α_{23}
Out Sentence	α_{31}	α_{32}	α_{33}

Table 3. Linear Kernel for Initial Weights

Recomputation

The association set is basically a pair of two vectors: $X = (x_1, \ldots, x_n)$ the set of words and $W = (w_1, \ldots, w_n)$ the set of the initial weights. Two PNMs corefer or not depending on whether the sum of their common part is bigger, respectively lesser than a threshold.

$$coref: \sum_{common\ xi} wi \geq T \quad (2)$$

$$non\ coref: \sum_{common\ xi} wi \leq T \quad (3)$$

Suppose now that we have an independent way to know the truth regarding the coreference. Then, we have to readjust the initial weights such that the real configuration of clusters is promoted also by Equations (2) and (3). For clarity, let us give an example: suppose that we know that in our corpus there is only one person named "Roberto Bizzo" and only one person named "Roberto Cuillo", and no other person is called "Roberto". Consequently the PNMs "Roberto" are clustered to the clusters "Robert Bizzo" xor "Roberto Cuillo". Suppose further that the named entity "Roma" is associated with some of the PNMs "Roberto". If only "Roberto Bizzo" is associated with "Roma", then the coreference between those "Roberto" associated with "Roma" and "Roberto Bizzo" can be made. However, it is often the

case that both "Roberto Bizzo" and "Roberto Cuillo" are associated with "Roma", which has its particular weight for each PNM. In this case this named entity, "Roma", may bear no relevance for the coreference of "Roberto" in either of the clusters. Consequently, whatever the initial value for "Roma" in certain association sets, it must be nullified. In order to find out which elements of the association sets are relevant, and what weights the relevant elements must have, we propose the following strategy: we replace the "Roberto Bizzo" with "Roberto X", and "Roberto Cuillo" with "Roberto X". We obtain a big set of association sets corresponding to the PNMs "Roberto X". We reweight the elements of their association sets and the threshold, such that, from this set of association sets, we obtain exactly two clusters, one that is identical with "Roberto Bizzo", and one that is identical with "Roberto Cuillo". Conceptually, this strategy is similar to the pseudo words technique used in building test corpora. After the reweighting of the elements associated with "Robert Bizzo" and "Roberto Cuillo" respectively, we can associate the simple PNMs "Roberto" to one of these two clusters.

In the above example we make use of the fact that if two persons have different last names then they are different persons. This is a prior ontological constraint. In fact, whenever we know the set of ontological constraints that correctly cluster a set of PNMs in two or more clusters, we can intentionally confound the PNMs, recompute the weights and the thresholds of their association sets, in order to obtain the initial cluster configuration. Now we use the new computed values to cluster new PNMs whose relationship with the ontological constraints could not have been determined from the corpus.

We show that we can use the Simplex method to recompute the initial weights. Indeed, by intentionally confounding a system of clusters, we determine the coefficients which, when multiplied with the initial weights, lead to the correct clustering. These coefficients are the solution to a set of inequalities like those presented in Equations (2), and (3). The objective function in Simplex is a max or a min depending on whether we know that the PNMs corefer or not: if they do not corefer then there is a max Simplex system, and the threshold is just higher than the value of the objective function. Let us give an example. Suppose we have the following configuration, where AS_i represents the association set of the PNM_i, where wi is the vector of the initial weights and T is the threshold: .

$AS_1 \cap AS_2 = \{x_1, x_2, x_3\}$	wi = (1, 2, 2)	T = 7	No Coreference	$x_1 + 2x_2 + 2x_3 \leq 7$
$AS_1 \cap AS_3 = \{x_1, x_3\}$	wi = (5,0,4)	T = 11	Coreference	$5x_1 + 4x_3 \geq 11$
$AS_2 \cap AS_4 = \{x_2, x_3\}$	wi = (0,3,4)	T = 9	Coreference	$3x_2 + 4x_3 \geq 9$
$AS_5 \cap AS_6 = \{x_1, x_2\}$	w i= (2,1,0)	T = ?	No Coreference	max $(2x_1 + x_2)$

The above cluster configuration leads to the following Simplex system:

$$\max 2x_1 + x_2$$

$$\begin{cases} x1 + 2x2 + 2x3 \leq 7 \\ 5x1 + 4x3 \geq 11 \\ 3x2 + 4x3 \geq 9 \end{cases}$$

which has the solution wr = (1.55, 1.91, 0.82) with max = 5. Therefore the initial weights for the elements x_1, x_2, x_3 must be multiplied with 1.55, 1.91, 0.82 respectively and the appropriate threshold for making a decision is 5.01.

5 Ontological Constrained Association Sets

In the preceding section we presented a strategy based on Simplex Algorithm developed for the border weight assignment. The similarity formula is recomputed such that a set of ontological restriction is satisfied. In this section we present the way the set of ontological restrictions is found. The set of special words is identified on the basis of an ontology. We have used SUMO (Niles 2003) because it has the advantage that its hierarchies are connected to the WordNet, which is a Multilanguage aligned resource. Below we present the main categories of the SUMO attributes used. Summing up, there are more than 7 000 special words taken into account.

Corporation
Organization
Occupational Role
Occupies Position
Social Interaction
Social Role
Unemployed

There are mainly three different ways to create the set of ontological restrictions: fixed, prior ontological constraints, local restrictions and exclusive ontological relationships.

The fixed, prior ontological constraints are those that tend to be expressed in a fixed pattern, making it easy to identify them in the context. Usually they express the date and place of birth,

contact information, but also the gender, the family relationship, the ethnic group etc.

The local restrictions are a very rich source of information. It has been argued that inside each piece of news the coreference of all the PNMs is a valid procedure, with more than 99% accuracy (Popescu et al. 2008). By comparing the structure of the largest nominal group headed by two locally corefered PNMs we can found ontological compatibilities. Table 4 shows a sample of the compatible pairs as extracted from corpus. These pairs can be used successfully for coreferencing purposes, but these do not form ontological hierarchies and cannot be used to build inference chains.

Pairs of compatible professions
albergatore comerciante
ala giocatore
agronomo professore
allenatore mister
alpinista guida alpina
architetto progettista
arcivescovo monsignore
monsignore teologo
monsignore sacerdote
assessore consigliere

Table 4. Compatible Occupational Role

The exclusive ontological relationships are given explicitly under the form of rules. These rules stipulate what is ontologically inacceptable. We have seen an example of such rules referring to the family names in Section 4. The Occupational Role and Social Role attributes are one of the most useful exclusive ontological ones, because they are frequently mentioned in a news corpus. In average, local information at the news level produces a special word from the above categories in approximately in 30% of cases (Magnini et al 2006.). An example of the realization of the exclusive rules for a sample of multi pairs of words as extracted from corpus is presented below:

Secretary≠Priest≠Judge
Architect≠Attorney
Waiter≠Manager
Actor≠Researcher

The system of clusters determined using the technique described in Section 4 obeys the set of these constraints. The set C of ontological constraints are used to generate active rules at the word level, which, by means of fixed text patterns, are compared against the association sets. This permits the realization of ontological motivated cluster systems, which in combination with the technique of reweighting presented, leads to accurate new coreferences outside the scope of C, while avoiding the border problems presented in Section 3 ..

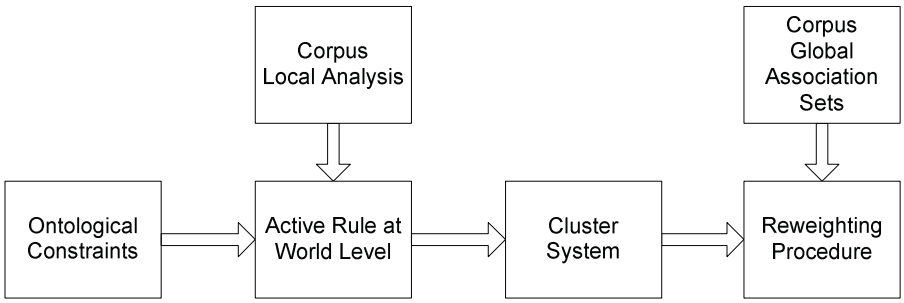


Figure 2. The dynamic reweighting schema flow

6 Evaluation

The technique we propose is designed for an accurate border detection between clusters of ambiguous names. We created a sample of the ambiguous names. For each name we computed the Gini's mean difference using the formula introduced in Section 3, which gives an indication of the spread of information relevant for coreference. We have noticed that there is a strong correlation between the Gini's mean difference and the difficulty of a coreference system. The names chosen for this experiment are such that the Gini's factor uniformly distributed in (0,1). However, the number of PNMs for each name is bigger than the number of individuals having that name. The choice is motivated by the fact that these are the most difficult cases for a CDC system, as they require strong and consistent evidence for accurate results. The opposite cases, when the number of the individuals is close to the number of PNMs or the Gini's coefficient is

close to 0 or 1, can be approached with a pure statistical approach (Popescu 2009).
The first column in Table 5 lists the names, the second column lists the number of the PNMs considered for each name, the third column lists the number of individuals having the respective name, the fourth column lists the number of PMNs for each individual, the fifth column lists the Gini's factor and the sixth column lists how many clusters have been found obeying ontological constraints/ and how many PNMs have been clustered in these clusters.

Name	#PNMs	#P	Distribution	Gini	Constraints
Angelo Elia	58	5	{20,24,7,2,2,3}	.428	2 / 18
Gifuni	89	3	{47,21,31}	.175	3/ 12
Giuseppe Rossi	185	12	{69,32,5,9,4,5,6,6,12,7,8,22}	.503	5 / 38
Paulo Rossi	137	9	{91,17,9,3,2,3,5,5,2}	.673	3 / 74
Schlesinger	62	4	{26,19,6,11}	.274	4 / 19
Tanzi	370	3	{315,49,16}	.524	3/129

Table 5. Name Test Set

We compare the technique proposed in Section 4 (DYN) against three different approaches: the first is a no weight coreference, requiring a fix number of similar elements in the association set (NOW), the second is Baga&Baldwin quadratic metric formula at sentence level (BB), and the third is an agglomerative vector space clustering algorithm as in Gooi&Allan(GA). All these three approaches use fixed similarity parameters.
The evaluation is done using the B-CUBED algorithm (Baga&Baldwin). The results, computed with F formula, are presented in Table 6.

Name	NW	BB	GA	DYN
Angelo Elia	.426	.639	.684	.672
Gifuni	.53	.635	.661	.726
Giuseppe Rossi	.481	.619	.589	.673
Paulo Rossi	.446	.623	.598	.691
Schlesinger	.528	.588	.723	.829
Tanzi	.572	.539	.699	.815
Average	**.417**	**.607**	**.659**	**.734**

Table 6. F-formula on B-CUBED

The BB and GA have been tested on the John Smith corpus, which contains the PNMs of just one name, John Smith. As John Smith is a very common name and no famous person carries it, this corpus is rather biased as the Gini's factor is small; that is why BB performs better than GA on "Giuseppe Rossi" and "Paulo Rossi". The DYN scores the best , gaining in average 7 points in F formula.

Conclusion and Further Work

In this paper we present a new technique for the CDC task which allows us to dynamically change the weights in the association sets in order to accurately account for border cases. As we showed in Section 3, the border cases are actually the most important ones due to the high dimensionality of the vector space which models the association sets.

The results we have obtained are superior to other approaches. We think that this is possible because the technique we used directly addresses the problem related to masking and superposition.

We plan to further study this technique by following mainly three directions. First, we want to study further the behavior of masking and superposition within a larger test corpus. Second, we want to extend the set of exclusive ontological relationships which can be determined from the context with shallow text analysis. Third, we want to understand better the ways in which the set of ontological constraints interact with the vector space in order to increase the overall accuracy of the coreference system.

A secondary effect of the proposed technique is that a stronger control of the inferences resulting from a cluster system can be obtained. In the future this seems to be a promising method to link the coreference tasks to the chain of inferences.

References

J. Artiles, Gonzalo, J., S. Sekine. 2007. *Establishing a benchmark for WePS*. In Proceedings of SemEval.

A. Bagga, B. Baldwin. 1998. *Entity-based Cross-Document Co-referencing using the Vector Space Model*. In Proceedings of ACL.

J. Chen, D. Ji, C. Tan, Z. Niu. 2006. *Unsupervised Relation Disambiguation Using Spectral Clustering*. In Proceedings of COLING

C. Gooi, J. Allan. 2004. *Cross-Document Coreference on a Large Scale Corpus*. In Proceedings of ACL.

G. Mann, D. Yarowsky. 2003. *Unsupervised Name Disambiguation*, in Proceeding of HLT-NAACL

I. Niles, A. Pease, 2003. *Linking Lexicons and Ontologies: Mapping WordNet to the Suggested Upper Merged Ontology*, in Proceeding IKE

R. Grishman. 1994. *Whither Written Language Evaluation?* In Proceedings of Human Language Technology Workshop, pp. 120-125. San Mateor.

E. Elmacioglu, Y. M. F. M.Y.Khan, D. Lee. 2007. *PSNUS: Web People Name Disambiguation by Simple Clustering with Rich Features,* in Proceedings of SemEval

H. Han, W. Xu. 2005. *A Hierarchical Bayes Mixture Model for Name Disambiguation in Author Citations*, in Proceedings of SAC'05

E. Lefever, V. Hoste, F. Timur. 2007. *AUG: A Combined Classification and Clustering Approach for Web People Disambiguation*, In Proceedings of SemEval

B. Magnini, M. Speranza, M. Negri, L. Romano, R. Sprugnoli. 2006. *I-CAB – the Italian Content Annotation Bank*. LREC 2006

V., Ng. 2007. *Shallow Semantics for Coreference Resolution*, In Proceedings of IJCAI

T. Pedersen, A. Purandare, A. Kulkarni. 2005. *Name Discrimination by Clustering Similar Contexts*, in Proceeding of CICLING

O. Popescu, C. Girardi. 2008. *Improving Cross Document Coreference*, in Proceedings of JADT

O. Popescu, B. Magnini. 2007. *Inferring Coreference among Person Names in a Large Corpus of News Collection*, in Proceedings of AIIA

O. Popescu 2009. *Name Perplexity*. In Proceedings of NAACL HLT

P. Buitelaar, B. Magnini (Eds.) 2005. *Ontology Learning from Text: Methods*, Evaluation and applications. *IOS Press*

Q. Vu, T. Massada, A. Takasu, J. Adachi. 2007. *Using Knowledge Base to Disambiguate Personal names in Web Search Results*, In Proceedings of SAC

T. Hastie, R. Tibshirani, J. Friedman, *2001. The elements of Statistical Learnig*, Springer Press

S. Gallant, *Neural Network Learning*, MIT Press

S. Sekine, 2008 *Extended Named Entity Ontology with Attribute Information*, in Proceeding of LREC

Y. Wei, M. Lin, H. Chen. 2006. *Name Disambiguation in Person Information Mining,* in Proceedings of IEEE

An Evaluation Framework for Plagiarism Detection

Martin Potthast **Benno Stein**
Web Technology & Information Systems
Bauhaus-Universität Weimar
{martin.potthast, benno.stein}@uni-weimar.de

Alberto Barrón-Cedeño **Paolo Rosso**
Natural Language Engineering Lab—ELiRF
Universidad Politécnica de Valencia
{lbarron, prosso}@dsic.upv.es

Abstract

We present an evaluation framework for plagiarism detection.[1] The framework provides performance measures that address the specifics of plagiarism detection, and the PAN-PC-10 corpus, which contains 64 558 *artificial* and 4 000 *simulated* plagiarism cases, the latter generated via Amazon's Mechanical Turk. We discuss the construction principles behind the measures and the corpus, and we compare the quality of our corpus to existing corpora. Our analysis gives empirical evidence that the construction of tailored training corpora for plagiarism detection can be automated, and hence be done on a large scale.

1 Introduction

The lack of an evaluation framework is a serious problem for every empirical research field. In the case of plagiarism detection this shortcoming has recently been addressed for the first time in the context of our benchmarking workshop PAN [15, 16]. This paper presents the evaluation framework developed in the course of the workshop. But before going into details, we survey the state of the art in evaluating plagiarism detection, which has not been studied systematically until now.

1.1 A Survey of Evaluation Methods

We have queried academic databases and search engines to get an overview of all kinds of contributions to automatic plagiarism detection. Altogether 275 papers were retrieved, from which 139 deal with plagiarism detection in text, 123 deal with plagiarism detection in code, and 13 deal with other media types. From the papers related to text and code we analyzed the 205 which present evaluations. Our analysis covers the following aspects: experiment tasks, performance measures, underlying corpora, and, whether comparisons to other plagiarism detection approaches were conducted. Table 1 summarizes our findings.

Table 1: Summary of the plagiarism detection evaluations in 205 papers, from which 104 deal with text and 101 deal with code.

Evaluation Aspect	Text	Code
Experiment Task		
local collection	80%	95%
Web retrieval	15%	0%
other	5%	5%
Performance Measure		
precision, recall	43%	18%
manual, similarity	35%	69%
runtime only	15%	1%
other	7%	12%
Comparison		
none	46%	51%
parameter settings	19%	9%
other algorithms	35%	40%

Evaluation Aspect	Text	Code
Corpus Acquisition		
existing corpus	20%	18%
homemade corpus	80%	82%
Corpus Size [# documents]		
$[1, 10)$	11%	10%
$[10, 10^2)$	19%	30%
$[10^2, 10^3)$	38%	33%
$[10^3, 10^4)$	8%	11%
$[10^4, 10^5)$	16%	4%
$[10^5, 10^6)$	8%	0%

With respect to the experiment tasks the majority of the approaches perform overlap detection by exhaustive comparison against some locally stored document collection—albeit a Web retrieval scenario is more realistic. We explain this shortcoming by the facts that the Web cannot be utilized easily as a corpus, and, that in the case of code plagiarism the focus is on collusion detection in student courseworks. With respect to performance measures the picture is less clear: a manual result evaluation based on similarity measures is used about the same number of times for text (35%), and even more often for code (69%), as an automatic computation of precision and recall. 21% and 13% of the evaluations on text and code use custom measures or examine only the de-

[1] The framework is available free of charge at `http://www.webis.de/research/corpora`.

Coling 2008: Poster Volume, pages 997–1005,
Beijing, August 2010

tection runtime. This indicates that precision and recall may not be well-defined in the context of plagiarism detection. Moreover, comparisons to existing research are conducted in less than half of the papers, a fact that underlines the lack of an evaluation framework.

The right-hand side of Table 1 overviews two corpus-related aspects: the use of existing corpora versus the use of handmade corpora, and the size distribution of the used corpora. In particular, we found that researchers follow two strategies to compile a corpus. Small corpora (<1 000 documents) are built from student courseworks or from arbitrary documents into which plagiarism-alike overlap is manually inserted. Large corpora (>1 000 documents) are collected from sources where overlap occurs more frequently, such as rewritten versions of news wire articles, or from consecutive versions of open source software. Altogether, we see a need for an open, commonly used plagiarism detection corpus.

1.2 Related Work

There are a few surveys about automatic plagiarism detection in text [7, 8, 14] and in code [12, 17, 19, 20]. These papers, as well as nearly all papers of our survey, omit a discussion of evaluation methodologies; the following 4 papers are an exception.

In [21] the authors introduce graph-based performance measures for code plagiarism detection that are intended for unsupervised evaluations. We argue that evaluations in this field should be done in a supervised manner. An aside: the proposed measures have not been adopted since their first publication. In [15] we introduce preliminary parts of our framework. However, the focus of that paper is less on methodology but on the comparison of the detection approaches that were submitted to the first PAN benchmarking workshop. In [9, 10] the authors report on an unnamed corpus that comprises 57 cases of simulated plagiarism. We refer to this corpus as the Clough09 corpus; a comparison to our approach is given later on. Finally, a kind of related corpus is the METER corpus, which has been the only alternative for the text domain up to now [11]. It comprises 445 cases of text reuse among 1 716 news articles. Although the corpus can be used to evaluate plagiarism detection its design does not support this task. This is maybe the reason why it has not been used more often. Furthermore, it is an open question whether or not cases of news reuse differ from plagiarism cases where the plagiarists strive to remain undetected.

1.3 Contributions

Besides the above survey, the contributions of our paper are threefold: Section 2 presents formal foundations for the evaluation of plagiarism detection and introduces three performance measures. Section 3 introduces methods to create artificial and simulated plagiarism cases on a large scale, and the PAN-PC-10 corpus in which these methods have been operationalized. Section 4 then compares our corpus with the Clough09 corpus and the METER corpus. The comparison reveals important insights for the different kinds of text reuse in these corpora.

2 Plagiarism Detection Performance

This section introduces measures to quantify the precision and recall performance of a plagiarism detection algorithm; we present a micro-averaged and a macro-averaged variant. Moreover, the so-called detection granularity is introduced, which quantifies whether the contiguity between plagiarized text passages is properly recognized. This concept is important: a low granularity simplifies both the human inspection of algorithmically detected passages as well as an algorithmic style analysis within a potential post-process. The three measures can be applied in isolation but also be combined into a single, overall performance score. A reference implementation of the performance measures is distributed with our corpus.

2.1 Precision, Recall, and Granularity

Let d_{plg} denote a document that contains plagiarism. A *plagiarism case* in d_{plg} is a 4-tuple $s = \langle s_{\text{plg}}, d_{\text{plg}}, s_{\text{src}}, d_{\text{src}} \rangle$, where s_{plg} is a plagiarized passage in d_{plg}, and s_{src} is its original counterpart in some source document d_{src}. Likewise, a *plagiarism detection* for document d_{plg} is denoted as $r = \langle r_{\text{plg}}, d_{\text{plg}}, r_{\text{src}}, d'_{\text{src}} \rangle$; r associates an allegedly plagiarized passage r_{plg} in d_{plg} with

a passage r_{src} in d'_{src}. We say that r *detects* s iff $r_{\text{plg}} \cap s_{\text{plg}} \neq \emptyset$, $r_{\text{src}} \cap s_{\text{src}} \neq \emptyset$, and $d'_{\text{src}} = d_{\text{src}}$. With regard to a plagiarized document d_{plg} it is assumed that different plagiarized passages of d_{plg} do not intersect; with regard to detections for d_{plg} no such restriction applies. Finally, S and R denote sets of plagiarism cases and detections.

While the above 4-tuples resemble an intuitive view of plagiarism detection we resort to an equivalent, more concise view to simplify the subsequent notations: a document d is represented as a set of references to its characters $\mathbf{d} = \{(1, d), \ldots, (|d|, d)\}$, where (i, d) refers to the i-th character in d. A plagiarism case s can then be represented as $\mathbf{s} = \mathbf{s}_{\text{plg}} \cup \mathbf{s}_{\text{src}}$, where $\mathbf{s}_{\text{plg}} \subseteq \mathbf{d}_{\text{plg}}$ and $\mathbf{s}_{\text{src}} \subseteq \mathbf{d}_{\text{src}}$. The characters referred to in $\mathbf{s}_{\text{plg}}$ and $\mathbf{s}_{\text{src}}$ form the passages s_{plg} and s_{src}. Likewise, a detection r can be represented as $\mathbf{r} = \mathbf{r}_{\text{plg}} \cup \mathbf{r}_{\text{src}}$. It follows that r detects s iff $\mathbf{r}_{\text{plg}} \cap \mathbf{s}_{\text{plg}} \neq \emptyset$ and $\mathbf{r}_{\text{src}} \cap \mathbf{s}_{\text{src}} \neq \emptyset$. Based on these representations, the micro-averaged precision and recall of R under S are defined as follows:

$$prec_{\text{micro}}(S, R) = \frac{|\bigcup_{(s,r) \in (S \times R)} (\mathbf{s} \sqcap \mathbf{r})|}{|\bigcup_{r \in R} \mathbf{r}|}, \quad (1)$$

$$rec_{\text{micro}}(S, R) = \frac{|\bigcup_{(s,r) \in (S \times R)} (\mathbf{s} \sqcap \mathbf{r})|}{|\bigcup_{s \in S} \mathbf{s}|}, \quad (2)$$

$$\text{where} \quad \mathbf{s} \sqcap \mathbf{r} = \begin{cases} \mathbf{s} \cap \mathbf{r} & \text{if } r \text{ detects } s, \\ \emptyset & \text{otherwise.} \end{cases}$$

The macro-averaged precision and recall are unaffected by the length of a plagiarism case; they are defined as follows:

$$prec_{\text{macro}}(S, R) = \frac{1}{|R|} \sum_{r \in R} \frac{|\bigcup_{s \in S} (\mathbf{s} \sqcap \mathbf{r})|}{|\mathbf{r}|}, \quad (3)$$

$$rec_{\text{macro}}(S, R) = \frac{1}{|S|} \sum_{s \in S} \frac{|\bigcup_{r \in R} (\mathbf{s} \sqcap \mathbf{r})|}{|\mathbf{s}|}, \quad (4)$$

Besides precision and recall there is another concept that characterizes the power of a detection algorithm, namely, whether a plagiarism case $s \in S$ is detected as a whole or in several pieces. The latter can be observed in today's commercial plagiarism detectors, and the user is left to combine these pieces to a consistent approximation of s. Ideally, an algorithm should report detections R in a one-to-one manner to the true cases S. To capture this characteristic we define the detection granularity of R under S:

$$gran(S, R) = \frac{1}{|S_R|} \sum_{s \in S_R} |R_s|, \quad (5)$$

where $S_R \subseteq S$ are cases detected by detections in R, and $R_s \subseteq R$ are the detections of a given s:

$$S_R = \{s \mid s \in S \wedge \exists r \in R : r \text{ detects } s\},$$
$$R_s = \{r \mid r \in R \wedge r \text{ detects } s\}.$$

The domain of $gran(S, R)$ is $[1, |R|]$, with 1 indicating the desired one-to-one correspondence and $|R|$ indicating the worst case, where a single $s \in S$ is detected over and over again.

Precision, recall, and granularity allow for a partial ordering among plagiarism detection algorithms. To obtain an absolute order they must be combined to an overall score:

$$plagdet(S, R) = \frac{F_\alpha}{\log_2(1 + gran(S, R))}, \quad (6)$$

where F_α denotes the F_α-Measure, i.e., the weighted harmonic mean of precision and recall. We suggest using $\alpha = 1$ (precision and recall equally weighted) since there is currently no indication that either of the two is more important. We take the logarithm of the granularity to decrease its impact on the overall score.

2.2 Discussion

Plagiarism detection is both a retrieval task and an extraction task. In light of this fact not only retrieval performance but also extraction accuracy becomes important, the latter of which being neglected in the literature. Our measures incorporate both. Another design objective of our measures is the minimization of restrictions imposed on plagiarism detectors. The overlap restriction for plagiarism cases within a document assumes that a certain plagiarized passage is unlikely to have more than one source. Imprecision or lack of evidence, however, may cause humans or algorithms to report overlapping detections, e.g., when being unsure about the true source of a plagiarized passage. The measures (1)-(4) provide for a sensible treatment of this fact since the set-based

passage representations eliminate duplicate detections of characters. The macro-averaged variants allot equal weight to each plagiarism case, regardless of its length. Conversely, the micro-averaged variants favor the detection of long plagiarism passages, which are generally easier to be detected. Which of both is to be preferred, however, is still an open question.

3 Plagiarism Corpus Construction

This section organizes and analyzes the practices that are employed—most of the time implicitly—for the construction of plagiarism corpora. We introduce three levels of *plagiarism authenticity*, namely, real plagiarism, simulated plagiarism, and artificial plagiarism. It turns out that simulated plagiarism and artificial plagiarism are the only viable alternatives for corpus construction. We propose a new approach to scale up the generation of simulated plagiarism based on crowdsourcing, and heuristics to generate artificial plagiarism. Moreover, based on these methods, we compile the PAN plagiarism corpus 2010 (PAN-PC-10) which is the first corpus of its kind that contains both a large number and a high diversity of artificial and simulated plagiarism cases.

3.1 Real, Simulated, and Artificial Plagiarism

Syntactically, a plagiarism case is the result of copying a passage s_{src} from a source document into another document d_{plg}. Since verbatim copies can be detected easily, plagiarists often rewrite s_{src} to obfuscate their illegitimate act. This behavior must be modeled when constructing a training corpus for plagiarism detection, which can be done at three levels of authenticity. Ideally, one would secretly observe a large number of plagiarists and use their *real plagiarism* cases; at least, one could resort to plagiarism cases which have been detected in the past. The following aspects object against this approach:

- The distribution of detected real plagiarism is skewed towards ease of detectability.
- The acquisition of real plagiarism is expensive since it is often concealed.
- Publishing real cases requires the consents from the plagiarist and the original author.
- A public corpus with real cases is questionable from an ethical and legal viewpoint.
- The anonymization of real plagiarism is difficult due to Web search engines and authorship attribution technology.

It is hence more practical to let people create plagiarism cases by "purposeful" modifications, or to tap resources that contain similar kinds of text reuse. We subsume these strategies under the term *simulated plagiarism*. The first strategy has often been applied in the past, though on a small scale and without a public release of the corpora; the second strategy comes in the form of the METER corpus [11]. Note that, from a psychological viewpoint, people who simulate plagiarism act under a different mental attitude than plagiarists. From a linguistic viewpoint, however, it is unclear whether real plagiarism differs from simulated plagiarism.

A third possibility is to generate plagiarism algorithmically [6, 15, 18], which we call *artificial plagiarism*. Generating artificial plagiarism cases is a non-trivial task if one requires semantic equivalence between a source passage s_{src} and the passage s_{plg} that is obtained by an automatic obfuscation of s_{src}. Such semantics-preserving algorithms are still in their infancy; however, the similarity computation between texts is usually done on the basis of document models like the bag of words model and not on the basis of the original text, which makes obfuscation amenable to simpler approaches.

3.2 Creating Simulated Plagiarism

Our approach to scale up the creation of simulated plagiarism is based on Amazon's Mechanical Turk, AMT, a commercial crowdsourcing service [3]. This service has gathered considerable interest, among others to recreate TREC assessments [1], but also to write and translate texts [2].

We offered the following task on the Mechanical Turk platform: *Rewrite the original text found below* [on the task Web page] *so that the rewritten version has the same meaning as the original, but with a different wording and phrasing. Imagine a scholar copying a friend's homework just before class, or imagine a plagiarist willing to use the*

Table 2: Summary of 4 000 Mechanical Turk tasks completed by 907 workers.

Worker Demographics			
Age		*Education*	
18, 19	10%	HS	11%
20–29	37%	College	30%
30–39	16%	BSc.	17%
40–49	7%	MSc.	11%
50–59	4%	Dr.	2%
60–69	1%		
n/a	25%	n/a	29%
Native Speaker		*Gender*	
yes	62%	male	37%
no	14%	female	39%
n/a	23%	n/a	24%
Prof. Writer		*Plagiarized*	
yes	10%	yes	16%
no	66%	no	60%
n/a	24%	n/a	25%

Task Statistics	
Tasks per Worker	
average	15
std. deviation	20
minimum	1
maximum	103
Work Time (minutes)	
average	14
std. deviation	21
minimum	1
maximum	180
Compensation	
pay per task	0.5 US$
rejected results	25%

original text without proper citation.

Workers were required to be fluent in English reading and writing, and they were informed that every result was to be reviewed. A questionnaire displayed alongside the task description asked about the worker's age, education, gender, and native speaking ability. Further we asked whether the worker is a professional writer, and whether he or she has ever plagiarized. Completing the questionnaire was optional in order to minimize false answers, but still, these numbers have to be taken with a grain of salt: the Mechanical Turk is not the best environment for such surveys. Table 2 overviews the worker demographics and task statistics. The average worker appears to be a well-educated male or female in the twenties, whose mother tongue is English. 16% of the workers claim to have plagiarized at least once, and if at least the order of magnitude of the latter number can be taken seriously this shows that plagiarism is a prevalent problem.

A number of pilot experiments were conducted to determine the pay per task, depending on the text length and the task completion time: for 50 US-cents about 500 words get rewritten in about half an hour. We observed that decreasing or increasing the pay per task has proportional effect on the task completion time, but not on the result quality. This observation is in concordance with earlier research [13]. Table 3 contrasts a source passage s_{src} and its rewritten, plagiarized passage s_{plg} obtained via the Mechanical Turk.

3.3 Creating Artificial Plagiarism

To create artificial plagiarism, we propose three obfuscation strategies. Given a source passage s_{src} a plagiarized passage s_{plg} can be created as follows (see Table 4):

- *Random Text Operations.* s_{plg} is created from s_{src} by shuffling, removing, inserting, or replacing words or short phrases at random. Insertions and replacements are taken from the document d_{plg} where s_{plg} is to be inserted.
- *Semantic Word Variation.* s_{plg} is created from s_{src} by replacing words by one of their synonyms, antonyms, hyponyms, or hypernyms, chosen at random. A word is kept if none of them is available.

Table 3: Example of a simulated plagiarism case s, generated with Mechanical Turk.

Source Passage s_{src}	**Plagiarized Passage** s_{plg}
The emigrants who sailed with Gilbert were better fitted for a crusade than a colony, and, disappointed at not at once finding mines of gold and silver, many deserted; and soon there were not enough sailors to man all the four ships. Accordingly, the Swallow was sent back to England with the sick; and with the remainder of the fleet, well supplied at St. John's with fish and other necessaries, Gilbert (August 20) sailed south as far as forty-four degrees north latitude. Off Sable Island a storm assailed them, and the largest of the vessels, called the Delight, carrying most of the provisions, was driven on a rock and went to pieces. [Excerpt from "*Abraham Lincoln: A History*" by John Nicolay and John Hay.]	The people who left their countries and sailed with Gilbert were more suited for fighting the crusades than for leading a settled life in the colonies. They were bitterly disappointed as it was not the America that they had expected. Since they did not immediately find gold and silver mines, many deserted. At one stage, there were not even enough man to help sail the four ships. So the Swallow was sent back to England carrying the sick. The other fleet was supplied with fish and the other necessities from St. John. On August 20, Gilbert had sailed as far as forty-four degrees to the north latitude. His ship known as the Delight, which bore all the required supplies, was attacked by a violent storm near Sable Island. The storm had driven it into a rock shattering it into pieces.

Table 4: Examples of the obfuscation strategies.

Obfuscation Examples
Original Text
The quick brown fox jumps over the lazy dog.
Manual Obfuscation (by a human)
Over the dog which is lazy jumps quickly the fox which is brown. Dogs are lazy which is why brown foxes quickly jump over them. A fast auburn vulpine hops over an idle canine.
Random Text Operations
over The. the quick lazy dog <*context word*> jumps brown fox over jumps quick brown fox The lazy. the brown jumps the. quick dog The lazy fox over
Semantic Word Variation
The quick brown dodger leaps over the lazy canine. The quick brown canine jumps over the lazy canine. The quick brown vixen leaps over the lazy puppy.
POS-preserving Word Shuffling
The brown lazy fox jumps over the quick dog. The lazy quick dog jumps over the brown fox. The brown lazy dog jumps over the quick fox.

- *POS-preserving Word Shuffling.* The sequence of parts of speech in s_{src} is determined and s_{plg} is created by shuffling words at random while retaining the original POS sequence.

To generate different degrees of obfuscation the strategies can be adjusted by varying the number of operations made on s_{src}, and by limiting the range of affected phrases within s_{src}. For our corpus, the strategies were combined and adjusted to match an intuitive understanding of a "low" and a "high" obfuscation. Of course other obfuscation strategies are conceivable, e.g., based on automatic paraphrasing methods [4], but for performance reasons simple strategies are preferred at the expense of readability of the obfuscated text.

3.4 Overview of the PAN-PC-10

To compile the PAN plagiarism corpus 2010, several other parameters besides the above plagiarism obfuscation methods have been varied. Table 5 gives an overview.

The documents used in the corpus are derived from books from the Project Gutenberg.[2] Every document in the corpus serves one of two purposes: it is either used as a source for plagiarism or as a document suspicious of plagiarism. The latter documents divide into documents that actually contain plagiarism and documents that don't.

[2] http://www.gutenberg.org

Table 5: Corpus statistics of the PAN-PC-10 for its 27 073 documents and 68 558 plagiarism cases.

Document Statistics	
Document Purpose	
source documents	50%
suspicious documents	
– with plagiarism	25%
– w/o plagiarism	25%
Intended Algorithms	
external detection	70%
intrinsic detection	30%
Plagiarism per Document	
hardly (5%-20%)	45%
medium (20%-50%)	15%
much (50%-80%)	25%
entirely (>80%)	15%
Document Length	
short (1-10 pp.)	50%
medium (10-100 pp.)	35%
long (100-1000 pp.)	15%

Plagiarism Case Statistics	
Topic Match	
intra-topic cases	50%
inter-topic cases	50%
Obfuscation	
none	40%
artificial	
– low obfuscation	20%
– high obfuscation	20%
simulated (AMT)	6%
translated ({de,es} to en)	14%
Case Length	
short (50-150 words)	34%
medium (300-500 words)	33%
long (3000-5000 words)	33%

The documents without plagiarism allow to determine whether or not a detector can distinguish plagiarism cases from overlaps that occur naturally between random documents.

The corpus is split into two parts, corresponding to the two paradigms of plagiarism detection, namely external plagiarism detection and intrinsic plagiarism detection. Note that in the case of intrinsic plagiarism detection the source documents used to generate the plagiarism cases are omitted: intrinsic detection algorithms are expected to detect plagiarism in a suspicious document by analyzing the document in isolation. Moreover, the intrinsic plagiarism cases are not obfuscated in order to preserve the writing style of the original author; the 40% of unobfuscated plagiarism cases in the corpus include the 30% of the cases belonging to the intrinsic part.

The fraction of plagiarism per document, the lengths of the documents and plagiarism cases, and the degree of obfuscation per case determine the difficulty of the cases: the corpus contains short documents with a short, unobfuscated plagiarism case, resulting in a 5% fraction of plagiarism, but it also contains large documents with several obfuscated plagiarism cases of varying lengths, drawn from different source documents and resulting in fractions of plagiarism up to 100%. Since the true distributions of these parameters in real plagiarism are unknown, sensible

estimations were made for the corpus. E.g., there are more simple plagiarism cases than complex ones, where "simple" refers to short cases, hardly plagiarism per document, and less obfuscation.

Finally, plagiarism cases were generated between topically related documents and between unrelated documents. To this end, the source documents and the suspicious documents were clustered into $k = 30$ clusters using bisecting k-means [22]. Then an equal share of plagiarism cases were generated for pairs of source documents and suspicious documents within as well as between clusters. Presuming the clusters correspond to (broad) topics, we thus obtained intra-topic plagiarism and inter-topic plagiarism.

4 Corpus Validation

This section reports on validation results about the "quality" of the plagiarism cases created for our corpus. We compare both artificial plagiarism cases and simulated plagiarism cases to cases of the two corpora Clough09 and METER. Presuming that the authors of these corpora put their best efforts into case construction and annotation, the comparison gives insights whether our scale-up strategies are reasonable in terms of case quality. To foreclose the results, we observe that simulated plagiarism and, in particular, artificial plagiarism behave similar to the two handmade corpora. In the light of the employed strategies to construct plagiarism this result may or may not be surprising—however, we argue that it is necessary to run such a comparison in order to provide a broadly accepted evaluation framework in this sensitive area.

The experimental setup is as follows: given a plagiarism case $s = \langle s_{\text{plg}}, d_{\text{plg}}, s_{\text{src}}, d_{\text{src}} \rangle$, the plagiarized passage s_{plg} is compared to the source passage s_{src} using 10 different retrieval models. Each model is an n-gram vector space model (VSM) where n ranges from 1 to 10 words, employing stemming, stop word removal, tf-weighting, and the cosine similarity. Similarity values are computed for all cases found in each corpus, but since the corpora are of different sizes, 100 similarities are sampled from each corpus to ensure comparability.

The rationale of this setup is as follows: a well-known fact from near-duplicate detection is that if two documents share only a few 8-grams—so-called shingles—it is highly probable that they are duplicates [5]. Another well-known fact is that two documents which are longer than a few sentences and which are exactly about the same topic will, with a high probability, share a considerable portion of their vocabulary. I.e., they have a high

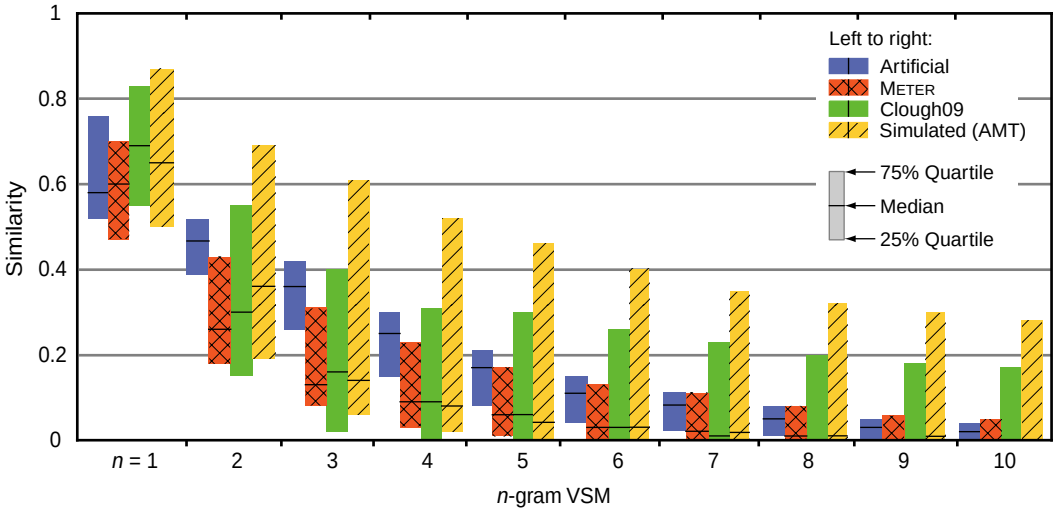


Figure 1: Comparison of four corpora of text reuse and plagiarism: each box plot shows the middle range of the measured similarities when comparing source passages to their rewritten versions. Basis is an n-gram VSM, where $n \in \{1, 2, \ldots, 10\}$ words.

similarity under a 1-gram VSM. It follows for plagiarism detection that a common shingle between s_{plg} and s_{src} pinpoints very accurately an unobfuscated portion of s_{plg}, while it is inevitable that even a highly obfuscated s_{plg} will share a portion of its vocabulary with s_{src}. The same holds for all other kinds of text reuse.

Figure 1 shows the obtained similarities, contrasting each n-gram VSM and each corpus. The box plots show the middle 50% of the respective similarity distributions as well as median similarities. The corpora divide into groups with comparable behavior: in terms of the similarity ranges covered, the artificial plagiarism compares to the METER corpus, except for $n \in \{2, 3\}$, while the simulated plagiarism from the Clough09 corpus behaves like that from our corpus, but with a different amplitude. In terms of median similarity, METER, Clough09, and our simulated plagiarism behave almost identical, while the artificial plagiarism differs. Also note that our simulated plagiarism as well as the Clough09 corpus contain some cases which are hardly obfuscated.

We interpret these results as follows: (1) Different kinds of plagiarism and text reuse do not differ very much under n-gram models. (2) Artificial plagiarism, if carefully generated, is a viable alternative to simulated plagiarism cases and real text reuse cases. (3) Our strategies to scale-up the construction of plagiarism corpora works well compared to existing, handmade corpora.

5 Summary

Current evaluation methodologies in the field of plagiarism detection research have conceptual shortcomings and allow only for a limited comparability. Our research contributes right here: we present tailored performance measures for plagiarism detection and the large-scale corpus PAN-PC-10 for the controlled evaluation of detection algorithms. The corpus features various kinds of plagiarism cases, including obfuscated cases that have been generated automatically and manually. An evaluation of the corpus in relation to previous corpora reveals a high degree of maturity. Until now, 31 plagiarism detectors have been compared using our evaluation framework. This high number of systems has been achieved based on two benchmarking workshops in which the framework was employed and developed, namely PAN'09 [15] and PAN'10 [16]. We hope that our framework will be beneficial as a challenging and yet realistic test bed for researchers in order to pinpoint the room for the development of better plagiarism detection systems.

Acknowledgements

We thank Andreas Eiselt for his devoted work on the corpus over the past two years. This work is partially funded by CONACYT-Mexico and the MICINN project TEXT-ENTERPRISE 2.0 TIN2009-13391-C04-03 (Plan I+D+i).

Bibliography

[1] Omar Alonso and Stefano Mizzaro. Can We Get Rid of TREC Assessors? Using Mechanical Turk for Relevance Assessment. In *SIGIR'09: Proceedings of the Workshop on The Future of IR Evaluation*, 2009.

[2] Vamshi Ambati, Stephan Vogel, and Jaime Carbonell. Active learning and crowd-sourcing for machine translation. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis, Mike Rosner, and Daniel Tapias, editors, *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, Valletta, Malta, may 2010. European Language Resources Association (ELRA). ISBN 2-9517408-6-7.

[3] Jeff Barr and Luis Felipe Cabrera. AI Gets a Brain. *Queue*, 4(4):24–29, 2006. ISSN 1542-7730. doi: 10.1145/1142055.1142067.

[4] Regina Barzilay and Lillian Lee. Learning to Paraphrase: An Unsupervised Approach Using Multiple-Sequence Alignment. In *NAACL'03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 16–23, Morristown, NJ, USA, 2003. Association for Computational Linguistics. doi: 10.3115/1073445.1073448.

[5] Andrei Z. Broder. Identifying and Filtering Near-Duplicate Documents. In *COM'00: Proceedings of the 11th Annual Symposium on Combinatorial Pattern Matching*, pages

1–10, London, UK, 2000. Springer-Verlag. ISBN 3-540-67633-3.

[6] Manuel Cebrian, Manuel Alfonseca, and Alfonso Ortega. Towards the Validation of Plagiarism Detection Tools by Means of Grammar Evolution. *IEEE Transactions on Evolutionary Computation*, 13(3):477–485, June 2009. ISSN 1089-778X.

[7] Paul Clough. Plagiarism in Natural and Programming Languages: An Overview of Current Tools and Technologies. Internal Report CS-00-05, University of Sheffield, 2000.

[8] Paul Clough. Old and New Challenges in Automatic Plagiarism Detection. National UK Plagiarism Advisory Service, http://ir.shef.ac.uk/cloughie/papers/pas_plagiarism.pdf, 2003.

[9] Paul Clough and Mark Stevenson. Creating a Corpus of Plagiarised Academic Texts. In *Proceedings of Corpus Linguistics Conference, CL'09 (to appear)*, 2009.

[10] Paul Clough and Mark Stevenson. Developing A Corpus of Plagiarised Short Answers. *Language Resources and Evaluation: Special Issue on Plagiarism and Authorship Analysis (in press)*, 2010.

[11] Paul Clough, Robert Gaizauskas, and S. L. Piao. Building and Annotating a Corpus for the Study of Journalistic Text Reuse. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC-02)*, pages 1678–1691, 2002.

[12] Wiebe Hordijk, María L. Ponisio, and Roel Wieringa. Structured Review of Code Clone Literature. Technical Report TR-CTIT-08-33, Centre for Telematics and Information Technology, University of Twente, Enschede, 2008.

[13] Winter Mason and Duncan J. Watts. Financial Incentives and the "Performance of Crowds". In *HCOMP'09: Proceedings of the ACM SIGKDD Workshop on Human Computation*, pages 77–85, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-672-4. doi: 10.1145/1600150.1600175.

[14] Hermann Maurer, Frank Kappe, and Bilal Zaka. Plagiarism - A Survey. *Journal of Universal Computer Science*, 12(8): 1050–1084, 2006.

[15] Martin Potthast, Benno Stein, Andreas Eiselt, Alberto Barrón-Cedeño, and Paolo Rosso. Overview of the 1st International Competition on Plagiarism Detection. In Benno Stein, Paolo Rosso, Efstathios Stamatatos, Moshe Koppel, and Eneko Agirre, editors, *SEPLN 2009 Workshop on Uncovering Plagiarism, Authorship, and Social Software Misuse (PAN 09)*, pages 1–9. CEUR-WS.org, September 2009. URL `http://ceur-ws.org/Vol-502`.

[16] Martin Potthast, Benno Stein, Andreas Eiselt, Alberto Barrón-Cedeño, and Paolo Rosso. Overview of the 2nd International Benchmarking Workshop on Plagiarism Detection. In Benno Stein, Paolo Rosso, Efstathios Stamatatos, and Moshe Koppel, editors, *Proceedings of PAN at CLEF 2010: Uncovering Plagiarism, Authorship, and Social Software Misuse*, September 2010.

[17] Chanchal K. Roy and James R. Cordy. Scenario-Based Comparison of Clone Detection Techniques. In *ICPC '08: Proceedings of the 2008 The 16th IEEE International Conference on Program Comprehension*, pages 153–162, Washington, DC, USA, 2008. IEEE Computer Society. ISBN 978-0-7695-3176-2.

[18] Chanchal K. Roy and James R. Cordy. Towards a Mutation-based Automatic Framework for Evaluating Code Clone Detection Tools. In *C3S2E '08: Proceedings of the 2008 C3S2E conference*, pages 137–140, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-101-9.

[19] Chanchal K. Roy, James R. Cordy, and Rainer Koschke. Comparison and Evaluation of Code Clone Detection Techniques and Tools: A Qualitative Approach. *Sci. Comput. Program.*, 74(7): 470–495, 2009. ISSN 0167-6423.

[20] Chanchal K. Roy and James R. Cordy. A survey on software clone detection research. Technical Report 2007-541, School of Computing, Queen's University at Kingston, Ontario, Canada, 2007.

[21] Geoffrey R. Whale. Identification of Program Similarity in Large Populations. *The Computer Journal*, 33(2):140–146, 1990. doi: 10.1093/comjnl/33.2.140.

[22] Ying Zhao, George Karypis, and Usama Fayyad. Hierarchical Clustering Algorithms for Document Datasets. *Data Min. Knowl. Discov.*, 10(2):141–168, 2005. ISSN 1384-5810. doi: 10.1007/s10618-005-0361-3.

Expressing OWL axioms by English sentences: dubious in theory, feasible in practice

Richard Power
Department of Computing
Open University
r.power@open.ac.uk

Allan Third
Department of Computing
Open University
a.third@open.ac.uk

Abstract

With OWL (Web Ontology Language) established as a standard for encoding ontologies on the Semantic Web, interest has begun to focus on the task of verbalising OWL code in controlled English (or other natural language). Current approaches to this task assume that axioms in OWL can be mapped to sentences in English. We examine three potential problems with this approach (concerning logical sophistication, information structure, and size), and show that although these could in theory lead to insuperable difficulties, in practice they seldom arise, because ontology developers use OWL in ways that favour a transparent mapping. This result is evidenced by an analysis of patterns from a corpus of over 600,000 axioms in about 200 ontologies.

1 Introduction

Since the adoption of OWL (Web Ontology Language) as a standard in 2004, several research groups have explored ways of mapping between OWL and controlled English, with the aim of presenting ontologies (both for viewing and editing) in natural language (Schwitter and Tilbrook, 2004; Kaljurand and Fuchs, 2007; Funk et al., 2007; Hart et al., 2008); this task has been called ontology 'verbalisation' (Smart, 2008). To develop generic methods for ontology verbalisation, some kind of structural mapping is needed between the formal and natural languages, and the assumption generally adopted has been a three-tier model in which identifiers for atomic terms (e.g., individuals, classes, properties) map to lexical entries, single axioms map to sentences, and groups of related axioms map to higher textual units such as paragraphs and sections. The purpose of this paper is to look in detail at one level of this model, the realisation of axioms by sentences, and to check its feasibility through an analysis of a large corpus of ontologies.

The input to a verbaliser is a file in one of the standard formats such as OWL/RDF or OWL/XML, containing axioms along with supporting statements such as annotations. As examples of the nature of the input, table 1 shows three axioms in OWL/XML format; without any attempt at aggregation or pronominalisation, they could be realised by the following sentences[1]:

> Horatio Nelson is an admiral.
>
> Horatio Nelson is the victor of the Battle of Trafalgar.
>
> Every admiral is commander of a fleet.

Without attempting anything like a full description of OWL, it will be useful to look more closely at the structure of these expressions. Note first that they are essentially in functor-argument form[2]. In the first axiom, for example, there is a functor called *ClassAssertion* with two arguments, one a class and the other an individual; the meaning of the axiom is that the individual belongs to the class. The second functor (*ObjectPropertyAssertion*) requires instead three arguments,

[1] Note that one limitation of OWL is that at present it contains no treatment of time; we therefore have to fall back on the historical present.

[2] In fact, there is an alternative format called OWL Functional Syntax in which, for example, the first axiom would be represented by a predication of the form `ClassAssertion(X,Y)`.

Coling 2008: Poster Volume, pages 1006–1013,
Beijing, August 2010

```
<ClassAssertion>
  <Class IRI="http://www.example.org#admiral"/>
  <NamedIndividual IRI="www.example.org#HoratioNelson"/>
</ClassAssertion>

<ObjectPropertyAssertion>
  <ObjectProperty IRI="http://www.example.org#victorOf"/>
  <NamedIndividual IRI="http://www.example.org#HoratioNelson"/>
  <NamedIndividual IRI="http://www.example.org#BattleOfTrafalgar"/>
</ObjectPropertyAssertion>

<SubClassOf>
  <Class IRI="http://www.example.org#admiral"/>
  <ObjectSomeValuesFrom>
    <ObjectProperty IRI="http://www.example.org#commanderOf"/>
    <Class IRI="http://www.example.org#fleet"/>
  </ObjectSomeValuesFrom>
</SubClassOf>
```

Table 1: Examples of axioms in OWL/XML

and describes a relation (in OWL these are called 'properties') holding between two individuals; the third (*SubClassOf*) requires two arguments, both classes, and asserts that the first class is a subclass of the second.

Turning to the structure of the arguments, there are two possibilities: either the argument is *atomic*, in which case it will be represented by an identifier (or a literal if it is a data value), or it is *complex*, in which case it will be represented by an OWL functor with arguments of its own. Most of the arguments in table 1 are atomic, the sole exception being the second argument of *SubClassOf*, which denotes a complex class meaning 'someone that is commander of a fleet'[3]. In general, then, the OWL functors denote *logical* concepts such as class membership and class inclusion, while atomic terms denote domain-specific concepts such as *Nelson* and *admiral*. A fundamental design decision of the Semantic Web is that logical concepts are standardised, while domain concepts are left open: ontology developers are free to name the class *admiral* in any way they please, provided that the identifier takes the form of an IRI (Internationalized Resource Identifier).

Given this distinction, the obvious strategy to follow in developing a verbaliser is to divide linguistic resources into two parts: (a) a generic set of rules for realising logical expressions (based on standardised OWL functors); (b) a domain-specific lexicon for realising atomic individuals, classes and properties. This obviously raises the problem of how to acquire the specialised lexicons needed for each ontology. All else failing, these would have to be crafted by hand, but provided that we are not too concerned about text quality, a provisional lexicon can often be derived automatically from internal evidence within the ontology (i.e., either from identifier names or annotation labels)[4].

Assuming that a lexicon for atomic terms can be obtained (by fair means or foul), there remains a question of whether we can find sentence patterns which provide understandable realisations of the logical patterns determined by (possibly nested) OWL functors. In section 2 we show that this is not guaranteed, for three reasons. First, there may be OWL functors that represent *logically sophisticated* concepts which cannot be expressed in non-technical English. Secondly, an OWL axiom may be hard to verbalise because it lacks the right kind of *information structure* (i.e., because it fails to make a statement about a recognisable topic such as an individual or atomic class). Finally, since arguments can be nested indefinitely, an axiom might contain so much *se-*

[3]To be more precise we should say 'someone that is commander of one or more fleets'; this kind of trade-off between elegance and precision often arises in systems that verbalise formal languages.

[4]We have discussed elsewhere whether phrases derived in this way provide suitable lexicalisations (Power, 2010), but this topic lies outside the scope of the present paper.

mantic complexity that it cannot be compressed clearly into a single sentence. We then describe (section 3) an empirical analysis of axiom patterns from about 200 ontologies, which investigates whether these potential problems are common in practice. Section 4 discusses the results, and section 5 concludes.

2 Potential problems in verbalising axioms

2.1 Logical sophistication

We show in table 2 the 16 most commonly used OWL functors for expressing axioms, each accompanied by a simple English sentence illustrating what the functor means. As will be seen, the functors divide into two groups. For those in the upper segment, it is relatively easy to find English constructions that realise the logical content of the axiom — assuming we have suitable lexicalisations of the atomic terms. For those in the lower segment, finding a good English realisation is harder, since statements describing properties are normally found only in the rarified worlds of mathematics and logic, not in everyday discourse. Our attempts to verbalise these axioms are accordingly clumsy (e.g., through resorting to variables like X and Y), and not even entirely precise (e.g., the sentence for *FunctionalObjectProperty* should really specify 'For any X...'); perhaps the reader can do better.

Does this mean that our aim of realising OWL axioms in non-technical English is doomed? We would argue that this depends on how the axioms describing properties are used in practice. First, for any difficult axiom functor, it is important to consider its frequency. If it turns out that a functor accounts for (say) only one axiom in every thousand, then it will give rise only to the occasional clumsy sentence, not a text that is clumsy through and through. Second, it is important to take account of argument complexity. If a functor is used invariably with atomic terms as arguments, then the sentence expressing it will contain only one source of complexity — logical sophistication; if instead the functor has non-atomic arguments, this additional strain might push it over a threshold from difficult to incomprehensible. Fortunately, OWL syntax requires that all property arguments for the difficult functors are atomic — for *FunctionalObjectProperty*, for instance, the argument cannot be a complex property expression. For statements about domains and ranges, however, class arguments can be non-atomic, so here a complexity issue might arise.

2.2 Information structure

We learn at school that sentences have a subject (preferably simple) and predicate (relatively complex), the purpose of the predicate being to say something about the subject. This rather simplified idea is developed technically in work on information structure (Kruijff-Korbayová and Steedman, 2003) and centering theory (Walker et al., 1998). Is there any equivalent to this topic-comment distinction in OWL? Formally speaking, one would have to answer in the negative. The two-argument functor *SubClassOf*, for example, can have class expressions of any complexity in either argument position, and there is no logical reason to claim that it is 'about' one of these classes rather than the other. This is still clearer in the case of *EquivalentClasses*, where the functor is commutative (so that switching the arguments leaves the meaning unchanged). Again there seems to be a difficulty here — and again we argue that this difficulty might disappear, or at least diminish, if we consider how OWL is used in practice.

Suppose, for instance, that although OWL syntax allows indefinitely complex arguments in either position for the *SubClassOf* functor, in practice users invariably construct axioms in which the first argument is an atomic term, with complex expressions occurring (if at all) only in second-argument position. This would strongly suggest, in our view, that developers are assigning a topic-comment structure to the two arguments, with the first expressing the topic and the second expressing the comment. As we will show later in the paper, this pattern is found overwhelmingly — so much so that in a sample of nearly half a million *SubClassOf* axioms, fewer than 1000 instances (0.2%) were found of non-atomic first arguments.

Functor	Example
SubClassOf	Every admiral is a sailor
EquivalentClasses	An admiral is defined as a person that commands a fleet
DisjointClasses	No sailor is a landlubber
ClassAssertion	Nelson is an admiral
ObjectPropertyAssertion	Nelson is victor of the Battle of Trafalgar
DataPropertyAssertion	The Battle of Trafalgar is dated 1805
ObjectPropertyDomain	If X commands Y, X must be a person
ObjectPropertyRange	If X commands Y, Y must be a fleet
SubObjectPropertyOf	If X is a child of Y, X must be related to Y
InverseObjectProperties	If X is a child of Y, Y must be a parent of X
TransitiveObjectProperty	If X contains Y and Y contains Z, X must contain Z
FunctionalObjectProperty	There can be only one Y such that X has as father Y
DataPropertyDomain	If X is dated Y, X must be an event
DataPropertyRange	If X is dated Y, Y must be an integer
SubDataPropertyOf	If X occurs during Y, X must be dated Y
FunctionalDataProperty	There can be only one Y such that X is dated Y

Table 2: Meanings of OWL functors

2.3 Semantic complexity

When encoding knowledge in description logic, developers have considerable freedom in distributing content among axioms, so that axiom size is partly a matter of style — rather like sentence length in composing a text. Development tools like Protégé (Rector et al., 2004) support *refactoring* of axioms, so that for example any axiom of the form $C_A \sqsubseteq C_S \sqcap C_L$ (e.g., 'Every admiral is a sailor and a leader') can be split into two axioms $C_A \sqsubseteq C_S$ and $C_A \sqsubseteq C_L$ ('Every admiral is a sailor. Every admiral is a leader.'), or vice-versa[5]. Indeed, it can be shown that *any* set of *SubClassOf* axioms can be amalgamated into a single axiom (Horrocks, 1997) of the form $\top \sqsubseteq M$, where $\top$ is the class containing all individuals in the domain, and M is a class to which any individual respecting the axiom set must belong[6]. Applying this transformation to just two axioms already yields an amalgam that will perplex most readers:

> Every admiral is a sailor
> Every admiral commands a fleet.
>
> Everything is (a) either a non-admiral or a sailor, and (b) either a non-admiral or something that commands a fleet.

There is thus no guarantee that an axiom in OWL can be verbalised transparently by a single sentence; in theory it could contain as much knowledge as a textbook. As before, we have to appeal to practice. Do ontology developers distribute content among knowledge units (axioms) equivalent in size to sentences? If they (almost always) do, then our approach is worth pursuing; if not, we have to reconsider.

3 Method

To investigate the issues of usage just described, we have analysed axiom patterns in a large corpus of ontologies of varying subject-matter and provenance. The corpus was based on the TONES Ontology Repository (TONES, 2010), which is a searchable database of RDF/XML ontologies from a range of sources. The repository is intended to be useful to developers of tools to work with ontologies, and as such represents a wide range of ontology kinds and features. It also classifies ontologies by 'expressivity' — the weakest description logic necessary to express every axiom. While the TONES site itself acknowledges that the expressivity categorisation is only a guideline, it can serve as a rough guide for comparison with the pattern frequency analysis carried out here.

The whole repository was downloaded, comprising 214 files each containing between 0 and 100726 logical axioms[7]. (Note that an OWL

[5]The symbols $\sqsubseteq$ and $\sqcap$ in logical notation correspond to the OWL functors *SubClassOf* and *ObjectIntersectionOf*.

[6]This all-embracing axiom or 'meta-constraint' is computed by the standard description logic reasoning algorithms when determining the consistency of a knowledge base.

[7]A few of the ontologies in the TONES repository were excluded, either because of syntax errors in the original files (2-3 files), or because they exceeded our processing limits —

file may contain no logical axioms and still be non-empty.) To develop quickly a program that could cope with the larger ontologies without memory problems, we used the Java-based OWL API (Horridge and Bechhofer, 2010) as much as possible, in conjunction with standard Unix text-processing tools ('grep', 'sed' and 'awk' (Dougherty and Robbins, 1997)) for pattern recognition[8].

Each ontology was converted into OWL Functional Syntax (Motik et al., 2010) and lists were automatically generated of the identifiers it contains — classes, named individuals, properties, and so on. The Unix tools were scripted to replace every occurrence of such an identifier with a string representing its type. This process generated a new file in which every axiom of the original ontology had been replaced with a string representing its logical structure: thus *SubClassOf(Admiral, Sailor)* and *SubClassOf(Sailor, Person)* would each have been replaced with *SubClassOf(Class, Class)*. The number of occurrences of each unique pattern was then counted and the results converted into a set of Prolog facts for further analysis. Some manual tidying-up of the data was necessary in order to correct some complex cases such as quoted string literals which themselves contained (escaped) quoted strings; however, these cases were so rare that any remaining errors should not adversely affect output quality.

4 Results

To address the issue of *logical sophistication*, we first calculated frequencies for each axiom functor, using two measures: (a) the number of ontologies in which the functor was used at least once, and (b) the number of axioms using the functor overall. The former measure (which we will call 'ontology frequency') is a useful corrective since a simple axiom count can be misleading when a functor is used profusely in a few very large ontologies, but rarely elsewhere. The results are presented in table 3, ordered by ontology frequency rather than overall axiom frequency[9]. As can be seen, the ten functors classified as logically sophisticated in table 2 are relatively rare, by both measures, accounting overall for just 2.2% of the axioms in the corpus, with none of them having a frequency reaching even 5 in 1000.

Next, to address *information structure*, we looked at the argument patterns for each axiom functor, distinguishing three cases: (a) all arguments simple (i.e., atomic); (b) all arguments complex (non-atomic); (c) mixed arguments (some atomic, some non-atomic). This comparison is relevant only for the functors *SubClassOf*, *EquivalentClasses* and *DisjointClasses*, for which OWL syntax allows multiple non-atomic arguments. The results (table 4) show a clear preference for patterns in which at least one argument is simple. Thus for *SubClassOf*, given the overall frequencies of simple and complex arguments for this functor, the expected frequency for the combination Complex-Complex would be 12606 (2.7%), whereas the observed frequency was only 978 (0.2%) (χ^2 = 16296 with df=2, $p < 0.0001$)[10]. The corresponding result for *EquivalentClasses* is even clearer, with not a single instance of an axiom in which all arguments are complex, against an expected frequency of 973 (16.0%) (χ^2 = 2692 with df=2, $p < 0.0001$)[11]. For *DisjointClasses* no complex arguments were obtained, so the only possible combination was 'All Simple'. Overall, 99.8% of axioms for these three functors contained at least one atomic term, suggesting that the arguments were interpreted according to intuitions of information structure, with one atomic argument serving as the topic. This point is reinforced by our next analysis, which considers detailed argument patterns.

e.g., the Foundational Model of Anatomy (Rosse and Mejino, 2003).

[8] A pure Java solution was not practical in the time available since the OWL API was designed to support reasoning and evaluation of OWL ontologies rather than syntactic analysis of their axioms. We hope to produce an extension of the OWL API to support straightforward and portable analysis of ontologies in the future.

[9] Note that the total in the first column of table 3 is simple the number of ontologies in our sample; the sum of the frequencies in the column is of no interest at all.

[10] The data for this test, with expected values in brackets, are SS = 297293 (312138), CC = 978 (12606), and SC = 170541 (144068), where S means 'Simple' and C means 'Complex'.

[11] The data for this test, with expected values in brackets, are SS = 1222 (2190), CC = 0 (973), and SC = 4860 (2919), where again S means 'Simple' and C means 'Complex'.

Functor	Ontology Frequency	Percent	Axiom Frequency	Percent
SubClassOf	190	94%	468812	74.0%
EquivalentClasses	94	46%	6082	1.0%
ObjectPropertyRange	92	45%	2275	0.4%
ObjectPropertyDomain	91	45%	2176	0.3%
DisjointClasses	88	43%	94390	14.9%
SubObjectPropertyOf	75	37%	2511	0.4%
InverseObjectProperties	63	31%	1330	0.2%
TransitiveObjectProperty	59	29%	221	0.0%
FunctionalObjectProperty	56	28%	1129	0.2%
DataPropertyRange	52	26%	2067	0.3%
ClassAssertion	49	24%	12798	2.0%
DataPropertyDomain	47	23%	2019	0.3%
FunctionalDataProperty	37	18%	931	0.1%
ObjectPropertyAssertion	22	11%	19524	3.1%
DataPropertyAssertion	14	7%	17488	2.8%
SubDataPropertyOf	6	3%	12	0.0%
TOTAL	203	100%	633791	100%

Table 3: Frequencies for OWL functors

Functor	All Simple	Percent	All Complex	Mixed	Percent
SubClassOf	297293	63%	978 (0.2%)	170541	37%
EquivalentClasses	1222	20%	0	4860	80%
DisjointClasses	94390	100%	0	0	0%
TOTAL	392905	69%	978 (0.2%)	175401	31%

Table 4: Simple and complex arguments of OWL functors

OWL Pattern	Frequency	Percent
SubClassOf(Class,Class)	297293	46.9%
SubClassOf(Class,ObjectSomeValuesFrom(ObjectProperty,Class))	158519	25.0%
DisjointClasses(Class,Class)	94358	14.9%
ObjectPropertyAssertion(ObjectProperty,NamedIndividual,NamedIndividual)	18552	3.0%
DataPropertyAssertion(DataProperty,NamedIndividual,Literal)	17433	2.7%
ClassAssertion(Class,NamedIndividual)	12767	2.0%
SubClassOf(Class,ObjectAllValuesFrom(ObjectProperty,Class))	4990	0.8%
SubObjectPropertyOf(ObjectProperty,ObjectProperty)	2453	0.4%
EquivalentClasses(Class,ObjectIntersectionOf(Class,ObjectSomeValuesFrom(ObjectProperty,Class)))	2217	0.3%
ObjectPropertyRange(ObjectProperty,Class)	2025	0.3%
ObjectPropertyDomain(ObjectProperty,Class)	1835	0.3%
DataPropertyDomain(DataProperty,Class)	1703	0.3%
SubClassOf(Class,ObjectHasValue(ObjectProperty,NamedIndividual))	1525	0.2%
SubClassOf(Class,DataHasValue(DataProperty,Literal))	1473	0.2%
InverseObjectProperties(ObjectProperty,ObjectProperty)	1318	0.2%
DataPropertyRange(DataProperty,Datatype)	1308	0.2%
EquivalentClasses(Class,Class)	1222	0.2%
FunctionalObjectProperty(ObjectProperty)	1121	0.2%
Other pattern...	11469	1.8%
TOTAL	633791	100%

Table 5: Frequencies for OWL Functor-Argument patterns

Finally, to address *semantic complexity* (i.e., axiom size), we counted the frequencies of detailed argument patterns, abstracting from atomic terms as explained in section 3. The results (ordered by pattern frequency) are presented in table 5, which reveals several clear trends:

- A small number of patterns covers most of the axioms in the corpus. Thus the top five patterns cover 91.9% of the axioms, the top 10 cover 95.8%, and the top 20 cover 97.2%.
- All of the frequent patterns (i.e., the top 20) can be expressed by a single sentence without problems of semantic complexity arising from size. The most complex is the *EquivalentClasses* pattern (number 10 in the list), but this can be realised comfortably by a sentence following the classical Aristotelian pattern for a definition — e.g., 'An admiral is defined as a person that commands a fleet'.
- None of the first ten patterns employs the axiom functors previously classified as logically sophisticated (bottom half of table 2).
- In the patterns where one argument is simple and the other is complex (i.e., *SubClassOf* and *EquivalentClasses*), the simple argument invariably comes first, supporting the intuition that developers conceptualise these statements in subject-predicate form, with (simple) topic preceding (possibly complex) comment.
- Among the frequent patterns, different functors have distinctive argument preferences. For instance, for *SubClassOf* most axioms have atomic arguments, presumably because it is through this functor that the class hierarchy is specified. For *EquivalentClasses*, instead, the Aristotelean definition pattern is by far the most frequent, although all-atomic arguments are occasionally employed (0.2% of axioms) to show that two class terms are synonymous.

5 Conclusion

Our analysis of over 600,000 axioms from 203 ontologies provides empirical support for the assumption that in practice OWL axioms can be transparently expressed by English sentences. In principle, as we have seen, OWL syntax grants users the freedom to construct axioms that would defeat this assumption entirely, either by concentrating too much semantic content into a single axiom, or by filling all argument positions by complex expressions that are unsuited to fulfilling the role of topic; it also allows logically sophisticated statements about properties, which would lead to impossibly clumsy texts if they occurred too often, or were exacerbated by complex arguments. In practice, if our sample is typical, none of these problems seems to arise, and we think it would be a fair summary of our results to say that ontology developers treat OWL axioms by analogy with sentences, by assigning a clear information structure (so that one atomic argument is identified with the topic) and including only an appropriate amount of content.

Having identified a relatively small set of common axiom patterns, it is obviously interesting to consider how each pattern can best be expressed in a given natural language. Considering the pattern *SubClassOf(Class,Class)* for instance (47% of all axioms), one could weigh the relative merits of 'Every admiral is a sailor', 'All admirals are sailors', 'Admirals are sailors', 'If X is an admiral, then X must be a sailor', and so forth. To address this issue we are planning a quite different kind of empirical study on how various sentence patterns are interpreted by human readers; by highlighting the logical patterns that occur most often in practice, the results reported here will help set the parameters for such an investigation.

Acknowledgments

The research described in this paper was undertaken as part of the SWAT project (Semantic Web Authoring Tool), which is supported by the UK Engineering and Physical Sciences Research Council (EPSRC) grants G033579/1 (Open University) and G032459/1 (University of Manchester). We thank the anonymous reviewers and our colleagues on the SWAT project for their comments.

References

Dougherty, Dale and Arnold Robbins. 1997. *sed and awk*. UNIX Power Tools. O'Reilly Media, 2nd edition.

Funk, Adam, Valentin Tablan, Kalina Bontcheva, Hamish Cunningham, Brian Davis, and Siegfried Handschuh. 2007. CLOnE: Controlled Language for Ontology Editing. In *6th International and 2nd Asian Semantic Web Conference (ISWC2007+ASWC2007)*, pages 141–154, November.

Hart, Glen, Martina Johnson, and Catherine Dolbear. 2008. Rabbit: Developing a control natural language for authoring ontologies. In *ESWC*, pages 348–360.

Horridge, Matthew and Sean Bechhofer. 2010. The OWL API. http://owlapi.sourceforge.net. Last accessed: 21st April 2010.

Horrocks, Ian. 1997. *Optimising Tableaux Decision Procedures for Description Logics*. Ph.D. thesis, University of Manchester.

Kaljurand, K. and N. Fuchs. 2007. Verbalizing OWL in Attempto Controlled English. In *Proceedings of OWL: Experiences and Directions*, Innsbruck, Austria.

Kruijff-Korbayová, Ivana and Mark Steedman. 2003. Discourse and information structure. *Journal of Logic, Language and Information*, 12(3):249–259.

Motik, Boris, Peter F. Patel-Schneider, and Bijan Parsia. 2010. OWL 2 web ontology language: Structural specification and functional-style syntax. http://www.w3.org/TR/owl2-syntax/. 21st April 2010.

Power, Richard. 2010. Complexity assumptions in ontology verbalisation. In *48th Annual Meeting of the Association for Computational Linguistics*.

Rector, Alan, Nick Drummond, Matthew Horridge, Jeremy Rogers, Holger Knublauch, Robert Stevens, Hai Wang, and Chris Wroe. 2004. OWL Pizzas: Practical Experience of Teaching OWL-DL: Common Errors and Common Patterns. In *14th International Conference on Knowledge Engineering and Knowledge Management*, pages 63–81.

Rosse, Cornelius and José L. V. Mejino. 2003. A reference ontology for biomedical informatics: the Foundational Model of Anatomy. *Journal of Biomedical Informatics*, 36(6):478–500.

Schwitter, R. and M. Tilbrook. 2004. Controlled natural language meets the semantic web. In *Proceedings of the Australasian Language Technology Workshop*, pages 55–62, Macquarie University.

Smart, Paul. 2008. Controlled Natural Languages and the Semantic Web. Technical Report Technical Report ITA/P12/SemWebCNL, School of Electronics and Computer Science, University of Southampton.

TONES. 2010. The TONES ontology repository. http://owl.cs.manchester.ac.uk/repository/browser. Last accessed: 21st April 2010.

Walker, M., A. Joshi, and E. Prince. 1998. *Centering theory in discourse*. Clarendon Press, Oxford.

Automatic Committed Belief Tagging

Vinodkumar Prabhakaran
Columbia University
vp2198@columbia.edu

Owen Rambow
Columbia University
rambow@ccls.columbia.edu

Mona Diab
Columbia University
mdiab@ccls.columbia.edu

Abstract

We go beyond simple propositional meaning extraction and present experiments in determining which propositions in text the author believes. We show that deep syntactic parsing helps for this task. Our best feature combination achieves an F-measure of 64%, a relative reduction in F-measure error of 21% over not using syntactic features.

1 Introduction

Recently, interest has grown in relating text to more abstract representations of its propositional meaning, as witnessed by work on semantic role labeling, word sense disambiguation, and textual entailment. However, there is more to "meaning" than just propositional content. Consider the following examples, and suppose we find these sentences in the *New York Times*:

(1) a. *GM will lay off workers.*

b. *A spokesman for GM said GM will lay off workers.*

c. *GM may lay off workers.*

d. *The politician claimed that GM will lay off workers.*

e. *Some wish GM would lay of workers.*

f. *Will GM lay off workers?*

g. *Many wonder if GM will lay off workers.*

If we are searching text to find out whether GM will lay off workers, all of the sentences above contain the proposition LAY-OFF(GM,WORKERS). However, they allow us very different inferences about whether GM will lay off workers or not. Supposing we consider the *Times* a trustworthy news source, we would be fairly certain if we read (1a) and (1b). (1c) suggests the *Times* is not certain about the layoffs, but considers them possible. When reading (1d), we know that someone else thinks that GM will lay off workers, but that the *Times* does not necessarily share this belief. (1e), (1f), and (1g) do not tell us anything about whether anyone believes whether GM will lay off workers.

In order to tease apart what is happening, we need to abandon a simple view of text as a repository of propositions about the world. We use two assumptions to aid us. The first assumption is that discourse participants model each other's cognitive state during discourse (we take the term to include the reading of monologic written text), and that language provides cues for the discourse participants to do the modeling. This assumption is commonly made, for example by Grice (1975) in his Maxim of Quantity. Following the literature in Artificial Intelligence (Bratman, 1999; Cohen and Levesque, 1990), we model cognitive state as beliefs, desires, and intentions. Crucially, these three dimensions are orthogonal; for example, we can desire something but not believe it.

(2) I know John won't be here, but I wouldn't mind if he were

However, we cannot both believe something and not believe it:

(3) #John won't be here, but nevertheless I think he may be here

Note that (2) requires *but* in order to be felicitous, but sentence (3) cannot be "saved" by any discourse markers – it is not interpretable. In this paper, we are interested in beliefs (and in distin-

Coling 2008: Poster Volume, pages 1014–1022,
Beijing, August 2010

guishing them from desires and intentions).

The second assumption is that communication is intention-driven, and understanding text actually means understanding the communicative intention of the writer. Furthermore, communicative intentions are intentions to affect the reader's cognitive state – his or her beliefs, desires, and/or intentions. This view has been adopted in the text generation and dialog community more than in the information extraction and text understanding communities (Mann and Thompson, 1987; Hovy, 1993; Moore, 1994; Bunt, 2000; Stone, 2004). In this paper we explore the following: we would like to recognize what the writer of the text intends the reader to believe about various people's beliefs about the world (including the writer's own). In this view, the result of text processing is not a list of facts about the world, but a list of facts about different people's cognitive states. In this paper, we limit ourselves to the writer's beliefs, but we specifically want to determine which propositions he or she intends us to believe he or she holds as beliefs, and with what strength. The result of such processing will be a much more fine-grained representation of the information contained in written text than has been available so far.

2 Belief Annotation and Data

We use a corpus of 10,000 words annotated for speaker belief of stated propositions (Diab et al., 2009). The corpus is very diverse in terms of genre, and it includes newswire text, email, instructions, and solicitations. The corpus annotates each verbal proposition (clause or small clause), by attaching one of the following tags to the head of the proposition (verbs and heads of nominal, adjectival, and prepositional predications).

• Committed belief (CB): the writer indicates in this utterance that he or she believes the proposition. For example, *GM has laid off workers*, or, even stronger, *We know that GM has laid off workers*. Committed belief can also include propositions about the future: people can have equally strong beliefs about the future as about the past, though in practice probably we have stronger beliefs about the past than about the future.

• Non-committed belief (NCB): the writer identifies the proposition as something which he or she could believe, but he or she happens not to have a strong belief in. There are two subcases. First, the writer makes clear that the belief is not strong, for example by using a modal auxiliary epistemically: *GM may lay off workers*. Second, in reported speech, the writer is not signaling to the reader what he or she believes about the reported speech: *The politician claimed that GM will lay off workers*. Again, the issue of tense is orthogonal.

• Not applicable (NA): for the writer, the proposition is not of the type in which he or she is expressing a belief, or could express a belief. Usually, this is because the proposition does not have a truth value in this world (be it in the past or in the future). This covers expressions of desire (*Some wish GM would lay of workers*), questions (*Will GM lay off workers?*), and expressions of requirements (*GM is required to lay off workers* or *Lay off workers!*).

All propositional heads are classified as one of the classes CB, NCB, or NA, and all other tokens are classified as O. Note that in this corpus, event nominals (such as *the lay-offs by GM were unexpected*) are, unfortunately, not annotated for belief and are always marked "O". Note also that the syntactic form does not determine the annotation, but the perceived writer's intention – a question will usually be an NA, but sometimes a question can be used to convey a belief (for example, a rhetorical question), in which case it would be labeled CB.

3 Automatic Belief Tagging

3.1 Approach

We applied a supervised learning framework to the problem of identifying committed belief in context. Our task consists of two conceptual subtasks: identifying the propositions, and classifying each proposition as CB, NCB, or NA. For the first subtask, we could use a system that cuts a sentence into propositions, but we are not aware of such a system that performs at an adequate level. Instead, we tag the heads of the proposition, which amounts to the same in the sense that there is a bijection between propositions and their heads. Practically, we have the choice between

No	Feature	Type	Description
Features that performed well			
1	isNumeric	L	Word is Alphabet or Numeric?
2	POS	L	Word's POS tag
3	verbType	L	Modal/Aux/Reg (= 'nil' if the word is not a verb)
4	whichModalAmI	L	If I am a modal, what am I? (= 'nil' if I am not a modal)
3	amVBwithDaughterTo	S	Am I a VB with a daughter *to*?
4	haveDaughterPerfect	S	Do I have a daughter which is one of *has, have, had*?
5	haveDaughterShould	S	Do I have a daughter *should*?
6	haveDaughterWh	S	Do I have a daughter who is one of *where, when, while, who, why*?
7	haveReportingAncestor	S	Am I a verb/predicate with an ancestor whose lemma is one of *tell, accuse, insist, seem, believe, say, find, conclude, claim, trust, think, suspect, doubt, suppose*?
8	parentPOS	S	What is my parent's POS tag?
9	whichAuxIsMyDaughter	S	If I have a daughter which is an auxiliary, what is it? (= 'nil' if I do not have an auxiliary daughter)
10	whichModalIsMyDaughter	S	If I have a daughter which is a modal, what is it? (= 'nil' if I do not have a modal daughter)
Features that were not useful			
1	Lemma	L	Word's Lemma
2	Stem	L	Word stem (Using Porter Stemmer)
3	Drole	S	Deep role (drole in MICA features)
4	isRoot	S	Is the word the root of the MICA Parse tree?
5	parentLemma	S	Parent word's Lemma
6	parentStem	S	Parent word stem (Using Porter Stemmer)
7	parentSupertag	S	Parent word's super tag (from Penn Treebank)
8	Pred	S	Is the word a predicate? (pred in MICA features)
9	wordSupertag	S	Word's Super Tag (from Penn Treebank)

Table 1: All Features Used

a joint model, in which the heads are chosen and classified simultaneously, and a pipeline model, in which heads are chosen first and then classified. In this paper, we consider the joint model in detail and in Section 3.5.3, we present results of the pipeline model; they support our choice.

In the joint model, we define a four-way classification task where each token is tagged as one of four classes – CB, NCB, NA, or O (nothing) – as defined in Section 2. For tagging, we experimented with Support Vector Machines (SVM) and Conditional Random Fields (CRF). For SVM, we used the YAMCHA(Kudo and Matsumoto, 2000) sequence labeling system,[1] which uses the TinySVM package for classification.[2] For CRF, we used the linear chain CRF implementation of the MALLET(McCallum, 2002) toolkit.[3]

3.2 Features

We divided our features into two types - Lexical and Syntactic. Lexical features are at the token level and can be extracted without any parsing with relatively high accuracy. We expect these features to be useful for our task. For example, isNumeric, which denotes whether the word is a number or alphabetic, is a lexical feature. Syntactic features of a token access its syntactic context in the dependency tree. For example, parentPOS, the POS tag of the parent word in the dependency parse tree, is a syntactic feature. We used the MICA deep dependency parser (Bangalore et al., 2009) for parsing in order to derive the syntactic features. We use MICA because we assume that the relevant information is the

[1] http://chasen.org/ taku/software/YAMCHA/

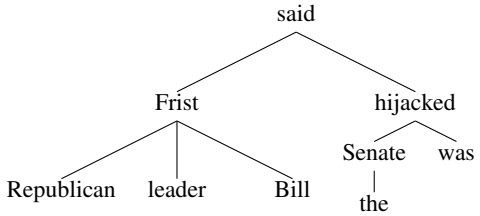
[2] http://chasen.org/ taku/software/TinySVM/

[3] http://MALLET.cs.umass.edu/

predicate-argument structure of the verbs, which is explicit in the MICA output. While it is clear that having a perfect parse would yield useful features, current parsers perform at levels of accuracy lower than that of part-of-speech taggers, so that it is not a foregone conclusion that using automatic parser output helps in our task.

The list of features we used in our experiments are summarized in Table 1. The column 'Type' denotes the type of the feature. 'L' stands for lexical features and 'S' stands for syntactic features.

The tree below shows the dependency parse tree output by MICA for the sentence *Republican leader Bill Frist said the Senate was hijacked*.



In the above sentence, *said* and *hijacked* are the propositions that should be tagged. Let's look at *hijacked* in detail. The feature haveReportingAncestor of *hijacked* is 'Y' because it is a verb with a parent verb *said*. Similarly, the feature haveDaughterAux would also be 'Y' because of daughter *was*, whereas whichAuxIsMyDaughter would get the value *was*.

We also considered several other features which did not yield good results. For example, the token's supertag (Bangalore and Joshi, 1999), the parent token's supertag, a binary feature isRoot (Is the word the root of the parse tree?) were deemed not useful. We list the features we experimented with and decided to discard in Table 1.

For finding the best performing features, we did an exhaustive search on the feature space, incrementally pruning away features that are not useful.

3.3 Experiments

This section describes different experiments we conducted in detail. It explains the experimental setup for both learning frameworks we used - YAMCHA and MALLET. We also explain the pipeline model in detail.

Class	Description
L_C	Lexical features with Context
$L_N S_N$	Lexical and Syntactic features with No-context
$L_C S_N$	Lexical features with Context and Syntactic features with No-context
$L_C S_C$	Lexical and Syntactic features with Context

Table 2: YAMCHA Experiment Sets

3.3.1 YAMCHA Experiments

We categorized our YAMCHA experiments into different experimental conditions as shown in Table 2. For each class, we did experiments with different feature sets and (linear) context widths. Here, context width denotes the window of tokens whose features are considered. For example, a context width of 2 means that the feature vector of any given token includes, in addition to its own features, those of 2 tokens before and after it as well as the tag prediction for 2 tokens before it. For $L_N S_N$, the context width of all features was set to 0. For $L_C S_N$, the context width of syntactic features alone was set to 0. A context width of 0 for a feature means that the feature vector includes that feature of the current token only. When context width was non-zero, we varied it from 1 to 5, and we report the results for the optimal context width.

We tuned the SVM parameters, and the best results were obtained using the *One versus All* method for multiclass classification on a quadratic kernel with a c value of 0.5. All results presented for YAMCHA here use this setting.

3.3.2 MALLET Experiments

Class	Description
L	Lexical features only
LS	Lexical and Syntactic features

Table 3: MALLET Experiment Sets

We categorized our MALLET experiments into two classes as shown in Table 3. We computed the features described in Section 3.2 at the token level and converted them to binary in order to use them for CRF. We experimented with varying orders and the best results were obtained for or-

Class	Feature Set	Parm	P	R	F
YAMCHA - Joint Model					
L_C	POS, whichModalAmI, verbType, isNumeric	CW=3	61.9	52.7	56.9
$L_N S_N$	POS, whichModalAmI, parentPOS, haveReportingAncestor, whichModalIsMyDaughter, haveDaughterPerfect, whichAuxIsMyDaughter, amVBwithDaughterTo, haveDaughterWh, haveDaughterShould	CW=0	62.5	57.5	59.9
$L_C S_N$	POS, whichModalAmI, parentPOS, haveReportingAncestor, whichModalIsMyDaughter, whichAuxIsMyDaughter, haveDaughterShould	CW=2	67.4	58.1	62.4
$L_C S_C$	POS, whichModalAmI, parentPOS, haveReportingAncestor, whichModalIsMyDaughter, haveDaughterPerfect, whichAuxIsMyDaughter, haveDaughterWh, haveDaughterShould	CW=2	68.5	60.0	64.0
MALLET - Joint Model					
L	POS, whichModalAmI, verbType	GV=1	55.1	45.0	49.6
LS	POS, whichModalAmI, parentPOS, haveReportingAncestor, whichModalIsMyDaughter, haveDaughterPerfect, whichAuxIsMyDaughter, haveDaughterWh, haveDaughterShould	GV=1	64.5	54.4	59.0
Pipeline Model					
$L_C S_C$	POS, whichModalAmI, parentPOS, haveReportingAncestor, whichModalIsMyDaughter, haveDaughterPerfect, whichAuxIsMyDaughter, haveDaughterWh, haveDaughterShould	CW=2	49.8	42.9	46.1

Table 4: Overall Results. CW = Context Width, GV = Gaussian Variance, P = Precision, R = Recall, F = F-Measure

der= "0,1", which makes the CRF similar to Hidden Markov Model. All results reported here use the order= "0,1". We also conducted experiments varying the Gaussian variance parameter from 1.0 to 10.0 using the same experimental setup (i.e. we did not have a distinct tuning corpus) and observed that best results were obtained with a low value of 1 to 3, instead of MALLET's default value of 10.0.

3.3.3 Pipeline Model

We also did experiments to support our choice of the joint model over the pipeline model. We chose the best performing feature configuration of the $L_C S_C$ class (which is the overall best performer as we present in Section 3.5), and set up the pipeline model. We trained a sequence classifier using YAMCHA to identify the head tokens, where tokens are tagged as just propositional heads without distinguishing between CB/NA/NCB. The predicted head tokens were then classified using a 3-Way SVM classifier trained on gold data.

3.4 Evaluation

For evaluation, we used 4-fold cross validation on the training data. The data was divided into 4 folds of which 3 folds were used to train a model which was tested on the 4th fold. We did this with all four configurations and all the reported results in this paper are averaged results across 4 folds. We report Recall and Precision on word tokens in our corpus for each of the three tags. It is worth noting that the majority of the words in our data will not be tagged with any of the three classes. (Recall that most words have neither of the three tags). We also report $F_{\beta=1}$ (F)-measure as the harmonic mean between (P)recision and (R)ecall.

3.5 Results

This section summarizes the results of various experiments we conducted. The best performing feature configuration and corresponding Precision, Recall and F-measure for each experimental setup discussed in previous section is presented in Table 4. The best F-measure for each category under various experimental setups is presented in Table 5.

We obtained the best performance using YAM-

Setup	Class	CB	NCB	NA
Joint-YAMCHA	L_C	61.5	15.2	63.2
Joint-YAMCHA	$L_N S_N$	67.0	28.3	59.9
Joint-YAMCHA	$L_C S_N$	67.6	33.2	64.5
Joint-YAMCHA	$L_C S_C$	69.6	34.1	64.5
Joint-MALLET	L	53.9	7.5	54.1
Joint-MALLET	LS	65.8	40.6	59.1
Pipeline	$L_C S_C$	55.2	16.5	51.3

Table 5: Results per Category (F-Measure)

CHA in a joint model. So, we first analyze this configuration in great detail in Section 3.5.1. We discuss results obtained using MALLET in Section 3.5.2 and the pipeline model in Section-3.5.3.

3.5.1 YAMCHA - Results

As described in Section 3.3.1, we divide our experiments into 4 classes - L_C, $L_N S_N$, $L_C S_N$ and $L_C S_C$. Table 4 presents the best performing feature sets and context width configuration for each class. For all experiments with context, the best result was obtained with a context width of 2, except for L_C, where a context width of 3 gave the best results. The results show that syntactic features improve the classifier performance considerably. The best model obtained for L_C has an F-measure of 56.9%. In $L_N S_N$ it improves marginally to 59.9%. Adding back context to lexical features improves it to 62.4% in $L_C S_N$ whereas addition of context to syntactic features further improves this to 64.0%. We observed that the feature parentPOS has the most impact on increased context widths, among syntactic features.

The improvement pattern of Precision and Recall across the classes is also interesting. Syntactic features with no context improve Recall by 4.8 percentage points over only lexical features with context, whereas Precision improves only by 0.6 points. However, adding back context to lexical features further improves Precision by 4.9 points while Recall just improves by 0.6 points. Finally, adding context of syntactic features improves both Precision and Recall moderately. We infer that syntactic features (without context) help identify more annotatable patterns thereby improving Recall, whereas linear context helps removing the wrong ones, thereby improving Precision.

The per-category F-measure results presented in Table 5 are also interesting. The CB F-measure improves by 8.1 points and NCB improves 18.9 points from L_C to $L_C S_C$. But, the improvement in NA F-measure is only a marginal 1.3 points between L_C and $L_C S_C$. Furthermore, the F-measure decreases by 3.3 points when syntactic and lexical features with no context are used. On analysis, we found that NAs often occur in syntactic structures like *want to find* or *should go* (deontic *should*), in which the relevant words occur in a small linear window. In contrast, NCBs are often signaled by deeper syntactic structures. For example, in *He said that his visit to the US will mainly focus on the humanitarian issues*, a simplified sentence from our training set, the verb *focus* is an NCB because it is in the scope of the reporting verb *said* (specifically, it is its daughter). This could not be captured using the context because *said* and *focus* are far apart in the sentence. But a correct parse tree gives *focus* as the daughter of *said*. So, a feature like haveReportingAncestor could easily capture this. It is also the case that the root of a dependency parse tree would mostly be a CB. This is captured by the feature parentPOS having value 'nil'. This property also cannot be captured by lexical features alone.

However, NCB performs much worse than the other two categories. NCB is a class which occurs rarely compared to CB and NA in our corpus. Out of the $1,357$ propositions tagged, only 176 were NCB. We assume that this could be a main factor of its poor performance.

We analyzed the performance across the folds. Fold-2 contains only 0.03% NCBs compared to 1.89% on the rest of the folds. Similarly, it contains 6.43% NAs compared to 3.82% across other folds. However, our best performing model gives a Recall of 59.1% with a Precision of 69.7% (F-measure 64.0%) for Fold-2, which is as good as other folds. Hence, we observe that our learned model is robust under distributional variations.

3.5.2 MALLET Results

As explained in Section 3.3.2, we explored MALLET-CRF using two experimental conditions L and LS. Table 4 presents the best performing feature sets for both classes. These re-

sults again show that syntactic features improve the classifier performance considerably. The best model obtained for L class has an F-measure of 49.6%, whereas addition of syntactic features improves this to 59.0%. Both Precision and Recall are improved by 9.4 percentage points as well.

However, MALLET-CRF's performance was comparatively worse than YAMCHA's SVM. The best model for MALLET (LS) obtained an F-measure of 59.0% which is 5.0 percentage points less than that of the best model for YAMCHA ($L_C S_C$).

It is interesting to note that MALLET performed well on predicting NCB. The highest NCB F-measure of MALLET - 40.6% is 6.5 percentage points higher than the highest NCB F-measure for YAMCHA. However, corresponding CB and NA F-measures were 61.2% and 56.1% which are much lower than YAMCHA's performance for these categories.

Also, MALLET was more time efficient than YAMCHA. On an average, for our corpus size and feature sets, MALLET ran 3 times as fast as YAMCHA in a cross validation setup (i.e. training and testing together).

3.5.3 Joint Model vs Pipeline Model

As discussed in Section 3.3.3, we set up a pipeline model for the best performing configuration of $L_C S_C$ class of YAMCHA experiments. The head prediction step of the pipeline obtained an F-measure of 83.9% with Precision and Recall of 86.7% and 81.2%, respectively, across all 4 folds. The 3-way classification step to classify the belief of the identified head obtained an accuracy of 72.7% across all folds. In the pipeline model, false positives and false negatives adds up from step 1 and step 2, where as only the true positives of step 2 is considered as the true positives overall. In this way, the overall Precision was only 49.8% and Recall was 42.9% with an F-measure of 46.1% as shown in Table 4. The results for CB/NCB/NA separately are given in Table 5. The per-category best F-measure was decreased by 14.4, 17.6 and 13.2 percentage points from the YAMCHA joint model for CB, NCB and NA, respectively. The performance gap is big enough to conclude that our choice of joint model was right.

4 Related Work

Our work falls in the rich tradition of modeling agents in terms of their cognitive states (for example, (Rao and Georgeff, 1991)) and relating this modeling to language use through extensions to speech act theory (for example, (Perrault and Allen, 1980; Clark, 1996; Bunt, 2000)). These notions have been particularly fruitful in the dialog community, where dialog act tagging is a major topic of research; to cite just one prominent example: (Stolcke et al., 2000). A dialog act represents the communicative intention of the speaker, and its recognition is crucial for the building of dialog systems. The specific contribution of this paper is to investigate exactly how discourse participants signal their beliefs using language, and the strength of their beliefs; this latter point is not usually included in dialog act tagging.

This paper is not concerned with issues relating to logics for belief representation or inferencing that can be done on beliefs (for an overview, see (McArthur, 1988)), nor theories of automatic belief ascription (Wilks and Ballim, 1987). For example, this paper is not concerned with determining whether a belief in the requirement of p entails the belief in p; instead, we are only interested in whether the writer wants the reader to understand whether the writer holds a belief in the requirement that p or in p directly. This paper is also not concerned with subjectivity (Wiebe et al., 2004), the nature of the proposition p (statement about interior world or external world) is not of interest, only whether the writer wants the reader to believe the writer believes p. This paper is also not concerned with opinion and determining the polarity (or strength) of opinion (for example: (Somasundaran et al., 2008)), which corresponds to the desire dimension. Thus, this work is orthogonal to the extensive literature on opinion classification.

The work of (Saurí and Pustejovsky, 2007; Saurí and Pustejovsky, 2008) is, in many respects, very similar to ours. They propose Factbank, which represents the factual interpretation as modality-polarity pairs, extracted from the basic structural elements denoting factuality encoded by Timebank. Also, they attribute the factuality to specific sources within the text. Our work

is more limited in several ways: we currently only model the writer's beliefs; we do not express polarity (we believe we can derive it from the syntax and lexicon); Saurí and Pustejovsky (2008) ask their annotators to perform extensive linguistic transformations on the text to obtain a "normalized" representation of propositional content (we simply ask the annotators to make a judgment about the writer's strength of belief with respect to a given proposition, and expect to be able to extract representations of pure propositional meaning independently); and finally, Saurí and Pustejovsky (2008) have a more fine-grained representation of non-committed belief. While it is plausible to distinguish between more or less firm non-committed belief, we believe the crucial distinction is between committed belief and non-committed belief. Furthermore, Saurí and Pustejovsky (2008) group reported speech with non-belief statements (our NA), while we group them with weak belief (our NCB). The reason for our decision is that we wanted to keep NA as a category which contains no-one's beliefs, as we assumed this is semantically more coherent. The category NCB thus covers beliefs which the writer does not hold firmly or has expressed no opinion on — which is different from propositions which the writer has clearly attributed to other cognitive states (such as desire). In principle, we believe a 4-way distinction is the right approach, but our NCB category is already the least frequent, and splitting it would have resulted in two very rare classes. Another difference include the use of the word "fact" in the FactBank manual, which we avoid because we are interested in cognitive modeling; however, this is merely a terminological issue.

Other related works explored belief systems in an inference scenario as opposed to an intentionality scenario. In work by (Krestel et al., 2008), the authors explore belief in the context of reported speech in news media: they track newspaper text looking for elements indicating evidentiality. This is different from our work, since we seek to make explicit the intention of the author or the speaker.

5 Future Work

We are exploring ways to utilize the FactBank annotated corpus for our purpose, with the goal of automatically converting it to our annotation format. With the added data from FactBank, we hope to be able to split the NCB category into WB (weak belief) and RS (reported speech). We will also explore learning embedded belief attributions, as annotated in FactBank.

We found that the per-sentence F-measure has a small positive correlation with the length-normalized probability of the MICA derivation (a measure of parse confidence). In case of a bad parse, syntax features add noise which in turn reduces classifier performance. We are planning to exploit this correlation in order to choose sentences for selective self-training. Another direction we are looking to extend this work is to employ active learning to overcome the shortcomings of a small training set. Also, we found frequent use of epistemic and deontic modals in our data. Both types of modals have identical syntactic structure, but they receive very different annotations. This is not easily captured in our system. We are exploring ways to handle this.

We will release our Committed Belief Tagging tool as a standalone black-box tool. We also intend to release the annotated corpus.

6 Acknowledgments

This work is supported, in part, by the Johns Hopkins Human Language Technology Center of Excellence. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the sponsor. We thank Bonnie Dorr, Lori Levin and our other partners on the TTO8 project. We also thank several anonymous reviewers for their constructive feedback.

References

Bangalore, Srinivas and Aravind Joshi. 1999. Supertagging: An approach to almost parsing. *Computational Linguistics*, 25(2):237–266.

Bangalore, Srinivas, Pierre Boullier, Alexis Nasr, Owen Rambow, and Benoît Sagot. 2009. MICA:

A probabilistic dependency parser based on tree insertion grammars. In *NAACL HLT 2009 (Short Papers)*.

Bratman, Michael E. 1999 [1987]. *Intention, Plans, and Practical Reason*. CSLI Publications.

Bunt, Harry. 2000. Dialogue pragmatics and context specification. In Bunt, Harry and William J. Black, editors, *Abduction, Belief and Context in Dialogue*, pages 81–150.

Clark, Herbert H. 1996. *Using Language*. cup, Cambridge, England.

Cohen, Philip R. and Hector J. Levesque. 1990. Rational interaction as the basis for communication. In Philip Cohen, Jerry Morgan and James Allen, editors, *Intentions in Communication*. MIT Press.

Diab, Mona T., Lori Levin, Teruko Mitamura, Owen Rambow, Vinodkumar Prabhakaran, and Weiwei Guo. 2009. Committed belief annotation and tagging. In *ACL-IJCNLP '09: Proceedings of the Third Linguistic Annotation Workshop*, pages 68–73, Morristown, NJ, USA. Association for Computational Linguistics.

Grice, Herbert Paul. 1975. Logic and conversation. In Cole, P. and J. Morgan, editors, *Syntax and semantics, vol 3*. Academic Press, New York.

Hovy, Eduard H. 1993. Automated discourse generation using discourse structure relations. *Artificial Intelligence*, 63:341–385.

Krestel, Ralf, Sabine Bergler, and René Witte. 2008. Minding the Source: Automatic Tagging of Reported Speech in Newspaper Articles. In (ELRA), European Language Resources Association, editor, *Proceedings of the Sixth International Language Resources and Evaluation (LREC 2008)*, Marrakech, Morocco, May 28–30.

Kudo, Taku and Yuji Matsumoto. 2000. Use of support vector learning for chunk identification. In *Proceedings of CoNLL-2000 and LLL-2000*, pages 142–144.

Mann, William C. and Sandra A. Thompson. 1987. Rhetorical Structure Theory: A theory of text organization. Technical Report ISI/RS-87-190, ISI.

McArthur, Gregory L. 1988. Reasoning about knowledge and belief: a survey. *Computational Intelligence*, 4:223–243.

McCallum, Andrew Kachites. 2002. Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu.

Moore, Johanna. 1994. *Participating in Explanatory Dialogues*. MIT Press.

Perrault, C. Raymond and James F. Allen. 1980. A plan-based analysis of indirect speech acts. *Computational Linguistics*, 6(3–4):167–182.

Rao, Anand S. and Michael P. Georgeff. 1991. Modeling rational agents within a BDI-architecture. In Allen, James, Richard Fikes, and Erik Sandewall, editors, *Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning*, pages 473–484. Morgan Kaufmann publishers Inc.: San Mateo, CA, USA.

Saurí, Roser and James Pustejovsky. 2007. Determining Modality and Factuality for Textual Entailment. In *First IEEE International Conference on Semantic Computing.*, Irvine, California.

Saurí, Roser and James Pustejovsky. 2008. From Structure to Interpretation: A Double-layered Annotation for Event Factuality. In *Proceedings of the 2nd Linguistic Annotation Workshop*. LREC 2008.

Somasundaran, Swapna, Janyce Wiebe, and Josef Ruppenhofer. 2008. Discourse level opinion interpretation. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 801–808, Manchester, UK, August. Coling 2008 Organizing Committee.

Stolcke, Andreas, Klaus Ries, Noah Coccaro, Elizabeth Shriberg, Rebecca Bates, Daniel Jurafsky, Paul Taylor, Rachel Martin, Carol Van Ess-Dykema, and Marie Meteer. 2000. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational Linguistics*, 26:339–373.

Stone, Matthew. 2004. Intention, interpretation and the computational structure of language. *Cognitive Science*, 24:781–809.

Wiebe, Janyce, Theresa Wilson, Rebecca Bruce, Matthew Bell, and Melanie Martin. 2004. Learning subjective language. In *Computational Linguistics, Volume 30 (3)*.

Wilks, Yorick and Afzal Ballim. 1987. Multiple agents and the heuristic ascription of belief. In *Proceedings of the 10th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 118–124.

Realization of Discourse Relations by Other Means: Alternative Lexicalizations

Rashmi Prasad and **Aravind Joshi**
University of Pennsylvania
rjprasad,joshi@seas.upenn.edu

Bonnie Webber
University of Edinburgh
bonnie@inf.ed.ac.uk

Abstract

Studies of discourse relations have not, in the past, attempted to characterize what serves as evidence for them, beyond lists of frozen expressions, or markers, drawn from a few well-defined syntactic classes. In this paper, we describe how the lexicalized discourse relation annotations of the Penn Discourse Treebank (PDTB) led to the discovery of a wide range of additional expressions, annotated as *AltLex* (*alternative lexicalizations*) in the PDTB 2.0. Further analysis of AltLex annotation suggests that the set of markers is open-ended, and drawn from a wider variety of syntactic types than currently assumed. As a first attempt towards automatically identifying discourse relation markers, we propose the use of syntactic paraphrase methods.

1 Introduction

Discourse relations that hold between the content of clauses and of sentences – including relations of cause, contrast, elaboration, and temporal ordering – are important for natural language processing tasks that require sensitivity to more than just a single sentence, such as summarization, information extraction, and generation. In written text, discourse relations have usually been considered to be signaled either explicitly, as lexicalized with some word or phrase, or implicitly due to adjacency. Thus, while the causal relation between the situations described in the two clauses in Ex. (1) is signalled explicitly by the connective *As a result*, the same relation is conveyed implicitly in Ex. (2).

(1) John was tired. As a result he left early.

(2) John was tired. He left early.

This paper focusses on the problem of how to characterize and identify explicit signals of discourse relations, exemplified in Ex. (1). To refer to all such signals, we use the term "discourse relation markers" (DRMs). Past research (e.g., (Halliday and Hasan, 1976; Martin, 1992; Knott, 1996), among others) has assumed that DRMs are frozen or fixed expressions from a few well-defined syntactic classes, such as conjunctions, adverbs, and prepositional phrases. Thus the literature presents *lists* of DRMs, which researchers try to make as complete as possible for their chosen language. In annotating lexicalized discourse relations of the Penn Discourse Treebank (Prasad et al., 2008), this same assumption drove the initial phase of annotation. A list of "explicit connectives" was collected from various sources and provided to annotators, who then searched for these expressions in the text and annotated them, along with their arguments and senses. The same assumption underlies methods for automatically identifying DRMs (Pitler and Nenkova, 2009). Since expressions functioning as DRMs can also have non-DRM functions, the task is framed as one of classifying given individual tokens as DRM or not DRM.

In this paper, we argue that placing such syntactic and lexical restrictions on DRMs limits a proper understanding of discourse relations, which can be realized in other ways as well. For example, one should recognize that the instantiation (or exemplification) relation between the two sentences in Ex. (3) is explicitly signalled in the second sentence by the phrase *Probably the most egregious example is*, which is sufficient to express the instantiation relation.

(3) Typically, these laws seek to prevent executive branch officials from inquiring into whether certain federal programs make any economic sense or proposing more market-oriented alternatives to regulations. Probably the most egregious example is a

Coling 2008: Poster Volume, pages 1023–1031,
Beijing, August 2010

proviso in the appropriations bill for the executive office that prevents the president's Office of Management and Budget from subjecting agricultural marketing orders to any cost-benefit scrutiny.

Cases such as Ex. (3) show that identifying DRMs cannot simply be a matter of preparing a list of fixed expressions and searching for them in the text. We describe in Section 2 how we identified other ways of expressing discourse relations in the PDTB. In the current version of the corpus (PDTB 2.0.), they are labelled as *AltLex* (*alternative lexicalizations*), and are "discovered" as a result of our lexically driven annotation of discourse relations, including explicit as well as implicit relations. Further analysis of AltLex annotations (Section 3) leads to the thesis that *DRMs are a lexically open-ended class of elements which may or may not belong to well-defined syntactic classes*. The open-ended nature of DRMs is a challenge for their automated identification, and in Section 4, we point to some lessons we have already learned from this annotation. Finally, we suggest that methods used for automatically generating candidate paraphrases may help to expand the set of recognized DRMs for English and for other languages as well (Section 5).

2 AltLex in the PDTB

The Penn Discourse Treebank (Prasad et al., 2008) constitutes the largest available resource of lexically grounded annotations of discourse relations, including both explicit and implicit relations.[1] Discourse relations are assumed to have two and only two arguments, called Arg1 and Arg2. By convention, Arg2 is the argument syntactically associated with the relation, while Arg1 is the other argument. Each discourse relation is also annotated with one of the several senses in the PDTB hierarchical sense classification, as well as the attribution of the relation and its arguments. In this section, we describe how the annotation methodology of the PDTB led to the identification of the AltLex relations.

Since one of the major goals of the annotation was to lexically ground each relation, a first step in the annotation was to identify the explicit markers of discourse relations. Following standard practice, a list of such markers – called "explicit connectives" in the PDTB – was collected from various sources (Halliday and Hasan, 1976; Martin, 1992; Knott, 1996; Forbes-Riley et al., 2006).[2] These were provided to annotators, who then searched for these expressions in the corpus and marked their arguments, senses, and attribution.[3] In the pilot phase of the annotation, we also went through several iterations of updating the list, as and when annotators reported seeing connectives that were not in the current list. Importantly, however, connectives were constrained to come from a few well-defined syntactic classes:

- *Subordinating conjunctions:* e.g., because, although, when, while, since, if, as.
- *Coordinating conjunctions:* e.g., and, but, so, either..or, neither..nor.
- *Prepositional phrases:* e.g., as a result, on the one hand..on the other hand, insofar as, in comparison.
- *adverbs:* e.g., then, however, instead, yet, likewise, subsequently

Ex. (4) illustrates the annotation of an explicit connective. (In all PDTB examples in the paper, Arg2 is indicated in boldface, Arg1 is in italics, the DRM is underlined, and the sense is provided in parentheses at the end of the example.)

(4) *U.S. Trust, a 136-year-old institution that is one of the earliest high-net worth banks in the U.S., has faced intensifying competition from other firms that have established, and heavily promoted, private-banking businesses of their own.* As a result, **U.S. Trust's earnings have been hurt.** (Contingency:Cause:Result)

After all explicit connectives in the list were annotated, the next step was to identify implicit discourse relations. We assumed that such relations are triggered by adjacency, and (because of resource limitations) considered only those that held between sentences within the same paragraph. Annotators were thus instructed to supply a connective – called "implicit connective" – for

[1] http://www.seas.upenn.edu/~pdtb

[2] All explicit connectives annotated in the PDTB are listed in the PDTB manual (PDTB-Group, 2008).

[3] These guidelines are recorded in the PDTB manual.

each pair of adjacent sentences, *as long as the relation was not already expressed with one of the explicit connectives provided to them*. This procedure led to the annotation of implicit connectives such as *because* in Ex. (5), where a causal relation is inferred but no explicit connective is present in the text to express the relation.

(5) *To compare temperatures over the past 10,000 years, researchers analyzed the changes in concentrations of two forms of oxygen.* (Implicit=because) **These measurements can indicate temperature changes**, . . . (Contingency:Cause:reason)

Annotators soon noticed that in many cases, they were not able to supply an implicit connective. Reasons supplied included (a) "there is a relation between these sentences but I cannot think of a connective to insert between them", (b) "there is a relation between the sentences for which I can think of a connective, but it doesn't sound good", and (c) "there is no relation between the sentences". For all such cases, annotators were instructed to supply "NONE" as the implicit connective. Later, we sub-divided these "NONE" implicits into "EntRel", for the (a) type above (an entity-based coherence relation, since the second sentence seemed to continue the description of some entity mentioned in the first); "NoRel" (no relation) for the (c) type; and "AltLex", for the (b) type, which we turn to next.

Closer investigation of the (b) cases revealed that the awkwardness perceived by annotators when inserting an implicit connective was due to *redundancy in the expression of the relation*: Although no explicit connective was present to relate the two sentences, some other expression appeared to be doing the job. This is indeed what we found. Subsequently, instances of AltLex were annotated if:

1. A discourse relation can be inferred between adjacent sentences.

2. There is no explicit connective present to relate them.

3. The annotator is not able to insert an implicit connective to express the inferred relation (having used "NONE" instead), because inserting it leads to an awkward redundancy in expressing the relation.

Under these conditions, annotators were instructed to look for and mark as *Altlex*, whatever *alternative expression* appeared to denote the relation. Thus, for example, Ex. (6) was annotated as AltLex because although a causal relation is inferred between the sentences, inserting a connective like *because* makes expression of the relation redundant. Here the phrase *One reason is* is taken to denote the relation and is marked as *AltLex*.

(6) *Now, GM appears to be stepping up the pace of its factory consolidation to get in shape for the 1990s.* **One reason is mounting competition from new Japanese car plants in the U.S. that are pouring out more than one million vehicles a year at costs lower than GM can match**. (Contingency:Cause:reason)

The result of this procedure led to the annotation of 624 tokens of AltLex in the PDTB. We turn to our analysis of these expressions in the next section.

3 What is found in AltLex?

Several questions arise when considering the AltLex annotations. What kind of expressions are they? What can we learn from their syntax? Do they project discourse relations of a different sort than connectives? How can they be identified, both during manual annotation and automatically? To address these questions, we examined the AltLex annotation for annotated senses, and for common lexico-syntactic patterns extracted using alignment with the Penn Treebank (Marcus et al., 1993).[4]

3.1 Lexico-syntactic Characterization

We found that we could partition AltLex annotation into three groups by (a) whether or not they belonged to one of the syntactic classes admitted as explicit connectives in the PDTB, and (b) whether the expression was frozen (ie, blocking free substitution, modification or deletion of any of its parts) or open-ended. The three groups are shown in Table 1 and discussed below.

[4]The source texts of the PDTB come from the Penn Treebank (PTB) portion of the Wall Street Journal corpus. The PDTB corpus provides PTB tree alignments of all its text span annotations, including connectives, AltLex's, arguments of relations, and attribution spans.

AltLex Group	No (%)	Examples
Syntactically admitted, lexically frozen	92 (14.7%)	quite the contrary (ADVP), for one thing (PP), as well (ADVP), too (ADVP), soon (ADVP-TMP), eventually (ADVP-TMP), thereafter (RB), even (ADVP), especially (ADVP), actually (ADVP), still (ADVP), only (ADVP), in response (PP)
Syntactically free, lexically frozen	54 (8.7%)	What's more (SBAR-ADV), Never mind that (ADVP-TMP;VB;DT), To begin with (VP), So (ADVP-PRD-TPC), Another (DT), further (JJ), As in (IN;IN), So what if (ADVP;IN), Best of all (NP)
Syntactically and lexically free	478 (76.6%)	That compares with (NP-SBJ;VBD;IN), After these payments (PP-TMP), That would follow (NP-SBJ;MD;VB), The plunge followed (NP-SBJ;VBD), Until then (PP-TMP), The increase was due mainly to (NP-SBJ;VBD;JJ;RB;TO), That is why (NP-SBJ;VBZ;WHADVP), Once triggered (SBAR-TMP)
TOTAL	624	–

Table 1: Breakdown of AltLex by Syntactic and Lexical Flexibility. Examples in the third column are accompanied (in parentheses) with their PTB POS tags and constituent phrase labels obtained from the PDTB-PTB alignment.

Syntactically admitted and lexically frozen: The first row shows that 14.7% of the strings annotated as AltLex belong to syntactic classes admitted as connectives and are similarly frozen. (Syntactic class was obtained from the PDTB-PTB alignment.) So, despite the effort in preparing a list of connectives (cf. Section 1), additional ones were still found in the corpus through AltLex annotation. This suggests that any pre-defined list of connectives should only be used to guide annotators in a strategy for "discovering" connectives.

Syntactically free and lexically frozen: AltLex expressions that were frozen but belonged to syntactic classes other than those admitted for the PDTB explicit connectives accounted for 8.7% (54/624) of the total (Table 1, row 2). For example, the AltLex *What's more* (Ex. 7) is parsed as a clause (SBAR) functioning as an adverb (ADV). It is also frozen, in not undergoing any change (eg, *What's less*, *What's bigger*, etc.[5]

(7) Marketers themselves are partly to blame: *They've increased spending for coupons and other short-term promotions at the expense of image-building advertising.* **What's more, a flood of new products has given consumers a dizzying choice of brands, many of which are virtually carbon copies of one other**. (Expansion:Conjunction)

[5]Apparently similar headless relative clauses such as *What's more exciting* differ from *What's more* in not functioning as adverbials, just as NPs.

Many of these AltLex annotations do not constitute a single constituent in the PTB, as with *Never mind that*. These cases suggest that either the restrictions on connectives as frozen expressions should be relaxed to admit all syntactic classes, or the syntactic analyses of these *multi-word expressions* is irrelevant to their function.

Both syntactically and lexically free: This third group (Table 1, row 3) constitutes the majority of AltLex annotations – 76.6% (478/624). Additional examples are shown in Table 2. Common syntactic patterns here include subjects followed by verbs (Table 2a-c), verb phrases with complements (d), adverbial clauses (e), and main clauses with a subordinating conjunction (f).

All these AltLex annotations are freely modifiable, with their fixed and modifiable parts shown in the regular expressions defined for them in Table 2. Each has a fixed "core" phrase shown as lexical tokens in the regular expression, e.g, *consequence of*, *attributed to*, plus obligatory and optional elements shown as syntactic labels. Optional elements are shown in parentheses. <NX> indicates any noun phrase, <PPX>, any prepositional phrase, <VX>, any verb phrase, and

AltLex String	AltLex Pattern
(a) A consequence of their departure could be ...	<DTX> consequence (<PPX>) <VX>
(b) A major reason is ...	<DTX> (<JJX>) reason (<PPX>) <VX>
(c) Mayhap this metaphorical connection made ...	(<ADVX>) <NX> made
(d) ... attributed the increase to ...	attributed <NX> to
(e) Adding to that speculation ...	Adding to <NX>
(f) That may be because ...	<NX> <VX> because

Table 2: Complex AltLex strings and their patterns

<JJX>, any adjectival phrase

These patterns show, for example, that other variants of the identified AltLex *A major reason is* include *The reason is*, *A possible reason for the increase is*, *A reason for why we should consider DRMs as an open class is*, etc. This is robust support for our claim that DRMs should be regarded as an open class: The task of identifying them cannot simply be a matter of checking an *a priori* list.

Note that the optional modification seen here is clearly also possible with many explicit connectives such as *if* (eg, *even if just if*, *only if*), as shown in Appendix C of the PDTB manual (PDTB-Group, 2008). This further supports the thesis that DRMs should be treated as an open class that includes explicit connectives.

3.2 Semantic Characterization

AltLex strings were annotated as denoting the discourse relation that held between otherwise unmarked adjacent utterances (Section 2). We found them to convey this relation in much the same way as anaphoric discourse adverbials. According to (Forbes-Riley et al., 2006), discourse adverbials convey both the discourse relation and an anaphoric reference to its Arg1. The latter may be either explicit (e.g., through the use of a demonstrative like "this" or "that"), or implicit. Thus, both *as a result of that* and *as a result* are discourse adverbials in the same way: the latter refers explicitly to Arg1 via the pronoun "that", while former does so via an implicit internal argument. (A *result* must be a result of something.)

The examples in Table 2 make this same two–part semantic contribution, albeit with more complex expressions referring to Arg1 and more complex modification of the expression denoting the relation. For example, in the AltLex shown in (Table 2c), *Mayhap this metaphorical connection made* (annotated in Ex. (8)), the relation is denoted by the causal verb *made*, while Arg1 is referenced through the definite description *this metaphorical connection*. In addition, the adverb *Mayhap* further modifies the relational verb.

(8) *Ms. Bartlett's previous work, which earned her an international reputation in the non-horticultural art world, often took gardens as its nominal subject.* **Mayhap this metaphorical connection made the BPC Fine Arts Committee think she had a literal green thumb**. (Contingency:Cause:Result)

These complex AltLex's also raise the question of why we find them at all in language. One part of the answer is that these complex AltLex's are used to convey more than just the meaning of the relation. In most cases, we found that substituting the AltLex with an adverbial connective led to some aspect of the meaning being lost, as in Ex. (9-10). Substituting *For example* for the AltLex with an (necessary) accompanying paraphrase of Arg2 loses the information that the example provided as Arg2 is possibly the most egregious one. The connective *for example* does not allow similar modification. This means that one must use a different strategy such as an AltLex expression.

(9) *Typically, these laws seek to prevent executive branch officials from inquiring into whether certain federal programs make any economic sense or proposing more market-oriented alternatives to regulations.* **Probably the most egregious example is a proviso in the appropriations bill for the executive office that prevents the president's Office of Management and Budget from subjecting agricultural marketing orders to any cost-benefit scrutiny**. (Expansion:Instantiation)

(10) For example, a proviso in the appropriations bill for the executive office prevents the president's Of-

fice of Management and Budget from subjecting agricultural marketing orders to any cost-benefit scrutiny.

Another part of the answer to *Why AltLex?* is that it can serve to convey a relation for which the lexicon lacks an adverbial connective. For example, while English has several adverbial connectives that express a "Cause:Consequence" relation (eg, *as a result, consequently*, etc.), it lacks an adverbial connective expressing "Cause:Reason" (or explanation) albeit having at least two subordinating conjunctions that do so (*because* and *since*). Thus, we find an AltLex whenever this relation needs to be expressed between sentences, as shown in Ex. (11).

(11) *But a strong level of investor withdrawals is much more unlikely this time around,* fund managers said. **A major reason is that investors already have sharply scaled back their purchases of stock funds since Black Monday.** (Contingency:Cause:reason)

Note, however, that even for such relations such as Cause:Reason, it is still not the case that a list of canned expressions will be sufficient to generate the Altlex or to identify them, since this relation can itself be further modified. In Ex. (12), for example, the writer intends to convey that there are multiple reasons for the walkout, although only one of them is eventually specified in detail.

(12) *In Chile, workers at two copper mines, Los Bronces and El Soldado,* which belong to the Exxon-owned Minera Disputada, *yesterday voted to begin a full strike tomorrow,* an analyst said. **Reasons for the walkout**, the analyst said, **included a number of procedural issues, such as a right to strike**. (Contingency:Cause:reason)

4 Lessons learned from AltLex

Like all lexical phenomena, DRMs appear to have a power-law distribution, with some very few high-frequency instances like (*and, but*), a block of mid-frequency instances (eg, *after, because, however*), and many many low-frequency instances in the "long tail" (eg, *much as, on the contrary, in short,* etc.). Given the importance of DRMs for recognizing and classifying discourse relations and their arguments, what have we learned from the annotation of AltLex?

First, the number of expressions found through AltLex annotation, that belong to syntactic classes admitted as connectives and also similarly frozen (Table 1, row 1) shows that even in the PDTB, there are additional instances of what we have taken to be explicit connectives. By recognizing them and unambiguously labelling their senses, we will start to reduce the number of "hard cases" of implicit connectives whose sense has to be recognized (Marcu and Echihabi, 2002; Sporleder and Lascarides, 2008; Pitler et al., 2009; Lin et al., 2009). Secondly, the number of tokens of expressions from other syntactic classes that have been annotated as AltLex (Table 1, rows 2 and 3) may actually be higher than was caught via our AltLex annotation, thus making them even more important for discourse processing. To assess this, we selected five of them and looked for all their tokens in the WSJ raw files underlying both the PTB and the PDTB. After eliminating those tokens that had already been annotated, we judged whether the remaining ones were functioning as connectives. Table 3 shows the expressions we used in the first column, with the second and third columns reporting the number of tokens annotated in PDTB, and the number of additional tokens in the WSJ corpus functioning as connectives. (The asterisk next to the expressions is a wild card to allow for variations along the lines discussed for Table 2.) These results show that these DRMs occur two to three times more frequently than already annotated.

Increased frequencies of AltLex occurrence are also observed in discourse annotation projects undertaken subsequent to the PDTB, since they were able to be more sensitive to the presence of AltLex. The Hindi Discourse Relation Bank (HDRB) (Oza et al., 2009), for example, reports that 6.5% of all discourse relations in the HDRB have been annotated as AltLex, compared to 1.5% in the PDTB. This also provides cross-linguistic evidence of the importance of recognizing the full range of DRMs in a language.

5 Identifying DRMs outside the PDTB

As the set of DRMs appears to be both open-ended and distributed like much else in language, with a very long tail, it is likely that many are missing from the one-million word WSJ corpus annotated in the PDTB 2.0. Indeed, in annotating En-

AltLex	Annotated	Unannotated
The reason*	8	15
That's because	11	16
The result*	12	18
That/This would*	5	16
That means	11	17
TOTAL	47	82

Table 3: Annotated and Unannotated instances of AltLex

glish biomedical articles with discourse relations, Yu et al (2008) report finding many DRMs that don't appear in the WSJ (e.g., *as a consequence*). If one is to fully exploit DRMs in classifying discourse relations, one must be able to identify them all, or at least many more of them than we have to date. One method that seems promising is Callison-Burch's paraphrase generation through back-translation on pairs of word-aligned corpora (Callison-Birch, 2007). This method exploits the frequency with which a word or phrase is back translated (from texts in language A to texts in language B, and then back from texts in language B to texts in language A) across a range of pivot languages, into other words or phrases.

While there are many factors that introduce low-frequency noise into the process, including lexical ambiguity and errors in word alignment, Callison-Burch's method benefits from being able to use the many existing word-aligned translation pairs developed for creating translation models for SMT. Recently, Callison-Burch showed that paraphrase errors could be reduced by syntactically constraining the phrases identified through back-translation to ones with the same syntactic category as assigned to the source (Callison-Birch, 2008), using a large set of syntactic categories similar to those used in CCG (Steedman, 2000).

For DRMs, the idea is to identify through back-translation, instances of DRMs that were neither included in our original set of explicit connective nor subsequently found through AltLex annotation. To allow us to carry out a quick pilot study, Callison-Burch provided us with back-translations of 147 DRMs (primarily explicit connectives annotated in the PDTB 2.0, but also including a few from other syntactic classes found through AltLex annotation). Preliminary analysis of the results reveals many DRMs that don't appear anywhere in the WSJ Corpus (eg, *as a consequence*, *as an example*, *by the same token*), as well as additional DRMs that appear in the corpus but were not annotated as AltLex (e.g., *above all*, *after all*, *despite that*). Many of these latter instances appear in the initial sentence of a paragraph, but the annotation of implicit connectives — which is what led to AltLex annotation in the first place (Section 2) — was not carried out on these sentences.

There are two further things to note before closing this discussion. First, there is an additional source of noise in using back-translation paraphrase to expand the set of identified DRMs. This arises from the fact that discourse relations can be conveyed either explicitly or implicitly, and a translated text may not have made the same choices vis-a-vis explicitation as its source, causing additional word alignment errors (some of which are interesting, but most of which are not). Secondly, this same method should prove useful for languages other English, although there will be an additional problem to overcome for languages (such as Turkish) in which DRMs are conveyed through morphology as well as through distinct words and phrases.

6 Related work

We are not the first to recognize that discourse relations can realized by more than just one or two syntactic classes. Halliday and Hasan (1976) document prepositional phrases like *After that* being used to express conjunctive relations. More importantly, they note that any definite description can be substituted for the demonstrative pronoun.

Similarly, Taboada (2006), in looking at how often RST-based rhetorical relations are realized by discourse markers, starts by considering only adverbials, prepositional phrases, and conjunctions, but then notes the occurrence of a single instance of a nominal fragment *The result* in her corpus. Challenging the RST assumption that the basic unit of a discourse is a clause, with discourse relations holding between adjacent clausal units, Kibble (1999) provides evidence that *informational* discourse relations (as opposed to *intentional* discourse relations) can hold intra-clausally as well, with the relation "verbalized" and its arguments realized as nominalizations, as in *Early treatment with Brand X can prevent a cold sore developing*. Since his focus is intra-clausal, he does not observe that verbalized discourse relations can hold across sentences as well, where a verb and one of its arguments function similarly to a discourse adverbial, and in the end, he does not provide a proposal for how to systematically identify these alternative realizations. Le Huong et al. (2003), in developing an algorithm for recognizing discourse relations, consider non-verbal realizations (called NP cues) in addition to verbal realizations (called VP cues). However, they provide only one example of such a cue ("the result"). Like Kibble (1999), Danlos (2006) and Power (2007) also focus only on identifying verbalizations of discourse relations, although they do consider cases where such relations hold across sentences.

What has not been investigated in prior work is the basis for the alternation between connectives and AltLex's, although there are several accounts of why a language may provide more than one connective that conveys the same relation. For example, the alternation in Dutch between *dus* ("so"), *daardoor* ("as a result"), and *daarom* ("that's why") is explained by Pander Maat and Sanders (2000) as having its basis in "subjectivity".

7 Conclusion and Future Work

Categorizing and identifying the range of ways in which discourse relations are realized is important for both discourse understanding and generation. In this paper, we showed that existing practices of cataloguing these ways as lists of closed class expressions is problematic. We drew on our experience in creating the lexically grounded annotations of the Penn Discourse Treebank, and showed that markers of discourse relations should instead be treated as open-class items, with unconstrained syntactic possibilities. Manual annotation and automatic identification practices should develop methods in line with this finding if they aim to exhaustively identify all discourse relation markers.

Acknowledgments

We want to thank Chris Callison-Burch, who graciously provided us with EuroParl back-translation paraphrases for the list of connectives we sent him. This work was partially supported by NSF grant IIS-07-05671.

References

Callison-Birch, Chris. 2007. *Paraphrasing and Translation*. Ph.D. thesis, School of Informatics, University of Edinburgh.

Callison-Birch, Chris. 2008. Syntactic constraints on paraphrases extracted from parallel corpora. In *Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Danlos, Laurence. 2006. Discourse verbs. In *Proceedings of the 2nd Workshop on Constraints in Discourse*, pages 59–65, Maynooth, Ireland.

Forbes-Riley, Katherine, Bonnie Webber, and Aravind Joshi. 2006. Computing discourse semantics: The predicate-argument semantics of discourse connectives in D-LTAG. *Journal of Semantics*, 23:55–106.

Halliday, M. A. K. and Ruqaiya Hasan. 1976. *Cohesion in English*. London: Longman.

Huong, LeThanh, Geetha Abeysinghe, and Christian Huyck. 2003. Using cohesive devices to recognize rhetorical relations in text. In *Proceedings of 4th Computational Linguistics UK Research Colloquium (CLUK 4)*, University of Edinburgh, UK.

Kibble, Rodger. 1999. Nominalisation and rhetorical structure. In *Proceedings of ESSLLI Formal Grammar conference*, Utrecht.

Knott, Alistair. 1996. *A Data-Driven Methodology for Motivating a Set of Coherence Relations*. Ph.D. thesis, University of Edinburgh, Edinburgh.

Lin, Ziheng, Min-Yen Kan, and Hwee Tou Ng. 2009. Recognizing implicit discourse relations in the penn discourse treebank. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Singapore.

Maat, Henk Pander and Ted Sanders. 2000. Domains of use or subjectivity? the distribution of three dutch causal connectives explained. *TOPICS IN ENGLISH LINGUISTICS*, pages 57–82.

Marcu, Daniel and Abdessamad Echihabi. 2002. An unsupervised approach to recognizing discourse relations. In *Proceedings of the Association for Computational Linguistics*.

Marcus, Mitchell P., Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of english: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Martin, James R. 1992. *English text: System and structure*. Benjamins, Amsterdam.

Oza, Umangi, Rashmi Prasad, Sudheer Kolachina, Dipti Mishra Sharma, and Aravind Joshi. 2009. The hindi discourse relation bank. In *Proceedings of the ACL 2009 Linguistic Annotation Workshop III (LAW-III)*, Singapore.

Pitler, Emily and Ani Nenkova. 2009. Using syntax to disambiguate explicit discourse connectives in text. In *Proceedings of the Joint Conference of the 47th Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing*, Singapore.

Pitler, Emily, Annie Louis, and Ani Nenkova. 2009. Automatic sense prediction for implicit discourse relations in text. In *Proceedings of the Joint Conference of the 47th Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing*.

Power, Richard. 2007. Abstract verbs. In *ENLG '07: Proceedings of the Eleventh European Workshop on Natural Language Generation*, pages 93–96, Morristown, NJ, USA. Association for Computational Linguistics.

Prasad, Rashmi, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Livio Robaldo, Aravind Joshi, and Bonnie Webber. 2008. The Penn Discourse TreeBank 2.0. In *Proceedings of 6th International Conference on Language Resources and Evaluation (LREC 2008)*.

PDTB-Group. 2008. The Penn Discourse TreeBank 2.0 Annotation Manual. Technical Report IRCS-08-01, Institute for Research in Cognitive Science, University of Pennsylvania.

Sporleder, Caroline and Alex Lascarides. 2008. Using automatically labelled examples to classify rhetorical relations: an assessment. *Natural Language Engineering*, 14(3):369–416.

Steedman, Mark. 2000. *The Syntactic Process*. MIT Press, Cambridge MA.

Taboada, Maite. 2006. Discourse markers as signals (or not) of rhetorical relations. *Journal of Pragmatics*, 38(4):567–592.

Yu, Hong, Nadya Frid, Susan McRoy, P Simpson, Rashmi Prasad, Alan Lee, and Aravind Joshi. 2008. Exploring discourse connectivity in biomedical text for text mining. In *Proceedings of the 16th Annual International Conference on Intelligent Systems for Molecular Biology BioLINK SIG Meeting*, Toronto, Canada.

Designing Agreement Features for Realization Ranking

Rajakrishnan Rajkumar and **Michael White**
Department of Linguistics
The Ohio State University
{raja,mwhite}@ling.osu.edu

Abstract

This paper shows that incorporating linguistically motivated features to ensure correct animacy and number agreement in an averaged perceptron ranking model for CCG realization helps improve a state-of-the-art baseline even further. Traditionally, these features have been modelled using hard constraints in the grammar. However, given the graded nature of grammaticality judgements in the case of animacy we argue a case for the use of a statistical model to rank competing preferences. Though subject-verb agreement is generally viewed to be syntactic in nature, a perusal of relevant examples discussed in the theoretical linguistics literature (Kathol, 1999; Pollard and Sag, 1994) points toward the heterogeneous nature of English agreement. Compared to writing grammar rules, our method is more robust and allows incorporating information from diverse sources in realization. We also show that the perceptron model can reduce balanced punctuation errors that would otherwise require a post-filter. The full model yields significant improvements in BLEU scores on Section 23 of the CCGbank and makes many fewer agreement errors.

1 Introduction

In recent years a variety of statistical models for realization ranking that take syntax into account have been proposed, including generative models (Bangalore and Rambow, 2000; Cahill and van Genabith, 2006; Hogan et al., 2007; Guo et al., 2008), maximum entropy models (Velldal and Oepen, 2005; Nakanishi et al., 2005) and averaged perceptron models (White and Rajkumar, 2009). To our knowledge, however, none of these models have included features specifically designed to handle grammatical agreement, an important task in surface realization. In this paper, we show that incorporating linguistically motivated features to ensure correct animacy and verbal agreement in an averaged perceptron ranking model for CCG realization helps improve a state-of-the-art baseline even further. We also demonstrate the utility of such an approach in ensuring the correct presentation of balanced punctuation marks.

Traditionally, grammatical agreement phenomena have been modelled using hard constraints in the grammar. Taking into consideration the range of acceptable variation in the case of animacy agreement and facts about the variety of factors contributing to number agreement, the question arises: tackle agreement through grammar engineering, or via a ranking model? In our experience, trying to add number and animacy agreement constraints to a grammar induced from the CCGbank (Hockenmaier and Steedman, 2007) turned out to be surprisingly difficult, as hard constraints often ended up breaking examples that were working without such constraints, due to exceptions, sub-regularities and acceptable variation in the data. With sufficient effort, it is conceivable that an approach incorporating hard agreement constraints could be refined to underspecify cases where variation is acceptable, but even so, one would want a ranking model to capture preferences in these cases, which might vary depending on genre, dialect or domain. Given that

Coling 2008: Poster Volume, pages 1032–1040,
Beijing, August 2010

a ranking model is desirable in any event, we investigate here the extent to which agreement phenomena can be more robustly and simply handled using a ranking model alone, with no hard constraints in the grammar.

We also show here that the perceptron model can reduce balanced punctuation errors that would otherwise require a post-filter. As White and Rajkumar (2008) discuss, in CCG it is not feasible to use features in the grammar to ensure that balanced punctuation (e.g. paired commas for NP appositives) is used in all and only the appropriate places, given the word-order flexibility that crossing composition allows. While a post-filter is a reasonably effective solution, it can be prone to search errors and does not allow balanced punctuation choices to interact with other choices made by the ranking model.

The starting point for our work is a CCG realization ranking model that incorporates Clark & Curran's (2007) normal-form syntactic model, developed for parsing, along with a variety of n-gram models. Although this syntactic model plays an important role in achieving top BLEU scores for a reversible, corpus-engineered grammar, an error analysis nevertheless revealed that many errors in relative pronoun animacy agreement and subject-verb number agreement remain with this model. In this paper, we show that features specifically designed to better handle these agreement phenomena can be incorporated into a realization ranking model that makes many fewer agreement errors, while also yielding significant improvements in BLEU scores on Section 23 of the CCGbank. These features make use of existing corpus annotations — specifically, PTB function tags and BBN named entity classes (Weischedel and Brunstein, 2005) — and thus they are relatively easy to implement.

1.1 The Graded Nature of Animacy Agreement

To illustrate the variation that can be found with animacy agreement phenomena, consider first animacy agreement with relative pronouns. In English, an inanimate noun can be modified by a relative clause introduced by *that* or *which*, while an animate noun combines with *who(m)*. With some nouns though — such as *team*, *group*, *squad*, etc. — animacy status is uncertain, and these can be found with all the three relative pronouns (*who, which* and *that*). Google counts suggest that all three choices are almost equally acceptable, as the examples below illustrate:

(1) The groups who protested against plans to remove asbestos from the nuclear submarine base at Faslane claimed victory when it was announced the government intends to dispose of the waste on site. (The Glasgow Herald; Jun 25, 2010)

(2) Mr. Dorsch says the HIAA is working on a proposal to establish a privately funded reinsurance mechanism to help cover small groups that ca n't get insurance without excluding certain employees . (WSJ0518.35)

1.2 The Heterogeneous Nature of Number Agreement

Subject-verb agreement can be described as a constraint where the verb agrees with the subject in terms of agreement features (number and person). Agreement has often been considered to be a syntactic phenomenon and grammar implementations generally use syntactic features to enforce agreement constraints (e.g. Velldal and Oepen, 2005). However a closer look at our data and a survey of the theoretical linguistics literature points toward a more heterogeneous conception of English agreement. Purely syntactic accounts are problematic when the following examples are considered:

(3) Five miles is a long distance to walk. (Kim, 2004)

(4) King prawns cooked in chili salt and pepper was very much better, a simple dish succulently executed. (Kim, 2004)

(5) " I think it will shake confidence one more time , and a lot of this business is based on client confidence . " (WSJ1866.10)

(6) It 's interesting to find that a lot of the expensive wines are n't always walking out the door . (WSJ0071.53)

In Example (3) above, the subject and determiner are plural while the verb is singular. In (4), the singular verb agrees with the dish, rather than with individual prawns. Measure nouns such as *lot*, *ton*, etc. exhibit singular agreement with the determiner *a*, but varying agreement with the verb depending on the head noun of the measure noun's *of*-complement. As is also well known, British and American English differ in subject-verb agreement with collective nouns. Kathol (1999) proposes an explanation where agreement is determined by the semantic properties of the noun rather than by its morphological properties. This accounts for all the cases above. In the light of this explanation, specifying agreement features in the logical form for realization could perhaps solve the problem. However, the semantic view of agreement is not completely convincing due to counterexamples like the following discussed in the literature (reported in Kim (2004)):

(7) Suppose you meet someone and they are totally full of themselves

(8) Those scissors are missing.

In Example (7), the pronoun *they* used in a generic sense is linked to the singular antecedent *someone*, but its plural feature triggers plural agreement with the verb. Example (8) illustrates a situation where the subject *scissors* is arguably semantically singular, but exhibits plural morphology and plural syntactic agreement with both the determiner as well as the verb. Thus this suggests that English has a set of heterogeneous agreement patterns rather than purely syntactic or semantic ones. This is also reflected in the proposal for a hybrid agreement system for English (Kim, 2004), where the morphology tightly interacts with the system of syntax, semantics, or even pragmatics to account for agreement phenomena. Our machine learning-based approach approximates the insights discussed in the theoretical linguistics literature. Writing grammar rules to get these facts right proved to be surprisingly difficult (e.g. discerning the actual nominal head contributing agreement feature in cases like *areas of the factory were/*was* vs. *a lot of wines are/*is*) and required a list of measure nouns and partitive quantifiers. We investigate here the extent to which a machine learning–based approach is a simpler, practical alternative for acquiring the relevant generalizations from the data by combining information from various information sources.

The paper is structured as follows. Section 2 provides CCG background. Section 3 describes the features we have designed for animacy and number agreement as well as for balanced punctuation. Section 4 presents our evaluation of the impact of these features in averaged perceptron realization ranking models, tabulating specific kinds of errors in the CCGbank development section as well as overall automatic metric scores on Section 23. Section 5 compares our results to those obtained with related systems. Finally, Section 6 concludes with a summary of the paper's contributions.

2 Background

2.1 Surface Realization with Combinatory Categorial Grammar (CCG)

CCG (Steedman, 2000) is a unification-based categorial grammar formalism which is defined almost entirely in terms of lexical entries that encode sub-categorization information as well as syntactic feature information (e.g. number and agreement). Complementing function application as the standard means of combining a head with its argument, type-raising and composition support transparent analyses for a wide range of phenomena, including right-node raising and long distance dependencies. An example syntactic derivation appears in Figure 1, with a long-distance dependency between *point* and *make*. Semantic composition happens in parallel with syntactic composition, which makes it attractive for generation.

OpenCCG is a parsing/generation library which works by combining lexical categories for words using CCG rules and multi-modal extensions on rules (Baldridge, 2002) to produce derivations. Conceptually these extensions are on lexical categories. Surface realization is the process by which logical forms are transduced to strings. OpenCCG uses a hybrid symbolic-statistical chart realizer (White, 2006) which takes logical forms as input and produces sentences by using CCG com-

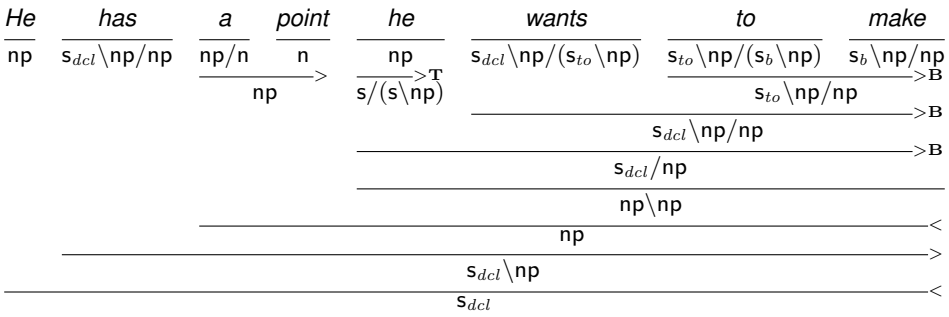


Figure 1: Syntactic derivation from the CCGbank for *He has a point he wants to make [. . .]*

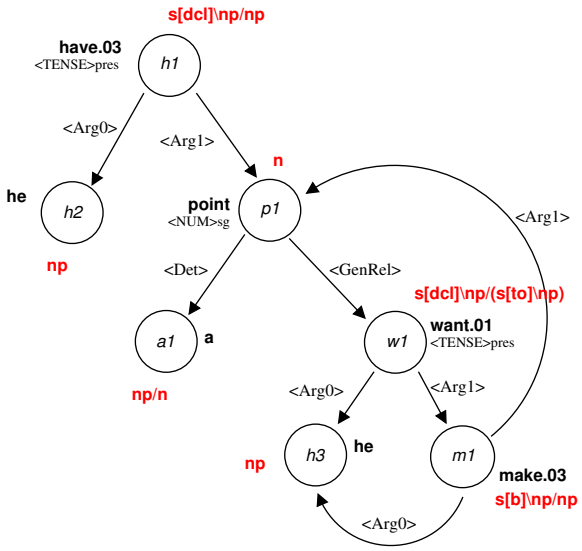


Figure 2: Semantic dependency graph from the CCGbank for *He has a point he wants to make [. . .]*, along with gold-standard supertags (category labels)

binators to combine signs. Edges are grouped into equivalence classes when they have the same syntactic category and cover the same parts of the input logical form. Alternative realizations are ranked using integrated n-gram or perceptron scoring, and pruning takes place within equivalence classes of edges. To more robustly support broad coverage surface realization, OpenCCG greedily assembles fragments in the event that the realizer fails to find a complete realization.

To illustrate the input to OpenCCG, consider the semantic dependency graph in Figure 2. In the graph, each node has a lexical predication (e.g. **make.03**) and a set of semantic features (e.g. ⟨NUM⟩sg); nodes are connected via dependency relations (e.g. ⟨ARG0⟩). (Gold-standard supertags, or category labels, are also shown; see Section 2.2 for their role in hypertagging.) Internally, such graphs are represented using Hybrid Logic Dependency Semantics (HLDS), a dependency-based approach to representing linguistic meaning (Baldridge and Kruijff, 2002). In HLDS, each semantic head (corresponding to a node in the graph) is associated with a nominal that identifies its discourse referent, and relations between heads and their dependents are modeled as modal relations.

For our experiments, we use an enhanced version of the CCGbank (Hockenmaier and Steedman, 2007)—a corpus of CCG derivations derived from the Penn Treebank—with Propbank (Palmer et al., 2005) roles projected onto it (Boxwell and White, 2008). Additionally, certain multi-word NEs were collapsed using underscores so that they are treated as atomic entities in the input to the realizer. To engineer a grammar from this corpus suitable for realization with OpenCCG, the derivations are first revised to reflect the lexicalized treatment of coordination and punctuation assumed by the multi-modal version of CCG that is implemented in OpenCCG (White and Rajkumar, 2008). Further changes are necessary to support semantic dependencies rather than surface syntactic ones; in particular, the features and unification constraints in the categories related to semantically empty function words such complementizers, infinitival-*to*, expletive subjects, and case-marking prepositions are adjusted to reflect their purely syntactic status.

2.2 Hypertagging

A crucial component of the OpenCCG realizer is the *hypertagger* (Espinosa et al., 2008), or supertagger for surface realization, which uses a maximum entropy model to assign the most likely lexical categories to the predicates in the input logical form, thereby greatly constraining the realizer's search space.[1] Category label prediction is done at run-time and is based on contexts within the directed graph structure as shown in Figure 2, instead of basing category assignment on linear word and POS context as in the parsing case.

3 Feature Design

The features we employ in our baseline perceptron ranking model are of three kinds. First, as in the log-linear models of Velldal & Oepen and Nakanishi et al., we incorporate the log probability of the candidate realization's word sequence according to our linearly interpolated language models as a single feature in the perceptron model. Since our language model linearly interpolates three component models, we also include the log prob from each component language model as a feature so that the combination of these components can be optimized. Second, we include syntactic features in our model by implementing Clark & Curran's (2007) normal form model in OpenCCG. The features of this model are listed in Table 1; they are integer-valued, representing counts of occurrences in a derivation. Third, we include discriminative n-gram features (Roark et al., 2004), which count the occurrences of each n-gram that is scored by our factored language model, rather than a feature whose value is the log probability determined by the language model. Table 2 depicts the new animacy, agreement and punctuation features being introduced as part of this work. The next two sections describe these features in more detail.

3.1 Animacy and Number Agreement

Underspecification as to the choice of pronoun in the input leads to competing realizations involving the relative pronouns *who*, *that*, *which* etc. The existing ranking models (n-gram models as well as perceptron) often allow the top-ranked output to have the relative pronoun *that* associated with animate nouns. The existing normal form model uses the word forms as well as part-of-speech tag based features. Though this is useful for associating proper nouns (tagged NNP or NNPS) with *who*, for other nouns (as in *consumers who* vs. *consumers that/which*), the model often prefers the infelicitous pronoun. So here we designed features which also took into account the named entity class of the head noun as well as the stem of the head noun. These features aid the discriminative n-gram features (*PERSON, which* has high negative weight). As the results section discusses,

[1] The approach has been dubbed *hypertagging* since it operates at a level "above" the syntax, moving from semantic representations to syntactic categories.

Feature Type	Example
LexCat + Word	$s/s/np$ + before
LexCat + POS	$s/s/np$ + IN
Rule	$s_{dcl} \rightarrow np\ s_{dcl}\backslash np$
Rule + Word	$s_{dcl} \rightarrow np\ s_{dcl}\backslash np$ + bought
Rule + POS	$s_{dcl} \rightarrow np\ s_{dcl}\backslash np$ + VBD
Word-Word	$\langle$company, $s_{dcl} \rightarrow np\ s_{dcl}\backslash np$, bought$\rangle$
Word-POS	$\langle$company, $s_{dcl} \rightarrow np\ s_{dcl}\backslash np$, VBD$\rangle$
POS-Word	$\langle$NN, $s_{dcl} \rightarrow np\ s_{dcl}\backslash np$, bought$\rangle$
Word + Δ_w	$\langle$bought, $s_{dcl} \rightarrow np\ s_{dcl}\backslash np\rangle + d_w$
POS + Δ_w	$\langle$VBD, $s_{dcl} \rightarrow np\ s_{dcl}\backslash np\rangle + d_w$
Word + Δ_p	$\langle$bought, $s_{dcl} \rightarrow np\ s_{dcl}\backslash np\rangle + d_p$
POS + Δ_p	$\langle$VBD, $s_{dcl} \rightarrow np\ s_{dcl}\backslash np\rangle + d_p$
Word + Δ_v	$\langle$bought, $s_{dcl} \rightarrow np\ s_{dcl}\backslash np\rangle + d_v$
POS + Δ_v	$\langle$VBD, $s_{dcl} \rightarrow np\ s_{dcl}\backslash np\rangle + d_v$

Table 1: Baseline features: Basic and dependency features from Clark & Curran's (2007) normal form model; distances are in intervening words, punctuation marks and verbs, and are capped at 3, 3 and 2, respectively

Feature	Example
Animacy features	
Noun Stem + Wh-pronoun	researcher + who
Noun Class + Wh-pronoun	PER_DESC + who
Number features	
Noun + Verb	people + are
NounPOS + Verb	NNS + are
Noun + VerbPOS	people + VBP
NounPOS + VerbPOS	NNS + VBP
Noun_of + Verb	lot_of + are
Noun_of + VerbPOS	lot_of + VBP
NounPOS_of + Verb	NN_of + are
NounPOS_of + VerbPOS	NN_of + VBP
Noun_of + *of*-complementPOS + VerbPOS	lot_of + NN + VBZ
NounPOS_of + *of*-complementPOS + VerbPOS	NN_of + NN + VBZ
Noun_of + *of*-complementPOS + Verb	lot_of + NN + is
NounPOS_of + *of*-complementPOS + Verb	NN_of + NN + is
Punctuation feature	
Balanced Punctuation Indicator	$unbalPunct=1

Table 2: New features introduced

NE classes like PER_DESC contribute substantially towards animacy preferences.

For number agreement, we designed three classes of features (c.f. *Number Agr* row in Table 2). Each of these classes results in 4 features. During feature extraction, subjects of the verbs tagged VBZ and VBP and verbs *was, were* were identified using the PTB NP-SBJ function tag annotation projected on to the appropriate arguments of lexical categories of verbs. The first class of features encoded all possible combinations of subject-verb word forms and parts of speech tags. In the case of NPs involving *of*-complements like *a lot of ...* (Examples 5 and 6), feature classes 2 and 3 were extracted (class 1 was excluded). Class 2 features encode the fact that the syntactic head has an associated *of*-complement, while class 3 features also include the part of speech tag of the complement. In the case of conjunct/disjunct VPs and subject NPs, the feature specifically looked at the parts of speech of both the NPs/VPs forming the conjunct/disjunct. The motivation behind such a design was to glean syntactic and semantic generalizations from the data. During feature extraction, from each derivation, counts of animacy and agreement features were obtained.

3.2 Balanced Punctuation

A complex issue that arises in the design of bidirectional grammars is ensuring the proper presentation of punctuation. Among other things, this involves the task of ensuring the correct realization of commas introducing noun phrase appositives.

(9) John, CEO of ABC, loves Mary.
(10) * John, CEO of ABC loves Mary.
(11) Mary loves John, CEO of ABC.
(12) * Mary loves John, CEO of ABC,.
(13) Mary loves John, CEO of ABC, madly.
(14) * Mary loves John, CEO of ABC madly.

As of now, n-gram models rule out examples like 12 above. All the other unacceptable examples are ruled out using a post-filter on realized derivations. As described in White and Rajkumar (2008), the need for the filter arises because a feature-based approach appears to be inadequate for dealing with the class of examples presented above in CCG. This approach involves the incorporation of syntactic features for punctuation into atomic categories so that certain combinations are blocked. To ensure proper appositive balancing sentence finally, the rightmost element in the sentence should transmit a relevant feature to the clause level, which the sentence-final period can then check for the presence of right-edge punctuation. However, the feature schema does not constrain cases of balanced punctuation in cases involving crossing composition and extraction. However, in this paper we explore a statistical approach to ensure proper balancing of NP apposition commas. The first step in this solution is the introduction of a feature in the grammar which indicates balanced vs. unbalanced marks. We modified the result categories of unbalanced appositive commas and dashes to include a feature marking unbalanced punctuation, as follows:

(15) $, \vdash \mathsf{np}_{\langle 1 \rangle unbal=comma} \backslash_\star \mathsf{np}_{\langle 1 \rangle} /_\star \mathsf{np}_{\langle 2 \rangle}$

Then, during feature extraction, derivations were examined to detect categories such as $\mathsf{np}_{unbal=comma}$, and checked to make sure this NP is followed by another punctuation mark in the string such as a full stop. The feature indicates the presence or absence of unbalanced punctuation in the derivation.

4 Evaluation

4.1 Experimental Conditions

For the experiments reported below, we used a lexico-grammar extracted from Sections 02–21 of our enhanced CCGbank with collapsed NEs, a hypertagging model incorporating named entity class features, and a trigram factored language model over words, named entity classes, part-of-speech tags and supertags. Perceptron training events were generated for each training section separately. The hypertagger and POS/supertag language model were trained on all the training sections, while separate word-based models were trained excluding each of the training sections in turn. Event files for 26530 training sentences with complete realizations were generated, with an average n-best list size of 18.2. The complete set of models is listed in Table 3.

Model	Description
full-model	All the feats from models below
agr-punct	Baseline Feats + Punct + Num-Agr
wh-punct	Baseline Feats + Punct + Animacy-Agr
baseline-punct	Baseline Feats + Punct
baseline	Log prob + n-gram +Syntactic features

Table 3: Legend for experimental conditions

4.2 Results

Realization results on the development and test sections are given in Table 4. For the development section, in terms of both exact matches and BLEU scores, the model with all the three features discussed above (agreement, animacy and punctuation) performs better than the baseline which does not have any of these features. However, using these criteria, the best performing model is actually the model which has agreement and punctuation features. The model containing all the features does better than the punctuation-feature only model, but performs slightly worse than the agreement-punctuation model. Section 23, the test section, confirms that the model with all the features performs better than the baseline model. We calculated statistical significance for the main results using bootstrap random sampling.[2] After re-sampling 1000 times, significance was calculated using a paired t-test (999 d.f.). The results indicated that the model with all the features in it (full-model) exceeded the baseline with $p < 0.0001$. However, exact matches and BLEU scores do not necessarily reflect the extent to which important grammatical flaws have been reduced. So to judge the effectiveness of the new features, we computed the percentage of errors of each type that were present in the best Section 00 realization selected by each of these models. Also note that our baseline results differ slightly from the corresponding results reported in White and Rajkumar (2009) in spite of using the same feature set because quotes were introduced into the corpus on which these experiments were conducted. Previous results were based on the original CCGbank text where quotation marks are absent.

Table 6 reports results of the error analysis. It

[2]Scripts for running these tests are available at `http://projectile.sv.cmu.edu/research/public/tools/bootStrap/tutorial.htm`

Section	Model	%Exact	%Compl.	BLEU
00	baseline	38.18	82.47	0.8341
	baseline-punct	37.97	82.47	0.8340
	wh-punct	38.93	82.53	0.8360
	full-model	40.47	82.53	0.8403
	agr-punct	40.84	82.53	**0.8414**
23	baseline	38.98	83.39	0.8442
	full-model	40.09	83.35	**0.8446**

Table 4: Results (98.9% coverage)—percentage of exact match and grammatically complete realizations and BLEU scores

Model	METEOR	TERP
baseline	0.9819	0.0939
baseline-punct	0.9819	0.0939
wh-punct	0.9827	0.0923
agr-punct	0.9821	0.0902
full-model	0.9826	0.0909

Table 5: Section 00 METEOR and TERP scores

can be seen that the punctuation-feature is effective in reducing the number of sentences with unbalanced punctuation marks. Similarly, the full model has fewer animacy mismatches and just about the same number of errors of the other two types, though it performs slightly worse than the agreement-only model in terms of BLEU scores and exact matches. We also manually examined the remaining cases of animacy agreement errors in the output of the full model here. Of the remaining 18 errors, 14 were acceptable paraphrases involving object relative clauses (eg. wsj_0083.40 ... *the business that*/∅ *a company can generate*). We also provide METEOR and TERP scores for these models (Table 5). In recently completed work on the creation of a human-rated paraphrase corpus to evaluate NLG systems, our analyses showed that BLEU, METEOR and TERP scores correlate moderately with human judgments of adequacy and fluency, and that the most reliable system-level comparisons can be made only by looking at all three metrics.

4.3 Examples

Table 7 presents four examples where the full model differs from the baseline. Example wsj_0003.8 illustrates an example where the NE tag PER_DESC for *researchers* helps the perceptron model enforce the correct animacy agreement, while the two baseline models prefer the

Ref-wsj_0003.8 full,agr,wh	neither Lorillard nor the **researchers who** studied the workers were aware of any research on smokers of the Kent cigarettes
baseline,baseline-punct	neither Lorillard nor the **researchers that** studied the workers were aware of any research on smokers of the Kent cigarettes .
Ref-wsj_0003.18 agr-punct, full	the **plant** , which is owned by Hollingsworth & Vose Co. , **was** under contract with lorillard to make the cigarette filters .
baselines, wh	the **plant** , which is owned by Hollingsworth & Vose Co. , **were** under contract with lorillard to make the cigarette filters .
Ref-wsj_0018.6 agr-punct, full model	while **many of the risks were** anticipated when minneapolis-based Cray Research first announced the spinoff ...
agr-punct, full	while **many of the risks were** anticipated when minneapolis-based Cray Research first announced the spinoff ...
baselines	while **many of the risks was** anticipated when minneapolis-based Cray Research announced the spinoff ...
Ref-wsj_0070.4 agr-punct, full	Giant Group is led by three Rally 's directors , Burt Sugarman , James M. Trotter III and William E. Trotter II that last month indicated that **they hold** a 42.5 % stake in Rally 's and **plan** to seek a majority of seats on ...
all others	Giant Group is led by three Rally 's directors , Burt Sugarman , James M. Trotter III and William E. Trotter II that last month indicated that **they holds** a 42.5 % stake in Rally 's and **plans** to seek a majority of seats on ...
Ref-wsj_0047.5	... the ban wo n't stop privately funded tissue-transplant research or federally funded fetal-tissue research that does n't involve transplants .
agr, full	... the ban wo n't stop tissue-transplant privately funded **research** or federally funded fetal-tissue **research** that **does n't** involve transplants .
baselines, wh	... the ban wo n't stop tissue-transplant privately funded **research** or federally funded fetal-tissue **research** that **do n't** involve transplants .

Table 7: Examples of realized output

Model	#Punct-Errs	%Agr-Errs	%WH-Errs
baseline	39	11.05	22.44
baseline-punct	0	10.79	20.77
wh-punct	11	10.87	13.53
agr-punct	8	4.0	21.84
full-model	10	4.31	15.53

Table 6: Error analysis of Section 00 complete realizations (total of 1554 agreement cases; total of 207 WH-pronoun cases)

that realization. Example wsj_0003.18 illustrates an instance of simple subject-verb agreement being enforced by the models containing the agreement features. Example wsj_0070.4 presents a more complex situation where a single subject has to agree with both verbs in a conjoined verb phrase. The last example in Table 7 shows the case of a NP subject which is a disjunction of two individual NPs. In both these cases, while the baseline models do not enforce the correct choice, the models with the agreement features do get this right. This is because our agreement features are sensitive to the properties of both NP and VP conjuncts/disjuncts. In addition, most of the realizations involving *of*-complements are also ranked correctly. In the final example sentence provided (i.e. wsj_0018.6), the models with the agreement features are able to enforce the correct the agreement constraints in the phrase *many of the risks were* in contrast to the baseline models.

5 Conclusion

In this paper, we have shown for the first time that incorporating linguistically motivated features to ensure correct animacy and number agreement in a statistical realization ranking model yields significant improvements over a state-of-the-art baseline. While agreement has traditionally been modelled using hard constraints in the grammar, we have argued that using a statistical ranking model is a simpler and more robust approach that is capable of learning competing preferences and cases of acceptable variation. Our approach also approximates insights about agreement which have been discussed in the theoretical linguistics literature. We have also shown how a targeted error analysis can reveal substantial reductions in agreement errors, whose impact on quality no doubt exceeds what is suggested by the small BLEU score increases. As future work, we also plan to learn such patterns from large amounts of unlabelled data and use models learned thus to rank paraphrases.

Acknowledgements

This work was supported in part by NSF grant IIS-0812297 and by an allocation of computing time from the Ohio Supercomputer Center. Our thanks also to Robert Levine and the anonymous reviewers for helpful comments and discussion.

References

Baldridge, Jason and Geert-Jan Kruijff. 2002. Coupling CCG and Hybrid Logic Dependency Semantics. In *Proc. ACL-02*.

Baldridge, Jason. 2002. *Lexically Specified Derivational Control in Combinatory Categorial Grammar*. Ph.D. thesis, University of Edinburgh.

Bangalore, Srinivas and Owen Rambow. 2000. Exploiting a probabilistic hierarchical model for generation. In *Proc. COLING-00*.

Boxwell, Stephen and Michael White. 2008. Projecting Propbank roles onto the CCGbank. In *Proc. LREC-08*.

Cahill, Aoife and Josef van Genabith. 2006. Robust PCFG-based generation using automatically acquired LFG approximations. In *Proc. COLING-ACL '06*.

Clark, Stephen and James R. Curran. 2007. Wide-Coverage Efficient Statistical Parsing with CCG and Log-Linear Models. *Computational Linguistics*, 33(4):493–552.

Espinosa, Dominic, Michael White, and Dennis Mehay. 2008. Hypertagging: Supertagging for surface realization with CCG. In *Proc. ACL-08: HLT*.

Guo, Yuqing, Josef van Genabith, and Haifeng Wang. 2008. Dependency-based n-gram models for general purpose sentence realisation. In *Proc. COLING-08*.

Hockenmaier, Julia and Mark Steedman. 2007. CCGbank: A Corpus of CCG Derivations and Dependency Structures Extracted from the Penn Treebank. *Computational Linguistics*, 33(3):355–396.

Hogan, Deirdre, Conor Cafferkey, Aoife Cahill, and Josef van Genabith. 2007. Exploiting multi-word units in history-based probabilistic generation. In *Proc. EMNLP-CoNLL*.

Kathol, Andreas. 1999. Agreement and the Syntax-Morphology Interface in HPSG. In Levine, Robert D. and Georgia M. Green, editors, *Studies in Contemporary Phrase Structure Grammar*, pages 223–274. Cambridge University Press, Cambridge.

Kim, Jong-Bok. 2004. Hybrid Agreement in English. *Linguistics*, 42(6):1105–1128.

Nakanishi, Hiroko, Yusuke Miyao, and Jun'ichi Tsujii. 2005. Probabilistic methods for disambiguation of an HPSG-based chart generator. In *Proc. IWPT-05*.

Palmer, Martha, Dan Gildea, and Paul Kingsbury. 2005. The proposition bank: A corpus annotated with semantic roles. *Computational Linguistics*, 31(1).

Pollard, Carl and Ivan Sag. 1994. *Head-Driven Phrase Structure Grammar*. University Of Chicago Press.

Roark, Brian, Murat Saraclar, Michael Collins, and Mark Johnson. 2004. Discriminative language modeling with conditional random fields and the perceptron algorithm. In *Proc. ACL-04*.

Steedman, Mark. 2000. *The syntactic process*. MIT Press, Cambridge, MA, USA.

Velldal, Erik and Stephan Oepen. 2005. Maximum entropy models for realization ranking. In *Proc. MT Summit X*.

Weischedel, Ralph and Ada Brunstein. 2005. BBN pronoun coreference and entity type corpus. Technical report, BBN.

White, Michael and Rajakrishnan Rajkumar. 2008. A more precise analysis of punctuation for broad-coverage surface realization with CCG. In *Proc. of the Workshop on Grammar Engineering Across Frameworks (GEAF08)*.

White, Michael and Rajakrishnan Rajkumar. 2009. Perceptron reranking for CCG realization. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 410–419, Singapore, August. Association for Computational Linguistics.

White, Michael. 2006. Efficient Realization of Coordinate Structures in Combinatory Categorial Grammar. *Research on Language and Computation*, 4(1):39–75.

Web-based and combined language models: a case study on noun compound identification

Carlos Ramisch[†*] **Aline Villavicencio**[*] **Christian Boitet**[†]
[†] GETALP – Laboratory of Informatics of Grenoble, University of Grenoble
[*] Institute of Informatics, Federal University of Rio Grande do Sul
{ceramisch,avillavicencio}@inf.ufrgs.br Christian.Boitet@imag.fr

Abstract

This paper looks at the *web* as a corpus and at the effects of using *web* counts to model language, particularly when we consider them as a domain-specific versus a general-purpose resource. We first compare three vocabularies that were ranked according to frequencies drawn from general-purpose, specialised and *web* corpora. Then, we look at methods to combine heterogeneous corpora and evaluate the individual and combined counts in the automatic extraction of noun compounds from English general-purpose and specialised texts. Better *n*-gram counts can help improve the performance of empirical NLP systems that rely on *n*-gram language models.

1 Introduction

Corpora have been extensively employed in several NLP tasks as the basis for automatically learning models for language analysis and generation. In theory, *data-driven* (*empirical* or *statistical*) approaches are well suited to take intrinsic characteristics of human language into account. In practice, external factors also determine to what extent they will be popular and/or effective for a given task, so that they have shown different performances according to the availability of corpora, to the linguistic complexity of the task, etc.

An essential component of most empirical systems is the *language model* (LM) and, in particular, *n-gram language models*. It is the LM that tells the system how likely a word or *n*-gram is in that language, based on the counts obtained from corpora. However, corpora represent a sample of a language and will be sparse, i.e. certain words or expressions will not occur. One alternative to minimise the negative effects of data sparseness and account for the probability of out-of-vocabulary words is to use discounting techniques, where a constant probability mass is discounted from each *n*-gram and assigned to unseen *n*-grams. Another strategy is to estimate the probability of an unseen *n*-gram by backing off to the probability of the smaller *n*-grams that compose it.

In recent years, there has also been some effort in using the *web* to overcome data sparseness, given that the *web* is several orders of magnitude larger than any available corpus. However, it is not straightforward to decide whether (a) it is better to use the *web* than a standard corpus for a given task or not, and (b) whether corpus and *web* counts should be combined and how this should be done (e.g. using interpolation or back-off techniques). As a consequence there is a strong need for better understanding of the impacts of *web* frequencies in NLP systems and tasks.

More reliable ways of combining word counts could improve the quality of empirical NLP systems. Thus, in this paper we discuss *web*-based word frequency distributions (§ 2) and investigate to what extent "*web*-as-a-corpus" approaches can be employed in NLP tasks compared to standard corpora (§ 3). Then, we present the results of two experiments. First, we compare word counts drawn from general-purpose corpora, from specialised corpora and from the *web* (§ 4). Second, we propose several methods to combine data from heterogeneous corpora (§ 5), and evaluate their effectiveness in the context of a specific multiword

Coling 2008: Poster Volume, pages 1041–1049,
Beijing, August 2010

expression task: automatic noun compound identification. We close this paper with some conclusions and future work (§ 6).

2 The *web* as a corpus

Conventional and, in particular, domain-specific corpora, are valuable resources which provide a closed-world environment where precise *n*-gram counts can be obtained. As they tend to be smaller than general purpose corpora, data sparseness can considerably hinder the results of statistical methods. For instance, in the biomedical Genia corpus (Ohta et al., 2002), 45% of the words occur only once (so-called *hapax legomena*), and this is a very poor basis for a statistical method to decide whether this is a significant event or just random noise.

One possible solution is to see the *web* as a very large corpus containing pages written in several languages and being representative of a large fraction of human knowledge. However, there are some differences between using regular corpora and the *web* as a corpus, as discussed by Kilgarriff (2003). One assumption, in particular, is that page counts can approximate word counts, so that the total number of pages is used as an estimator of the *n*-gram count, regardless of how many occurrences of the *n*-gram they contain.

This simple underlying assumption has been employed for several tasks. For example, Grefenstette (1999), in the context of example-based machine translation, uses *web* counts to decide which of a set of possible translations is the most natural one for a given sequence of words (e.g. *groupe de travail* as *work group* vs *labour collective*). Likewise, Keller and Lapata (2003) use the *web* to estimate the frequencies of unseen nominal bigrams, while Nicholson and Baldwin (2006) look at the interpretation of noun compounds based on the individual counts of the nouns and on the global count of the compound estimated from the *web* as a large corpus.

Villavicencio et al. (2007) show that the *web* and the British National Corpus (BNC) could be used interchangeably to identify general-purpose and type-independent multiword expressions. Lapata and Keller (2005) perform a careful and systematic evaluation of the *web* as a corpus in other general-purpose tasks both for analysis and generation, comparing it with a standard corpus (the BNC) and using two different techniques to combine them: linear interpolation and back-off. Their results show that, while *web* counts are not as effective for some tasks as standard counts, the combined counts can generate results, for most tasks, that are as good as the results produced by the best individual corpus between the BNC and the *web*. Nakov (2007) further investigates these tasks and finds that, for many of them, effective attribute selection can produce results that are at least comparable to those from the BNC using counts obtained from the *web*.

On the one hand, the *web* can minimise the problem of sparse data, helping distinguish rare from invalid cases. Moreover, a search engine allows access to ever increasing quantities of data, even for rare constructions and words, which counts are usually equated to the number of pages in which they occur. On the other hand, *n*-grams in the highest frequency ranges, such as the words *the*, *up* and *down*, are often assigned the estimated size of the *web*, uniformly. While this still gives an idea of their massive occurrence, it does not provide a finer grained distinction among them (e.g. in the BNC, *the*, *down* and *up* occur 6,187,267, 84,446 and 195,426 times, respectively, while in Yahoo! they all occur in 2,147,483,647 pages).

3 Standard vs *web* corpora

When we compare *n*-gram counts estimated from the *web* with counts taken from a well-formed standard corpus, we notice that *web* counts are "estimated" or "approximated" as page counts, whereas standard corpus counts are the exact number of occurrences of the *n*-gram. In this way, *web* counts are dependent on the particular search engine's algorithms and representations, and these may perform approximations to handle the large size of their indexing structures and procedures, such as ignoring punctuation and using stopword lists (Kilgarriff, 2007). This assumption, as well as the following discussion, are not valid for for controlled data sets derived from Web data, such

as the Google 1 trillion n-grams[1]. Thus, our results cannot be compared to those using this kind of data (Bergsma et al., 2009).

In data-driven techniques, some statistical measures are based on contingency tables, and the counts for each of the table cells can be straightforwardly computed from a standard corpus. However, this is not the case for the *web*, where the occurrences of an n-gram are not precisely calculated in relation to the occurrences of the $(n-1)$-grams composing it. For instance, the n-gram *the man* may appear in 200,000 pages, while the words *the* and *man* appear in respectively 1,000,000 and 200,000 pages, implying that the word *man* occurs with no other word than *the*[2].

In addition, the distribution of words in a standard corpus follows the well known Zipfian distribution (Baayen, 2001) while, in the *web*, it is very difficult to distinguish frequent words or n-grams as they are often estimated as the size of the *web*. For instance, the Yahoo! frequencies plotted in figure 1(a) are flattened in the upper part, giving the same page counts for more than 700 of the most frequent words. Another issue is the size of the corpus, which is an important information, often needed to compute frequencies from counts or to estimate probabilities in n-gram models. Unlike the size of a standard corpus, which is easily obtained, it is very difficult to estimate how many pages exist on the *web*, especially as this number is always increasing.

But perhaps the biggest advantage of the *web* is its availability, even for resource-poor languages and domains. It is a free, expanding and easily accessible resource that is representative of language use, in the sense that it contains a great variability of writing styles, text genres, language levels and knowledge domains.

4 Analysing n-gram frequencies

In this section, we describe an experiment to compare the probability distribution of the vocabulary of two corpora, Europarl (Koehn, 2005) and Genia (Ohta et al., 2002), that represent a sample of general-purpose and specialised English. In addition to both corpora, we also considered the counts from the *web* as a corpus, using Google and Yahoo! APIs, and these four corpora act as *n-gram count sources*. To do that, we preprocessed the data (§ 4.1), extracted the vocabularies from each corpus and calculated their counts in our four n-gram count sources (§ 4.2), analysing their rank plots to compare how each of these sources models general-purpose and specialised language (§ 4.3). The experiments described in this section were implemented in the `mwetoolkit` and are available at `http://sf.net/projects/mwetoolkit/`.

	V_{ep}	V_{genia}	V_{inter}
types	104,144	20,876	6,798
hapax	41,377	9,410	–
tokens	39,595,352	486,823	–

Table 1: Some characteristics of general vs domain-specific corpora.

4.1 Preprocessing

The Europarl corpus v3.0 (*ep*) contains transcriptions of the speeches held at the European Parliament, with more than 1.4M sentences and 39,595,352 words. The Genia corpus (*genia*) contains abstracts of scientific articles in biomedicine, with around 1.8K sentences and 486,823 words. These standard corpora were preprocessed in the following way:

1. conversion to XML, lemmatisation and POS tagging[3];
2. case homogenisation, based on the following criteria:
   - all-uppercase and mixed case words were normalised to their predominant form, if it accounts for at least 80% of the occurrences;
   - uppercase words at the beginning of sentences were lowercased;
   - other words were not modified.

[1] This dataset is released through LDC and is not freely available. Therefore, we do not consider it in our evaluation.

[2] In practice, this procedure can lead to negative counts.

[3] Genia contains manual POS tag annotation. Europarl was tagged using the TreeTagger (`www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger`).

This lowercasing algorithm helps to deal with the massive use of abbreviations, acronyms, named entities, and formulae found in specialised corpora, such as those containing biomedical (and other specialised) scientific articles.

For calculating arbitrary-sized n-grams in large textual corpora efficiently, we implemented a structure based on suffix arrays (Yamamoto and Church, 2001). While suffix trees are often used in LM tools, where n-grams have a fixed size, they are not fit for arbitrary length n-gram searches and can consume quite large amounts of memory to store all the node pointers. Suffix arrays, on the other hand, allow for arbitrary length n-grams to be counted in a time that is proportional to $\log(N)$, where N is the number of words (which is equivalent to the number of suffixes) in the corpus. Suffix arrays use a constant amount of memory proportional to N. In our implementation, where every word and every word position in the corpus are encoded as a 4-byte integer, it corresponds precisely to $4 \times 2 \times N$ plus the size of the vocabulary, which is generally very small if compared to N, given a typical token/type ratio. The construction of the suffix array takes $O(N \log_2 N)$ operations, due to a sorting step at the end of the process.

4.2 Vocabulary creation

After preprocessing, we extracted all the unigram surface forms (i.e. all words) from *ep* and from *genia*, generating two vocabularies, V_{ep} and V_{genia}, where the words are ranked in descending frequency order with respect to the corpus itself seen as a n-gram count source. Formally, we can model a *vocabulary* as a set V of words $v_i \in V$ taken from a corpus. A word *count* is the value $c(v_i) = n$ of a function that goes from words to natural numbers, $c : V \to \mathbb{N}$. Therefore, there is always an implicit word order relation $\leq_r$ in a vocabulary, that can be generated from V and c by using the order relation $\geq$ in $\mathbb{N}$[4]. Thus, a *rank* is defined as a partially-ordered set formed by a vocabulary–word order pair relation: $\langle V, \leq_r \rangle$.

Table 1 summarises some measures of the extracted vocabularies, where V_{inter} denotes the intersection of V_{ep} and V_{genia}. Notice that V_{inter} contains considerably less entries than the smallest vocabulary (V_{genia}). This shows to what extent both types of text differ and how important it is to use the correct techniques when working with domain-specific data in empirical approaches. The table also shows the number of hapax legomena (i.e. words that occur only once) in each corpus, and in this aspect both corpora are similar[5]. It also shows how sparseness affects language, since a vocabulary that is 400% bigger has only 5% less hapax legomena.

n-gram	genia	ep	google	yahoo
642	1	4	8090K	220M
African	2	2028	15400K	916M
fatty	16	22	2550K	59700K
medicine	4	643	21900K	934M
Mac	15	3	34500K	1910M
SH2	27	1	113K	3270K
advances	4	646	6200K	173M
thereby	29	2370	8210K	145M

Table 2: Distribution of some words in V_{inter}.

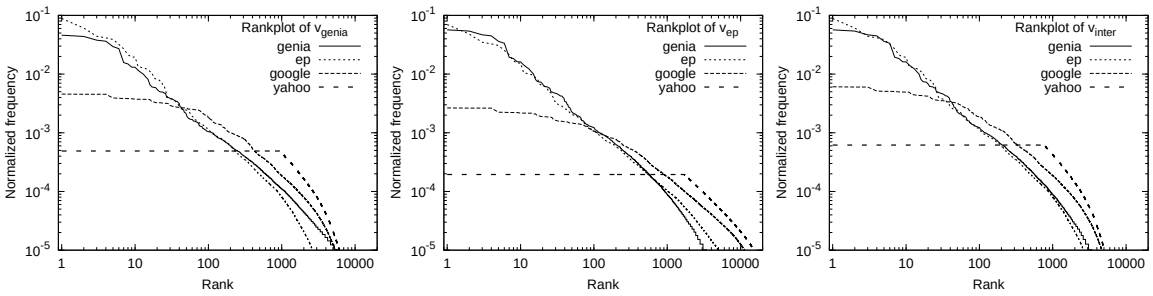
For each entry in each vocabulary, we obtained a count estimated from four different n-gram count sources: *ep*, *genia*, Google as a corpus (*google*) and Yahoo! as a corpus (*yahoo*). The latter were configured to return only results for pages in English. Table 2 shows an example of entries extracted from V_{inter}. Notice that there are no zeroes in columns *genia* and *ep*, since this vocabulary only contains words that occur at least once in these corpora. Also, some words like *Mac* and *SH2*, that are probably specialised terms, occur more in *genia* than in *ep* even if the latter is more than 80 times larger than the former.

4.3 Rank analyses

For each vocabulary, we want to estimate how similar the ranks generated by each of the four count sources are. Figure 1 shows the rank position (x) against the frequency (y) of words in V_{genia}, V_{ep} and V_{inter}, where each plotted point represents a rank position according to corpus fre-

[4] That is, $\forall v_1, v_2 \in V$, suppose $c(v_1) = n_1$ and $c(v_2) = n_2$, then $v_1 \leq_r v_2$ if and only if $n_1 \geq n_2$.

[5] The percentual difference in the proportion of hapax legomena can be explained by the fact that *genia* is much smaller than *ep*.



(a) Rank plot of V_{genia}. (b) Rank plot of V_{ep}. (c) Rank plot of V_{inter}.

Figure 1: Plot of normalised frequencies of vocabularies according to rank positions, log-log scale.

quencies and may correspond to several different words.[6] The four sources have similar shaped curves for each of the three vocabularies: *ep* and *genia* could be reasonably approximated by a linear regression curve (in the log-log domain). *google* and *yahoo* present Zipfian curves for low frequency ranges but have a flat line for higher frequencies, and the phenomenon seems consistent in all vocabularies and more intense on *yahoo*. This is related to the problem discussed in section 3 which is that *web*-based frequencies are not accurate to model common words because *web* counts correspond to page counts and not to word counts, and that a common word will probably appear dozens of times in a single page. Nonetheless, *google* seems more robust to this effect, and indeed *yahoo* returns exactly the same value (roughly 2 billion pages) for a large number of common words, producing the perfectly straight line in the rank plots. Moreover, the problem seems less serious in V_{inter}, but this could be due to its much smaller size. These results show that *google* is incapable of distinguishing among the top-100 words while *yahoo* is incapable of distinguishing among the top-1000 words, and this can be a serious drawback for *web*-based counts both in general-purpose and specialised NLP tasks.

The curves agree in a large portion of the frequency range, and the only interval in which *genia* and *ep* disagree is in lower frequencies (shown in the bottom right corner). This happens because general-purpose *ep* frequencies are much less accurate to model the specialised *genia* vocabulary, specially in low frequency ranges when sparseness becomes more marked (figure 1(a)), and vice-versa (figure 1(b)). This effect is minimised in figure 1(c), corresponding to V_{inter}.

Although both vocabularies present the same word frequency distributions, it does not mean that their ranks are similar for the four count sources. Tables 3 and 4 show the correlation scores for the compared count sources and for the two vocabularies, using Kendall's τ. The τ correlation index estimates the probability that a word pair in a given rank has the same *respective* position in another rank, in spite of the distance between the words[7].

In the two vocabularies, correlation is low, which indicates that the ranks tend to order words differently even if there are some similarities in terms of the shape of the frequency distribution. When we compare *genia* with *google* and with *yahoo*, we observe that *yahoo* is slightly less correlated with *genia* than *google*, probably because of its uniform count estimates for frequent words. However, both seem to be more similar to *genia* than *ep*.

A comparison of *ep* with *google* and with *yahoo* shows that *web* frequencies are much more similar to a general-purpose count source like *ep* than to a specialised source like *genia*. Additionally, both *yahoo* and *google* seem equally correlated to *ep*.

[6] Given the Zipfian behaviour of word probability distributions, a log-log scale was used to plot the curves.

[7] For all correlation values, $p < 0.001$ for the alternative hypothesis that τ is greater than 0.

	V_{genia}	V_{genia} top	V_{genia} middle	V_{genia} bottom
genia-ep	0.26	0.24	0.13	0.06
genia-google	0.28	0.24	0.18	0.09
genia-yahoo	0.27	0.22	0.17	0.09
ep-google	0.57	0.68	0.53	0.49
ep-yahoo	0.57	0.68	0.53	0.49
google-yahoo	0.90	0.90	0.89	0.89

Table 3: Kendall's τ for count sources in V_{genia}.

	V_{ep}	V_{ep} top	V_{ep} middle	V_{ep} bottom
genia-ep	0.26	0.36	0.07	0.04
genia-google	0.27	0.39	0.15	0.12
genia-yahoo	0.24	0.35	0.12	0.10
ep-google	0.40	0.45	0.22	0.09
ep-yahoo	0.38	0.44	0.20	0.08
google-yahoo	0.86	0.89	0.84	0.83

Table 4: Kendall's τ for count sources in V_{ep}.

Surprisingly, this correlation is higher for V_{genia} than for V_{ep}, as *web* frequencies and *ep* frequencies are more similar for a specialised vocabulary than for a general-purpose vocabulary. This could mean that the three perform similarly (poorly) at estimating frequencies for the biomedical vocabulary (V_{genia}) whereas they differ considerably at estimating general-purpose frequencies.

The correlation of the rank (first column) is also decomposed into the correlation for *top* words (more than 10 occurrences), *middle* words (10 to 3 occurrences) and *bottom* words (2 and 1 occurrences). Except for the pair *google-yahoo*, the correlation is much higher in the top portion of the vocabulary and is close to zero in the long tail. In spite of the logarithmic scale of the graphics in figure 1, that show the largest difference in the top part, the bottom part is actually the most irregular. The only exception is *ep* compared with the *web* count sources in V_{genia}: these two pairs do not present the high variability of the other compared pairs, and this means that using *ep* counts (general-purpose) to estimate *genia* counts (specialised) is similar to using *web* counts, independently of the position of the word in the rank. Counts from *google* and from *yahoo* are also very similar, specially if we also consider Spearman's ρ, that is very close to total correlation. Web ranks are also more similar for a specialised vocabulary than for a general-purpose one, providing further evidence for the hypothesis that the higher correlation is a consequence of both sources being poor frequency estimators. That is, for a given vocabulary, when *web* count sources are good estimators, they will be more distinct (e.g. having less zero frequencies).

5 Combining corpora frequencies

In our second experiment, the goal is to propose and to evaluate techniques for the combination of *n*-gram counts from heterogeneous sources. Therefore, we will use the insights about the vocabulary differences presented in the previous section. In this evaluation, we measure the impact of the suggested techniques in the identification of noun–noun compounds in corpora. Noun compounds are very frequent in general-purpose and specialised texts (e.g. *bus stop*, *European Union* and *gene activation*). We extract them automatically from *ep* and from *genia* using a standard method based on POS patterns and association measures (Evert and Krenn, 2005; Pecina, 2008; Ramisch et al., 2010).

5.1 Experimental setup

The evaluation task consists of, given a corpus of N words, extract all occurrences of adjacent pairs of nouns[8] and then rank them using a standard statistical measure that estimates the association strength between the two nouns. Analogously to the formalism adopted in section 4.2, we assume that, for each corpus, we generate a set NN containing n-grams $v_{1\ldots n} \in NN$[9] for which we obtain n-gram counts from four sources. The elements in NN are generated by comparing the POS pattern *noun–noun* against all the bigrams in the corpus and keeping only those pairs of adjacent words that match the pattern. The calculation of the association measure, considering a bigram $v_1 v_2$, is based on a contingency table which cells

[8] We ignore other types of compounds, e.g. adjective–noun pairs.

[9] We abbreviate a sequence $v_1 \ldots v_n$ as $v_{1\ldots n}$.

contain all possible outcomes $a_1a_2, a_i \in \{v_i, \neg v_i\}$. For *web*-based counts, we corrected up to 2% of them by forcing the frequency of a unigram to be at least equal to the frequency of the bigram in which it occurs. Such inconsistencies are incompatible with statistical approaches based on contingency table, as discussed in section 2.

The log-likelihood association measure (*LL*, alternatively called *expected mutual information*), estimates the difference between the observed table and the expected table under the assumption of independent events, where $E(a_1 \ldots a_n) = \frac{\prod_{i=1}^{n} c(a_i)}{N^{n-1}}$ is calculated using maximum likelihood:

$$LL(v_1v_2) = \sum_{a_1a_2} c(a_1a_2) \times \log_2 \frac{c(a_1a_2)}{E(a_1a_2)}$$

The evaluation of the *NN* lists is performed automatically with the help of existing noun compound dictionaries. The general-purpose gold standard, used to evaluate NN_{ep}, is composed of bigram noun compounds extracted from several resources: 6,212 entries from the Cambridge International Dictionary of English, 22,981 from Wordnet and 2,849 from the data sets of MWE 2008[10]. Those were merged into a single general-purpose gold standard that contains 28,622 bigram noun compounds. The specialised gold standard, used to evaluate NN_{genia}, is composed of 7,441 bigrams extracted from constituent annotation of the *genia* corpus with respect to concepts in the Genia ontology (Kim et al., 2006).

True positives (TPs) are the *n*-grams of *NN* that are contained in the respective gold standard, while *n*-grams that do not appear in the gold standard are considered false positives[11]. While this is a simplification that underestimates the performance of the method, it is appropriate for the purpose of this evaluation because we compare only the *mean average precision* (MAP) between two *NN* ranks, in order to verify whether improvements obtained by the combined frequencies are significant. Additionaly, MWEs are complex linguistic phenomena, and their annotation, specially in a domain corpus, is a difficult task that reaches low agreement rates, sometimes even for expert native speakers. Therefore, not only for theoretical reasons but also for practical reasons, we adopted an automatic evaluation procedure rathern than annotating the top candidates in the lists by hand.

Since the log-likelihood measure is a function that assigns a real value to each *n*-gram, there is a rank relation $\leq_r$ that will be used to calculate MAP as follows:

$$MAP(NN, \leq_r) = \frac{\sum_{v_{1\ldots n} \in NN} P(v_{1\ldots n}) \times p(v_{1\ldots n})}{|\text{TPs in } NN|},$$

where $p = 1$ if $v_{1\ldots n}$ is a TP, 0 else, and the precision $P(v_{1\ldots n})$ of a given *n*-gram corresponds to the number of TPs before $v_{1\ldots n}$ in $\langle NN, \leq_r \rangle$ over the total number of *n*-grams before $v_{1\ldots n}$ in $\langle NN, \leq_r \rangle$.

5.2 Combination heuristics

From the initial list of 176,552 lemmatised *n*-grams in NN_{ep} and 14,594 in NN_{genia}, we filtered out all hapax legomena in order to remove noise and avoid useless computations. Then, we counted the occurrences of v_1, v_2 and v_1v_2 in our four sources, and those were used to calculate the four *LL* values of *n*-grams in both lists. We also propose three heuristics to combine a set of m count sources c_1 through c_m into a single count source c_{comb}:

$$c_{comb}(v_{1\ldots n}) = \sum_{i=1}^{m} w_i(v_{1\ldots n}) \times c_i(v_{1\ldots n}),$$

where $w(v_{1\ldots n})$ is a function that assigns a weight between 0 and 1 for each count source according to the *n*-gram $v_{1\ldots n}$. Three different functions were used in our experiments: *uniform* linear interpolation assumes a constant and uniform weight $w(v_{1\ldots n}) = 1/m$ for all *n*-grams; *proportional* linear interpolation assumes a constant weight $w_i(v_{1\ldots n}) = ((\sum_{j=1}^{m} N_j) - N_i)/\sum_{j=1}^{m} N_j$ that is proportional to the inverse size of the corpus; and *back-off* uses the uniform interpolation of *web* frequencies whenever the *n*-gram count in the original corpus falls below a threshold (empirically defined as $\log_2(N/100,000)$).

[10] 420 entries provided by Timothy Baldwin, 2,169 entries provided by Su Nam Kim and 250 entries provided by Preslav Nakov, freely available at `http://multiword.sf.net/`

[11] In fact, nothing can be said about an *n*-gram that is not in a (limited-coverage) dictionary, further manual annotation would be necessary to asses its relevance.

MAP of rank	NN_{genia}	NN_{ep}
LL_{genia}	0.4400	0.0462
LL_{ep}	0.4351	0.0371
LL_{google}	0.4297	0.0532
LL_{yahoo}	0.4209	0.0508
$LL_{uniform}$	0.4254	0.0508
$LL_{proportional}$	0.4262	0.0520
$LL_{backoff}$	0.3719	0.0370

Table 5: Performance of compound extraction.

Table 5 shows that the performance of *backoff* is below all other techniques for both vocabularies, thus excluding it as a successful combination heuristic. The large difference between MAP scores for NN_{ep} and for NN_{genia} is explained by the relative size of the gold standards: while the general-purpose reference accounts for 16% of the size of the NN_{ep} set, the specialised reference has as many entries as 50% of NN_{genia}. Moreover, the former was created by joining heterogeneous resources while the latter was compiled by human annotators from the Genia corpus itself. The goal of our evaluation, however, is not to compare the difficulty of each task, but to compare the combination heuristics presented in each row of the table.

The best MAP for NN_{genia} was obtained with *genia*, that significantly outperforms all other sources except *ep*[12]. On the other hand, the use of *web*-based or interpolated counts in extracting specialised noun–noun compounds does not improve the performance of results based on sparse but reliable counts drawn from well-formed corpora. Nonetheless, the performance of *ep* in specialised extraction is surprising and could only be explained by some overlap between the corpora. Moreover, the interpolated counts are not significantly different from *google* counts, even if this corpus should have the weakest weight in *proportional* interpolation.

General-purpose compound extraction, however, benefits from the counts drawn from large corpora as *google* and *yahoo*. Indeed, the former significantly outperforms all other count sources, closely followed by *proportional* counts. In both vocabularies, *proportional* interpolation performs very similar to the best count source, but, strangely enough, it still does not outperform *google*. Further data inspection would be needed to explain these results for the interpolated combination and to try to shed some light on the reason why the *backoff* method performs so poorly.

[12]Significance was assessed through a standard one-tailed t test for equal sample sizes and variances, $\alpha = 0.005$.

6 Future perspectives

In this work, we presented a detailed evaluation of the use of *web* frequencies as estimators of corpus frequencies in general-purpose and specialised tasks, discussing some important aspects of corpus-based versus *web*-based *n*-gram frequencies. The results indicate that they are not only very distinct but they are so in different ways. The importance of domain-specific data for modelling a specialised vocabulary is discussed in terms of using *ep* to get V_{genia} counts. Furthermore, the *web* corpora were more similar to *genia* than to *ep*, which can be explained by the fact that "similar" is different from "good", i.e. they might be equally bad in modelling *genia* while they are distinctly better for *ep*.

We also proposed heuristics to combine count sources inspired by standard interpolation and back-off techniques. Results show that we cannot use *web*-based or combined counts to identify specialised noun compounds, since they do not help minimise data sparseness. However, general-purpose extraction is improved with the use of *web* counts instead of counts drawn from standard corpora.

Future work includes extending this research to other languages and domains in order to estimate how much of these results depend on the corpora sizes. Moreover, as current interpolation techniques usually combine two corpora, weights are estimated in a more or less ad hoc procedure (Lapata and Keller, 2005). Interpolating several corpora would need a more controlled learning technique to obtain optimal weights for each frequency function. Additionally, the evaluation shows that corpora perform differently according to the frequency range. This insight could be used to define weight functions for interpolation.

Acknowledgements

This research was partly supported by CNPq (Projects 479824/2009-6 and 309569/2009-5), FINEP and SEBRAE (COMUNICA project FINEP/SEBRAE 1194/07). Special thanks to Flávio Brun for his thorough work as volunteer proofreader.

References

Baayen, R. Harald. 2001. *Word Frequency Distributions*, volume 18 of *Text, Speech and Language Technology*. Springer.

Bergsma, Shane, Dekang Lin, and Randy Goebel. 2009. Web-scale N-gram models for lexical disambiguation. In Boutilier, Craig, editor, *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI 2009)*, pages 1507–1512, Pasadena, CA, USA, July.

Evert, Stefan and Brigitte Krenn. 2005. Using small random samples for the manual evaluation of statistical association measures. *Computer Speech & Language Special issue on Multiword Expression*, 19(4):450–466.

Grefenstette, Gregory. 1999. The World Wide Web as a resource for example-based machine translation tasks. In *Proceedings of the Twenty-First International Conference on Translating and the Computer*, London, UK, November. ASLIB.

Keller, Frank and Mirella Lapata. 2003. Using the web to obtain frequencies for unseen bigrams. *Computational Linguistics Special Issue on the Web as Corpus*, 29(3):459–484.

Kilgarriff, Adam and Gregory Grefenstette. 2003. Introduction to the special issue on the web as corpus. *Computational Linguistics Special Issue on the Web as Corpus*, 29(3):333–347.

Kilgarriff, Adam. 2007. Googleology is bad science. *Computational Linguistics*, 33(1):147–151.

Kim, Jin-Dong, Tomoko Ohta, Yuka Teteisi, and Jun'ichi Tsujii. 2006. GENIA ontology. Technical report, Tsujii Laboratory, University of Tokyo.

Koehn, Philipp. 2005. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of the Tenth Machine Translation Summit(MT Summit 2005)*, Phuket, Thailand, September. Asian-Pacific Association for Machine Translation.

Lapata, Mirella and Frank Keller. 2005. Web-based models for natural language processing. *ACM Transactions on Speech and Language Processing (TSLP)*, 2(1):1–31.

Nakov, Preslav. 2007. *Using the Web as an Implicit Training Set: Application to Noun Compound Syntax and Semantics*. Ph.D. thesis, EECS Department, University of California, Berkeley, CA, USA.

Nicholson, Jeremy and Timothy Baldwin. 2006. Interpretation of compound nominalisations using corpus and web statistics. In Moirón, Begoña Villada, Aline Villavicencio, Diana McCarthy, Stefan Evert, and Suzanne Stevenson, editors, *Proceedings of the COLING/ACL Workshop on Multiword Expressions: Identifying and Exploiting Underlying Properties (MWE 2006)*, pages 54–61, Sidney, Australia, July. Association for Computational Linguistics.

Ohta, Tomoko, Yuka Tateishi, and Jin-Dong Kim. 2002. The GENIA corpus: an annotated research abstract corpus in molecular biology domain. In *Proceedings of the Second Human Language Technology Conference (HLT 2002)*, pages 82–86, San Diego, CA, USA, March. Morgan Kaufmann Publishers.

Pecina, Pavel. 2008. Reference data for czech collocation extraction. In Gregoire, Nicole, Stefan Evert, and Brigitte Krenn, editors, *Proceedings of the LREC Workshop Towards a Shared Task for Multiword Expressions (MWE 2008)*, pages 11–14, Marrakech, Morocco, June.

Ramisch, Carlos, Aline Villavicencio, and Christian Boitet. 2010. mwetoolkit: a framework for multiword expression identification. In Calzolari, Nicoletta, Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odjik, Stelios Piperidis, Mike Rosner, and Daniel Tapias, editors, *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC 2010)*, Valetta, Malta, May. European Language Resources Association.

Villavicencio, Aline, Valia Kordoni, Yi Zhang, Marco Idiart, and Carlos Ramisch. 2007. Validation and evaluation of automatically acquired multiword expressions for grammar engineering. In Eisner, Jason, editor, *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL 2007)*, pages 1034–1043, Prague, Czech Republic, June. Association for Computational Linguistics.

Yamamoto, Mikio and Kenneth W. Church. 2001. Using suffix arrays to compute term frequency and document frequency for all substrings in a corpus. *Computational Linguistics*, 27(1):1–30.

Streaming Cross Document Entity Coreference Resolution

Delip Rao and **Paul McNamee** and **Mark Dredze**
Human Language Technology Center of Excellence
Center for Language and Speech Processing
Johns Hopkins University
delip,mcnamee,mdredze@jhu.edu

Abstract

Previous research in cross-document entity coreference has generally been restricted to the offline scenario where the set of documents is provided in advance. As a consequence, the dominant approach is based on greedy agglomerative clustering techniques that utilize pairwise vector comparisons and thus require $O(n^2)$ space and time. In this paper we explore identifying coreferent entity mentions across documents in high-volume streaming text, including methods for utilizing orthographic and contextual information. We test our methods using several corpora to quantitatively measure both the efficacy and scalability of our streaming approach. We show that our approach scales to at least an order of magnitude larger data than previous reported methods.

1 Introduction

A key capability for successful information extraction, topic detection and tracking, and question answering is the ability to identify equivalence classes of entity mentions. An entity is a real-world person, place, organization, or object, such as the person who serves as the 44th president of the United States. An entity mention is a string which refers to such an entity, such as "Barack Hussein Obama", "Senator Obama" or "President Obama". The goal of coreference resolution is to identify and connect all textual entity mentions that refer to the same entity.

The first step towards this goal is to identify all references within the same document, or *within document coreference resolution*. A document often has a leading canonical reference to the entity ("Barack Obama") followed by additional expressions for the same entity ("President Obama.") An intra-document coreference system must first identify each reference, often relying on named entity recognition, and then decide if these references refer to a single individual or multiple entities, creating a *coreference chain* for each unique entity. Feature representations include surface form similarity, lexical context of mentions, position in the document and distance between references. A variety of statistical learning methods have been applied to this problem, including use of decision trees (Soon et al., 2001; Ng and Cardie, 2002), graph partitioning (Nicolae and Nicolae, 2006), maximum-entropy models (Luo et al., 2004), and conditional random fields (Choi and Cardie, 2007).

Given pre-processed documents, in which entities have been identified and entity mentions have been linked into chains, we seek to identify across an entire document collection all chains that refer to the same entity. This task is called cross document coreference resolution (CDCR). Several of the challenges associated with CDCR differ from the within document task. For example, it is unlikely that the same document will discuss John Phillips the American football player and John Phillips the musician, but it is quite probable that documents discussing each will appear in the same collection. Therefore, while matching entities with the same mention string can work well for within document coreference, more sophisticated approaches are necessary for the cross document scenario where a *one-entity-per-name* assumption is unreasonable.

One of the most common approaches to both within document and cross document coreference resolution has been based on agglomerative clustering, where vectors might be bag-of-word contexts (Bagga and Baldwin, 1998; Mann and

Coling 2008: Poster Volume, pages 1050–1058,
Beijing, August 2010

Yarowsky, 2003; Gooi and Allan, 2004; Chen and Martin, 2007). These algorithms creates a $O(n^2)$ dependence in the number of *mentions* – for within document – and *documents* – for cross document. This is a reasonable limitation for within document, since the number of references will certainly be small; we are unlikely to encounter a document with millions of references. In contrast to the small n encountered within a document, we fully expect to run a CDCR system on hundreds of thousands or millions of documents. Most previous approaches cannot handle collections of this size.

In this work, we present a new method for cross document coreference resolution that scales to very large corpora. Our algorithm operates in a *streaming* setting, in which documents are processed one at a time and only a single time. This creates a linear ($O(n)$) dependence on the number of documents in the collection, allowing us to scale to millions of documents and millions of unique entities. Our algorithm uses streaming clustering with common coreference similarity computations to achieve large scale. Furthermore, our method is designed to support both name disambiguation and name variation.

In the next section, we give a survey of related work. In Section 3 we detail our streaming setup, giving a description of the streaming algorithm and presenting efficient techniques for representing clusters over streams and for computing similarity. Section 4 describes the data sets on which we evaluate our methods and presents results. We conclude with a discussion and description of ongoing work.

2 Related Work

Traditional approaches to cross document coreference resolution have first constructed a vector space representation derived from local (or global) contexts of entity mentions in documents and then performed some form of clustering on these vectors. This is a simple extension of Firth's distributional hypothesis applied to entities (Firth, 1957). We describe some of the seminal work in this area.

Some of the earliest work in CDCR was by Bagga and Baldwin (1998). Key contributions of their research include: promotion of a set-theoretic evaluation measure, *B-CUBED*; introduction of a data set based on 197 New York Times articles which mention a person named *John Smith*; and, use of TF/IDF weighted vectors and cosine similarity in single-link greedy agglomerative clustering.

Mann and Yarowsky (2003) extended Bagga and Baldwin's work and contributed several innovations, including: use of biographical attributes (*e.g.,* year of birth, occupation), and evaluation using *pseudonames*. Pseudonames are sets of artificially conflated names that are used as an efficient method for producing a set of gold-standard disambiguations.[1] Mann and Yarowsky used 4 pairs of conflated names in their evaluation. Their system did not perform as well on named entities with little available biographic information.

Gooi and Allan (2004) expanded on the use of pseudonames by semi-automatically creating a much larger evaluation set, which they called the 'Person-X' corpus. They relied on automated named-entity tagging and domain-focused text retrieval. This data consisted of 34,404 documents where a single person mention in each document was rewritten as 'Person X'. Besides their novel construction of a large-scale resource, they investigated several minor variations in clustering, namely (a) use of Kullback-Leibler divergence as a distance measure, (b) use of 55-word snippets around entity mentions (vs. entire documents or extracted sentences), and (c) scoring clusters using average-link instead of single- or complete-link.

Finally, in more recent work, Chen and Martin (2007) explore the CDCR task in both English and Chinese. Their work focuses on use of both local, and document-level noun-phrases as features in their vector-space representation.

There have been a number of open evaluations of CDCR systems. For example, the Web People Search (WePS) workshops (Artiles et al., 2008) have created a task for disambiguating personal names from HTML pages. A set of ambiguous names is chosen and each is submitted to a popular web search engine. The top 100 pages are then manually clustered.We discuss several other data

[1] See Sanderson (2000) for use of this technique in word sense disambiguation.

sets in Section 4.[2]

All of the papers mentioned above focus on disambiguating personal names. In contrast, our system can also handle organizations and locations. Also, as was mentioned earlier, we are committed to a scenario where documents are presented in sequence and entities must be disambiguated instantly, without the benefit of observing the entire corpus. We believe that such a system is better suited to highly dynamic environments such as daily news feeds, blogs, and tweets. Additionally, a streaming system exposes a set of known entity clusters after each document is processed instead of waiting until the end of the stream.

3 Approach

Our cross document coreference resolution system relies on a streaming clustering algorithm and efficient calculation of similarity scores. We assume that we receive a stream of coreference chains, along with entity types, as they are extracted from documents. We use SERIF (Ramshaw and Weischedel, 2005), a state of the art document analysis system which performs intra-document coreference resolution. BBN developed SERIF to address information extraction tasks in the ACE program and it is further described in Pradhan et al. (2007).

Each unique entity is represented by an entity cluster c, comprised of entity chains from many documents that refer to the same entity. Given an entity coreference chain e, we identify the best known entity cluster c. If a suitable entity cluster is not found, a new entity cluster is formed.

An entity cluster is selected for a given coreference chain using several similarity scores, including document context, predicted entity type, and orthographic similarity between the entity mention and previously discovered references in the entity cluster. An efficient implementation of the similarity score allows the system to identify the top k most likely mentions without considering all m entity clusters. The final output of our system is a collection of entity clusters, each containing a list of coreference chains and their documents. Additionally, due to its streaming nature, the system can be examined at any time to produce this information based on only the documents that have been processed thus far.

In the next sections, we describe both the clustering algorithm and efficient computation of the entity similarity scores.

3.1 Clustering Algorithm

We use a streaming clustering algorithm to create entity clusters as follows. We observe a set of points from a potentially infinite set $\mathcal{X}$, one at a time, and would like to maintain a fixed number of clusters while minimizing the maximum *cluster radius*, defined as the radius of the smallest ball containing all points of the cluster. This setup is well known in the theory and information retrieval community and is referred to as the dynamic clustering problem (Can and Ozkarahan, 1987).

Others have attempted to use an incremental clustering approach, such as Gooi and Allan (2004) (who eventually prefer a hierarchical clustering approach), and Luo et al. (2004), who use a Bell tree approach for incrementally clustering within document entity mentions. Our work closely follows the *Doubling Algorithm* of Charikar et al. (1997), which has better performance guarantees for streaming data. Streaming clustering means potentially linear performance in the number of observations since each document need only be examined a single time, as opposed to the quadratic cost of agglomerative clustering.[3]

The Doubling Algorithm consists of two stages: update and merge. Update adds points to existing clusters or creates new clusters while merge combines clusters to prevent the clusters from exceeding a fixed limit. New clusters are created according to a threshold set using development data. We selected a threshold of 0.5 since it worked well in preliminary experiments. Since the number of entities grows with time, we have skipped the merge step in our initial experiments so as not to limit cluster growth.

We use a dynamic caching scheme which backs the actual clusters in a disk based index, but re-

[2]We preferred other data sets to the WePS data in our evaluation because it is not easily placed in temporal order.

[3]It is possible to implement hierarchical agglomerative clustering in $O(n \log m)$ time where n is the number of points and m in the number of clusters. However this is still superlinear and expensive in situations where m continually increases like in streaming coreference resolution.

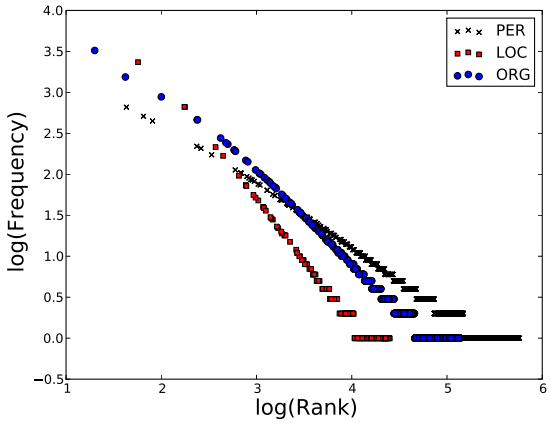


Figure 1: Frequency vs. rank for 567k people, 136k organizations, and 25k locations in the New York Times Annotated Corpus (Sandhaus, 2008).

tains basic cluster information in memory (see below). Doing so improves paging performance as observed in Omiecinski and Scheuermann (1984). Motivated by the Zipfian distribution of named entities in news sources (Figure 1), we organize our cluster store using an LRU policy, which facilitates easy access to named entities that were observed in the recent past. We obtain additional performance gains by hashing the clusters based on the constituent mention string (details below). This allows us to quickly retrieve a small but related number of clusters, k. It is always the case that $k << m$, the current number of clusters.

3.2 Candidate Cluster Selection

As part of any clustering algorithm, each new item must be compared against current clusters. As we see more documents, the number of unique clusters (entities) grows. Therefore, we need efficient methods to select candidate clusters.

To select the top candidate clusters, we obtain those that have high orthographic similarity with the head name mention in the coreference chain e. We compute this similarity using the dice score on either word unigrams or character skip bigrams. For each entity mention string associated with a cluster c, we generate all possible n-grams using one of the above two policies. We then index the cluster by each of its n-grams in a hash maintained in memory. In addition, we keep the number of n-grams generated for each cluster.

When given a new head mention e for a coreference chain, we generate all of the n-grams and look up clusters that contain these n-grams using the hash. We then compute the dice score:

$$\text{dice}(e, c) = \frac{|\{\text{ngram}(e)\} \cap \{\text{ngram}(c)\}|}{|\{\text{ngram}(e)\} \cup \{\text{ngram}(c)\}|},$$

where $\{\text{ngram}(e)\}$ are the set of n-grams in entity mention e and $\{\text{ngram}(c)\}$ are the set of n-grams for all entity mentions in cluster c. Note that we can calculate the numerator (the intersection) by looking up the n-grams of e in the hash and counting matches with c. The denominator is equivalent to the number of n-grams unique to e and to c plus the number that are shared. The number that are shared is the intersection. The number unique to e is the total number of n-grams in e minus the intersection. The final term, the number unique to c, is computed by taking the total number of n-grams in c (a single integer stored in memory) minus the intersection.

Through this strategy, we can select only those clusters that have the highest orthographic similarity to e without requiring the cluster contents, which may not be stored in memory. In our experiments, we evaluate settings where we select all candidates with non-zero score and a pruned set of the top k dice score candidates. We also include in the n-gram list known aliases to facilitate orthographically dissimilar, but reasonable matches (*e.g.,* IBM or 'Big Blue' for 'International Business Machines, Inc.').[4]

For further efficiency, we keep separate caches for each named entity type.[5] We then select the appropriate cache based on the automatically determined type of the named entity provided by the named entity tagger, which also prevents spurious matches of non-matching entity types.

3.3 Similarity Metric

After filtering by orthographic information to quickly obtain a small set of candidate clusters, a full similarity score is computed for the current

[4]We generated alias lists for entities from Freebase.

[5]Persons (PER), organizations (ORG), and locations (LOC).

entity coreference chain and each retrieved candidate cluster. These computations require information about each cluster, so the cluster's sufficient statistics are loaded using the LRU cache described above.

We define several similarity metrics between coreference chains and clusters to deal with both name variation and disambiguation. For name variation, we define an orthographic similarity metric to match similar entity mention strings. As before, we use word unigrams and character skip bigrams. For each of these methods, we compute a similarity score as dice(e, c) and select the highest scoring cluster.

To address name disambiguation, we use two types of context from the document. First, we use *lexical* features represented as TF/IDF weighted vectors. Second, we consider *topic* features, in which each word in a document is replaced with the topic inferred from a topic model. This yields a distribution over topics for a given document. We use an LDA (Blei et al., 2003) model trained on the New York Times Annotated Corpus (Sandhaus, 2008). We note that LDA can be computed over streams (Yao et al., 2009).

To compare context vectors we use cosine similarity, where the cluster vector is the average of all document vectors assigned to the cluster. Note that the filtering step in Section 3.2 returns only those candidates with some orthographic similarity with the coreference chain, so a similarity metric that uses context only is still restricted to orthographically similar entities.

Finally, we consider a combination of orthographic and context similarity as a linear combination of the two metrics as:

$$\text{score}(e, c) = \alpha \, \text{dice}(e, c) + (1 - \alpha)\text{cosine}(e, c) \; .$$

We set $\alpha = 0.8$ based on initial experiments.

4 Evaluation

We used several corpora to evaluate our methods, including two data sets commonly used in the coreference community. We also created a new test set using artificially conflated names. And finally to test scalability, we ran our algorithm over a large text collection that, while it did not have ground truth entity clusters, was useful for computing other performance statistics. Properties for each data set are given in Table 1.

Attribute	*smith*	*nytac*	*ace08*	*kbp09*
Total Documents	197	1.85M	10k	1.2M
Annotated Docs	197	19,360	415	**
Annotated Entities	35	200	3,943	**

Table 1: Data sets used in our experiments. For the *kbp09* data we did not have annotations.

4.1 John Smith corpus

Bagga and Baldwin (1998) evaluated their disambiguation system on a set of 197 articles from the New York Times that mention a person named 'John Smith'. This data exhibits no name variants and is strictly a disambiguation task. We include this data (*smith*) to allow comparison to previous work.

4.2 NYTAC Pseudo-name corpus

To study the effects of word sense ambiguity and disambiguation several researchers have artificially conflated dissimilar words together and then attempted to disambiguate them (Sanderson, 2000). The obvious advantage is cheaply obtained ground truth for disambiguation.

The same trick has also been employed in person name disambiguation (Mann and Yarowsky, 2003; Gooi and Allan, 2004). We adopt the same method on a somewhat larger scale using annotations from the New York Times Annotated Corpus (*NYTAC*) (Sandhaus, 2008), which annotates documents based on whether or not they mention an entity. The NYTAC data contains documents from 20 years of the New York Times and contains rich metadata and document-level annotations that indicate when an entity is mentioned in the document using a standard lexicon of entities. (Note that mention strings are not tagged.) Using these annotations we created a set of 100 pairs of conflated person names.

The names were selected to be medium frequency (*i.e.,* occurring in between 50 and 200 articles) and each pair matches in gender. The first 50 pairs are for names that are topically similar, for example, Tim Robbins and Tom Hanks (both actors); Barbara Boxer and Olympia Snowe (both

	smith			*nytac*			*ace08*		
Approach	P	R	F	P	R	F	P	R	F
Baseline	1.000	0.178	0.302	1.000	0.010	0.020	1.000	0.569	0.725
ExactMatch	0.233	1.000	0.377	0.563	0.897	0.692	0.977	0.697	0.814
Ortho	0.603	0.629	0.616	0.611	0.784	0.687	0.975	0.694	0.811
BoW	0.956	0.367	0.530	0.930	0.249	0.349	0.989	0.589	0.738
Topic	0.847	0.592	0.697	0.815	0.244	0.363	0.983	0.605	0.750
Ortho+BoW	0.603	0.634	0.618	0.801	0.601	0.686	0.976	0.691	0.809
Ortho+Topic	0.603	0.634	0.618	0.800	0.591	0.680	0.975	0.704	0.819

Table 2: Best B^3 performance on the *smith*, *nytac*, and *ace08* test sets.

US politicians). We imagined that this would be a more challenging subset because of presumed lexical overlap. The second set of 50 name pairs were arbitrarily conflated. We sub-selected the data to ensure that no two entities in our collection co-occur in the same document and this left us with 19,360 documents for which ground-truth was known. In each document we rewrote the conflated name mentions using a single gender-neutral name; any middle initials or names were discarded.

4.3 ACE 2008 corpus

The NIST ACE 2008 (*ace08*) evaluation studied several related technologies for information extraction, including named-entity recognition, relation extraction, and cross-document coreference for person names in both English and Arabic. Approximately 10,000 documents from several genres (predominantly newswire) were given to participants, who were expected to cluster person and organization entities across the entire collection. However, only a selected set of about 400 documents were annotated and used to evaluate system performance. Baron and Freedman (2008) describe their work in this evaluation, which included a separate task for within-document coreference.

4.4 TAC-KBP 2009 corpus

The NIST TAC 2009 Knowledge Base Population track (*kbp09*) (McNamee and Dang, 2009) conducted an evaluation of a system's ability to link entity mentions to corresponding Wikipedia-derived knowledge base nodes. The TAC-KBP task focused on ambiguous person, organization, and geo-political entities mentioned in newswire, and required systems to cope with name variation (*e.g.,* "Osama Bin Laden" / "Usama Bin Laden" or "Mark Twain" / "Samuel Clemens") as well as name disambiguation. Furthermore, the task required detection of when no appropriate KB entry exists, which is a departure from the conventional disambiguation problem. The collection contains over 1.2 million documents, primarily newswire. Wikipedia was used as a surrogate knowledge base, and it has been used in several previous studies (*e.g.,* Cucerzan (2007)). This task is closely related to CDCR, as mentions that are aligned to the same knowledge base entry create a coreference cluster. However, there are no actual CDCR annotations for this corpus, though we used it nonetheles as a benchmark corpus to evaluate speed and to demonstrate scalability.

5 Discussion

5.1 Accuracy

In Table 2 we report cross document coreference resolution performance for a variety of experimental conditions using the B^3 method, which includes precision, recall, and calculated $F_{\beta=1}$ values. For each of the three evaluation corpora (*smith*, *nytac*, and *ace08*) we report values for two baseline methods and for similarity metrics using different types of features. The first baseline, called *Baseline*, places each coreference chain in its own cluster while the second baseline, called *ExactMatch*, merges all mentions that match exactly orthographically into the same cluster.

Use of name similarity scores as features (in addition to their use for candidate cluster selection) is indicated by rows labeled *Ortho*. Use of lexical features is indicated by *BoW* and use of topic model features by *Topic*.

Using topic models as features was more helpful than lexical contexts on the *smith* corpus.

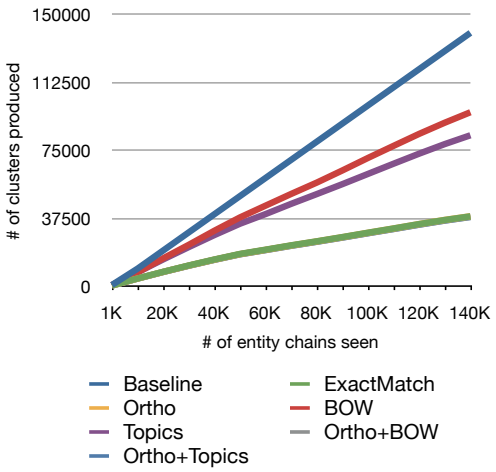


Figure 2: Number of clusters produced vs. number of entity chains observed in the stream. Number of entity chains is proportional to the number of documents.

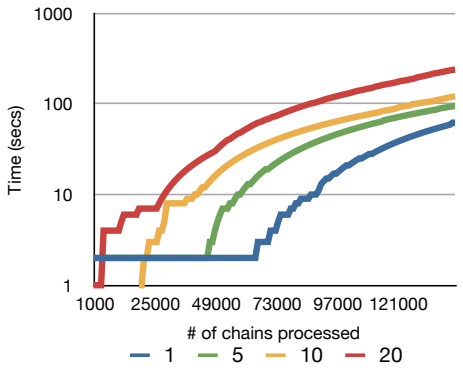


Figure 3: Elapsed processing time as a function of bounding the number of candidate clusters considered for an entity. When fewer candidates are considered, clustering decisions can be made much faster.

When used alone *topic* beats *BoW*, but in combination with the *ortho* features performance is equivalent. For both *nytac* and *ace08* heavy reliance on orthographic similarity proved hard to beat. On the *ace08* corpus Baron and Freedman (2008) report B^3 F-scores of 83.2 for persons and 67.8 for organizations, and our streaming results appear to be comparable to their offline method.

The cluster growth induced by the various measures can be seen in Figure 2. The two baseline methods, *Baseline* and *ExactMatch*, provide bounds on the cluster growth with all other methods falling in between.

5.2 Hashing Strategies for Candidate Selection

Table 3 contains B^3 F-scores when different hashing strategies are employed for candidate selection. The trend appears to be that stricter matching outperforms fuzzier matching; full mentions tended to beat words, which beat use of the character bigrams. This agrees with the results described in the previous section, which show heavy reliance on orthographic similarity.

5.3 Timing Results

Figure 3 shows how processing time increases with the number of entities observed in the *ace08* document stream. We experimented with using an upper bound on the number of candidate clusters to consider for an entity.

Figure 4 compares the efficiency of using three different methods for candidate cluster identification. The most restrictive hashing strategy, using exact mention strings, is the most efficient, followed by the use of words, then the use of character skip bigrams. This makes intuitive sense – the strictest matching reduces the number of candidate clusters that have to be considered when processing an entity.[6]

The *ace08* corpus contained over 10,000 documents and is one of the largest CDCR test sets. In Figure 5 we show how processing time grows when processing the *kbp09* corpus. Doubling the number of entities processed increases the runtime by about a factor of 5. The curve is not linear due to the increasing number of entity cluster's that must be considered. Future work will examine how to keep the number of clusters considered constant over time, such as ignoring older entities.

6 Conclusion

We have presented a new streaming cross document coreference resolution system. Our approach is substantially faster than previous sys-

[6]In the limit, if names were unique, hashing on strings would completely solve the CDCR problem and processing an entity would be O(1)

	smith			*nytac*			*ace08*		
Approach	bigrams	words	mention	bigrams	words	mention	bigrams	words	mention
Ortho	0.382	0.553	0.616	0.120	0.695	0.687	0.540	0.797	0.811
BoW	0.480	0.530	0.467	0.344	0.339	0.349	0.551	0.700	0.738
Topic	0.697	0.661	0.579	0.071	0.620	0.363	0.544	0.685	0.750
Ortho+BoW	0.389	0.554	0.618	0.340	0.691	0.686	0.519	0.783	0.809
Ortho+Topic	0.398	0.555	0.618	0.120	0.477	0.680	0.520	0.776	0.819

Table 3: B^3 F-scores using different hashing strategies for candidate selection. Name/cluster similarity could be based on character skip bigrams, words appear in names, or exact matching of mention.

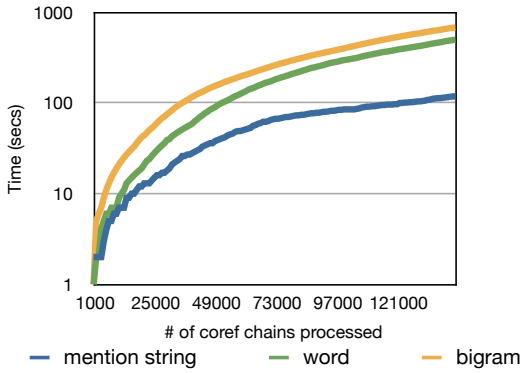


Figure 4: Comparison of three hashing strategies for identifying candidate clusters for a given entity. The more restrictive strategies lead to faster processing as fewer candidates are considered.

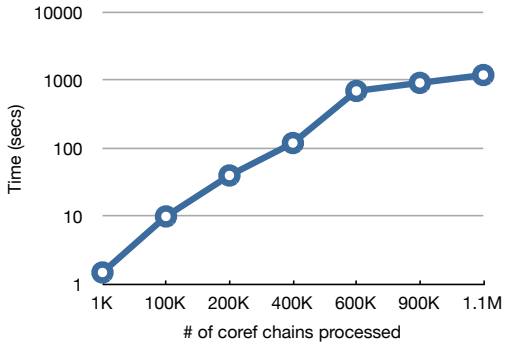


Figure 5: The number of coreference chains processed over time in the *kbp09* corpus. The processing of over 1 million coreference chains is at least an order of magnitude larger than previous systems reported.

tems, and our experiments have demonstrated scalability to an order of magnitude larger data than previously published evaluations. Despite its speed and simplicity, we still obtain competitive results on a variety of data sets as compared with batch systems. In future work, we plan to investigate additional similarity metrics that can be computed efficiently, as well as experiments on web scale corpora.

References

Artiles, Javier, Satoshi Sekine, and Julio Gonzalo. 2008. Web people search: results of the first evaluation and the plan for the second. In *World Wide Web (WWW)*.

Bagga, Amit and Breck Baldwin. 1998. Entity-based cross-document coreferencing using the vector space model. In *Conference on Computational Linguistics (COLING)*.

Baron, Alex and Marjorie Freedman. 2008. Who is Who and What is What: Experiments in cross-document co-reference. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Blei, D.M., A.Y. Ng, and M.I. Jordan. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research (JMLR)*, 3:993–1022.

Can, F. and E. Ozkarahan. 1987. A dynamic cluster maintenance system for information retrieval. In *Conference on Research and Development in Information Retrieval (SIGIR)*.

Charikar, Moses, Chandra Chekuri, Tomás Feder, and Rajeev Motwani. 1997. Incremental clustering and dynamic information retrieval. In *ACM Symposium on Theory of Computing (STOC)*.

Chen, Ying and James Martin. 2007. Towards robust unsupervised personal name disambiguation. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Choi, Y. and C. Cardie. 2007. Structured local training and biased potential functions for conditional random fields with application to coreference resolution. In *North American Chapter of the Association*

for Computational Linguistics (NAACL), pages 65–72.

Cucerzan, Silviu. 2007. Large-scale named entity disambiguation based on wikipedia data. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 708–716.

Firth, J.R. 1957. A synopsis of linguistic theory 1930-1955. In *Studies in Linguistic Analysis*, pages 1–32. Oxford: Philological Society.

Gooi, Chung Heong and James Allan. 2004. Cross-document coreference on a large scale corpus. In *North American Chapter of the Association for Computational Linguistics (NAACL)*.

Luo, X., A. Ittycheriah, H. Jing, N. Kambhatla, and S. Roukos. 2004. A mention-synchronous coreference resolution algorithm based on the bell tree. In *Association for Computational Linguistics (ACL)*.

Mann, Gideon S. and David Yarowsky. 2003. Unsupervised personal name disambiguation. In *Conference on Natural Language Learning (CONLL)*.

McNamee, Paul and Hoa Dang. 2009. Overview of the TAC 2009 knowledge base population track. In *Text Analysis Conference (TAC)*.

Ng, V. and C. Cardie. 2002. Improving machine learning approaches to coreference resolution. In *Association for Computational Linguistics (ACL)*, pages 104–111.

Nicolae, C. and G. Nicolae. 2006. Bestcut: A graph algorithm for coreference resolution. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 275–283. Association for Computational Linguistics.

Omiecinski, Edward and Peter Scheuermann. 1984. A global approach to record clustering and file reorganization. In *Conference on Research and Development in Information Retrieval (SIGIR)*.

Pradhan, S.S., L. Ramshaw, R. Weischedel, J. MacBride, and L. Micciulla. 2007. Unrestricted coreference: Identifying entities and events in ontonotes. In *International Conference on Semantic Computing (ICSC)*.

Ramshaw, L. and R. Weischedel. 2005. Information extraction. In *IEEE ICASSP*.

Sanderson, Mark. 2000. Retrieving with good sense. *Information Retrieval*, 2(1):45–65.

Sandhaus, Evan. 2008. The new york times annotated corpus. Linguistic Data Consortium, Philadelphia.

Soon, Wee Meng, Hwee Tou Ng, and Daniel Chung Yong Lim. 2001. A machine learning approach to coreference resolution of noun phrases. *Computational Linguistics*.

Yao, L., D. Mimno, and A. McCallum. 2009. Efficient methods for topic model inference on streaming document collections. In *Knowledge discovery and data mining (KDD)*.

Multilingual Summarization Evaluation without Human Models

Horacio Saggion
TALN - DTIC
Universitat Pompeu Fabra
horacio.saggion@upf.edu

Juan-Manuel Torres-Moreno
LIA/Université d'Avignon
École Polytechnique de Montréal
juan-manuel.torres@univ-avignon.fr

Iria da Cunha
IULA/Universitat Pompeu Fabra
LIA/Université d'Avignon
iria.dacunha@upf.edu

Eric SanJuan
LIA/Université d'Avignon
eric.sanjuan@univ-avignon.fr

Patricia Velázquez-Morales
VM Labs
patricia_vazquez@yahoo.com

Abstract

We study correlation of rankings of text summarization systems using evaluation methods with and without human models. We apply our comparison framework to various well-established content-based evaluation measures in text summarization such as coverage, Responsiveness, Pyramids and ROUGE studying their associations in various text summarization tasks including generic and focus-based multi-document summarization in English and generic single-document summarization in French and Spanish. The research is carried out using a new content-based evaluation framework called FRESA to compute a variety of divergences among probability distributions.

1 Introduction

Text summarization evaluation has always been a complex and controversial issue in computational linguistics. In the last decade, significant advances have been made in the summarization evaluation field. Various evaluation frameworks have been established and evaluation measures developed. SUMMAC (Mani et al., 2002), in 1998, provided the first system independent framework for summary evaluation; the Document Understanding Conference (DUC) (Over et al., 2007) was the main evaluation forum from 2000 until 2007; nowadays, the Text Analysis Conference (TAC)[1] provides a forum for assessment of different information access technologies including text summarization.

Evaluation in text summarization can be extrinsic or intrinsic (Spärck-Jones and Galliers, 1996). In an extrinsic evaluation, the summaries are assessed in the context of an specific task a human or machine has to carry out; in an intrinsic evaluation, the summaries are evaluated in reference to some ideal model. SUMMAC was mainly extrinsic while DUC and TAC followed an intrinsic evaluation paradigm. In order to intrinsically evaluate summaries, the automatic summary (*peer*) has to be compared to a *model* summary or summaries. DUC used an interface called SEE to allow human judges compare a peer summary to a model summary. Using SEE, human judges give a *coverage* score to the peer summary representing the degree of overlap with the model summary. Summarization systems obtain a final coverage score which is the average of the coverage's scores associated to their summaries. The system's coverage score can then be used to rank summarization systems. In the case of query-focused summarization (e.g. when the summary has to respond to a question or set of questions) a *Responsiveness* score is also assigned to each summary which indicates how responsive the summary is to the question(s).

Because manual comparison of peer summaries with model summaries is an arduous and costly

[1] http://www.nist.gov/tac

Coling 2008: Poster Volume, pages 1059–1067,
Beijing, August 2010

process, a body of research has been produced in the last decade on automatic content-based evaluation procedures. Early studies used text similarity measures such as cosine similarity (with or without weighting schema) to compare peer and model summaries (Donaway et al., 2000), various vocabulary overlap measures such as set of n-grams overlap or longest common subsequence between peer and model have also been proposed (Saggion et al., 2002; Radev et al., 2003). The *Bleu* machine translation evaluation measure (Papineni et al., 2002) has also been tested in summarization (Pastra and Saggion, 2003). The DUC conferences adopted the ROUGE package for content-based evaluation (Lin, 2004). It implements a series of recall measures based on n-gram co-occurrence statistics between a peer summary and a set of model summaries. ROUGE measures can be used to produce systems ranks. It has been shown that system rankings produced by some ROUGE measures (e.g., ROUGE-2 which uses bi-grams) correlate with rankings produced using coverage. In recent years the Pyramids evaluation method (Nenkova and Passonneau, 2004) was introduced. It is based on the distribution of "content" in a set of model summaries. Summary Content Units (SCUs) are first identified in the model summaries, then each SCU receives a weight which is the number of models containing or expressing the same unit. Peer SCUs are identified in the peer, matched against model SCUs, and weighted accordingly. The Pyramids score given to the peer is the ratio of the sum of the weights of its units and the sum of the weights of the best possible ideal summary with the same number of SCUs as the peer. The Pyramids scores can be used for ranking summarization systems. Nenkova and Passonneau (2004) showed that Pyramids scores produced reliable system rankings when multiple (4 or more) models were used and that Pyramids rankings correlate with rankings produced by ROUGE-2 and ROUGE-SU2 (i.e. ROUGE with skip bi-grams). Still this method requires the creation of models and the identification, matching, and weighting of SCUs in both models and peers.

Donaway et al. (2000) put forward the idea of using directly the full document for comparison purposes, and argued that content-based measures which compare the document to the summary may be acceptable substitutes for those using model summaries. A method for evaluation of summarization systems without models has been recently proposed (Louis and Nenkova, 2009). It is based on the direct content-based comparison between summaries and their corresponding source documents. Louis and Nenkova (2009) evaluated the effectiveness of the Jensen-Shannon (Lin, 1991b) theoretic measure in predicting systems ranks in two summarization tasks query-focused and update summarization. They have shown that ranks produced by Pyramids and ranks produced by the Jensen-Shannon measure correlate. However, they did not investigate the effect of the measure in past summarization tasks such as generic multi-document summarization (DUC 2004 Task 2), biographical summarization (DUC 2004 Task 5), opinion summarization (TAC 2008 OS), and summarization in languages other than English.

We think that, in order to have a better understanding of document-summary evaluation measures, more research is needed. In this paper we present a series of experiments aimed at a better understanding of the value of the Jensen-Shannon divergence for ranking summarization systems.

We have carried out experimentation with the proposed measure and have verified that in certain tasks (such as those studied by (Louis and Nenkova, 2009)) there is a strong correlation among Pyramids and Responsiveness and the Jensen-Shannon divergence, but as we will show in this paper, there are datasets in which the correlation is not so strong. We also present experiments in Spanish and French showing positive correlation between the Jensen-Shannon measure and ROUGE.

The rest of the paper is organized in the following way: First in Section 2 we introduce related work in the area of content-based evaluation identifying the departing point for our inquiry; then in Section 3 we explain the methodology adopted in our work and the tools and resources used for experimentation. In Section 4 we present the experiments carried out together with the results. Section 5 discusses the results and Section 6 concludes the paper.

2 Related Work

One of the first works to use content-based measures in text summarization evaluation is due to (Donaway et al., 2000) who presented an evaluation framework to compare rankings of summarization systems produced by recall and cosine-based measures. They showed that there was weak correlation between rankings produced by recall, but that content-based measures produce rankings which were strongly correlated, thus paving the way for content-based measures in text summarization evaluation.

Radev et al. (2003) also compared various evaluation measures based on vocabulary overlap. Although these measures were able to separate random from non-random systems, no clear conclusion was reached on the value of each of the measures studied.

Nowadays, a widespread summarization evaluation framework is ROUGE (Lin and Hovy, 2003) which, as we have mentioned before, offers a set of statistics that compare peer summaries with models. Various statistics exist depending on the used n-gram and on the type of text processing applied to the input texts (e.g., lemmatization, stopword removal).

Lin et al. (2006) proposed a method of evaluation based on the use of "distances" or divergences between two probability distributions (the distribution of units in the automatic summary and the distribution of units in the model summary). They studied two different Information Theoretic measures of divergence: the Kullback-Leibler ($\mathcal{KL}$) (Kullback and Leibler, 1951) and Jensen-Shannon ($\mathcal{JS}$) (Lin, 1991a) divergences. In this work we use the Jensen-Shannon ($\mathcal{JS}$) divergence that is defined as follows:

$$\begin{aligned} D_{\mathcal{JS}}(P||Q) &= \frac{1}{2}\sum_w P_w \log_2 \frac{2P_w}{P_w+Q_w} \\ &+ Q_w \log_2 \frac{2Q_w}{P_w+Q_w} \end{aligned} \quad (1)$$

This measure can be applied to the distribution of units in system summaries P and reference summaries Q and the value obtained used as a score for the system summary. The method has been tested by (Lin et al., 2006) over the DUC 2002 corpus for single and multi document summarization tasks showing good correlation among divergence measures and both coverage and ROUGE rankings.

Louis and Nenkova (2009) went even further and, as in (Donaway et al., 2000), proposed to directly compare the distribution of words in full documents with the distribution of words in automatic summaries to derive a content-based evaluation measure. They found high correlation among rankings produced using models and rankings produced without models. This work is the departing point for our inquiry into the value of measures that do not rely on human models.

3 Methodology

The methodology of this paper mirrors the one adopted in past work (Donaway et al., 2000; Louis and Nenkova, 2009). Given a particular summarization task T, p data points to be summarized with input material $\{I_i\}_{i=0}^{p-1}$ (e.g. document(s), questions, topics), s peer summaries $\{\text{SUM}_{i,k}\}_{k=0}^{s-1}$ for input i, and m model summaries $\{\text{MODEL}_{i,j}\}_{j=0}^{m-1}$ for input i, we will compare rankings of the s peer summaries produced by various evaluation measures. Some measures we use compare summaries with n out of the m models:

$$\text{MEASURE}_M(\text{SUM}_{i,k}, \{\text{MODEL}_{i,j}\}_{j=0}^{n}) \quad (2)$$

while other measures compare peers with all or some of the input material:

$$\text{MEASURE}_M(\text{SUM}_{i,k}, I_i') \quad (3)$$

where I_i' is some subset of input I_i. The values produced by the measures for each summary $\text{SUM}_{i,k}$ are averaged for each system $k = 0, \ldots, s-1$ and these averages are used to produce a ranking. Rankings are compared using Spearman Rank correlation (Spiegel and Castellan, 1998) used to measure the degree of association between two variables whose values are used to rank objects. We use this correlation to directly compare results to those presented in (Louis and Nenkova, 2009). Computation of correlations is

done using the CPAN Statistics-RankCorrelation-0.12 package[2], which computes the rank correlation between two vectors.

3.1 Tools

We carry out experimentation using a new summarization evaluation framework: FRESA –FRamework for Evaluating Summaries Automatically– which includes document-based summary evaluation measures based on probabilities distribution. As in the ROUGE package, FRESA supports different n-grams and skip n-grams probability distributions. The FRESA environment can be used in the evaluation of summaries in English, French, Spanish and Catalan, and it integrates filtering and lemmatization in the treatment of summaries and documents. It is developed in Perl and will be made publicly available. We also use the ROUGE package to compute various ROUGE statistics in new datasets.

3.2 Summarization Tasks and Data Sets

We have conducted our experimentation with the following summarization tasks and data sets:

Generic multi-document-summarization in English (i.e. production a short summary of a cluster of related documents) using data from DUC 2004[3] corpus task 2: 50 clusters (10 documents each) – 294,636 words.

Focused-based summarization in English (i.e. production a short focused multi-document summary focused on the question "who is X?", where X is a person's name) using data from the DUC 2004 task 5: 50 clusters (10 documents each plus a target person name) – 284,440 words.

Update-summarization task that consists of creating a summary out of a cluster of documents and a topic. Two sub-tasks are considered here: A) an initial summary has to be produced based on an initial set of documents and topic; B) an update summary has to be produced from a different (but related) cluster assuming documents used in A) are known. The English TAC 2008 Update Summarization dataset is used which consists of 48 topics with 20 documents each – 36,911 words.

Opinion summarization where systems have to analyze a set of blog articles and summarize the opinions about a target in the articles. The TAC 2008 Opinion Summarization in English[4] data set (taken from the Blogs06 Text Collection) is used: 25 clusters and targets (i.e., target entity and questions) were used – 1,167,735 words.

Generic single-document summarization in Spanish using the "Spanish *Medicina Clínica*"[5] corpus which is composed of 50 biomedical articles in Spanish, each one with its corresponding author abstract – 124,929 words.

Generic single document summarization in French using the "Canadien French Sociological Articles" corpus from the journal *Perspectives interdisciplinaires sur le travail et la santé* (PISTES)[6]. It contains 50 sociological articles in French with their corresponding author abstracts – 381,039 words.

3.3 Summarization Systems

For experimentation in the TAC and the DUC datasets we directly use the peer summaries produced by systems participating in the evaluations. For experimentation in Spanish and French (single-document summarization) we have created summaries at the compression rates of the model summaries using the following summarization systems:

- *CORTEX* (Torres-Moreno et al., 2002), a single-document sentence extraction system for Spanish and French that combines various statistical measures of relevance (angle between sentence and topic, various Hamming weights for sentences, etc.) and applies an optimal decision algorithm for sentence selection;
- *ENERTEX* (Fernandez et al., 2007), a summarizer based on a theory of textual energy;

[2] http://search.cpan.org/~gene/Statistics-RankCorrelation-0.12/

[3] http://www-nlpir.nist.gov/projects/duc/guidelines/2004.html

[4] http://www.nist.gov/tac/data/index.html

[5] http://www.elsevier.es/revistas/ctl_servlet?_f=7032&revistaid=2

[6] http://www.pistes.uqam.ca/

- *SUMMTERM* (Vivaldi et al., 2010), a terminology-based summarizer that is used for summarization of medical articles and uses specialized terminology for scoring and ranking sentences;
- *JS* summarizer, a summarization system that scores and ranks sentences according to their Jensen-Shannon divergence to the source document;
- a *lead-based* summarization system that selects the lead sentences of the document;
- a *random-based* summarization system that selects sentences at random;
- the multilingual word-frequency *Open Text Summarizer* (Yatsko and Vishnyakov, 2007);
- the *AutoSummarize* program of Microsoft Word;
- the commercial SSSummarizer[7];
- the *Pertinence* summarizer[8];
- the *Copernic* summarizer[9].

3.4 Evaluation Measures

The following measures derived from human assessment of the content of the summaries are used in our experiments:

- *Coverage* is understood as the degree to which one peer summary conveys the same information as a model summary (Over et al., 2007). Coverage was used in DUC evaluations.
- *Responsiveness* ranks summaries in a 5-point scale indicating how well the summary satisfied a given information need (Over et al., 2007). It is used in focused-based summarization tasks. Responsiveness was used in DUC-TAC evaluations.
- *Pyramids* (briefly introduced in Section 1) (Nenkova and Passonneau, 2004) is a content assessment measure which compares content units in a peer summary to weighted content units in a set of model summaries. Pyramids is the adopted metric for content-based evaluation in the TAC evaluations.

For DUC and TAC datasets the values of these measures are available and we used them directly.

We used the following automatic evaluation measures in our experiments:

- We use the *Rouge* package (Lin, 2004) to compute various statistics. For the experiments presented here we used uni-grams, bi-grams, and the skip bi-grams with maximum skip distance of 4 (ROUGE-1, ROUGE-2 and ROUGE-SU4). ROUGE is used to compare a peer summary to a set of model summaries in our framework.
- Jensen-Shannon divergence formula given in Equation 1 is implemented in our FRESA package with the following specification for the probability distribution of words w.

$$P_w = \frac{C_w^T}{N} \tag{4}$$

$$Q_w = \begin{cases} \frac{C_w^S}{N_S} & \text{if } w \in S \\ \frac{C_w^T + \delta}{N + \delta * B} & \text{elsewhere} \end{cases} \tag{5}$$

Where P is the probability distribution of words w in text T and Q is the probability distribution of words w in summary S; N is the number of words in text and summary $N = N_T + N_S$, $B = 1.5|V|$, C_w^T is the number of words in the text and C_w^S is the number of words in the summary. For smoothing the summary's probabilities we have used $\delta = 0.005$.

4 Experiments and Results

We first replicated the experiments presented in (Louis and Nenkova, 2009) to verify that our implementation of JS produced correlation results compatible with that work. We used the TAC 2008 Update Summarization data set and computed JS and ROUGE measures for each peer

[7] http://www.kryltech.com/summarizer.htm

[8] http://www.pertinence.net

[9] http://www.copernic.com/en/products/summarizer

summary. We produced two system rankings (one for each measure), which were compared to rankings produced using the manual Pyramids and Responsiveness scores. Spearman correlations were computed among the different rankings. The results are presented in Table 1. These results confirm a high correlation among Pyramids, Responsiveness, and JS. We also verified high correlation between JS and ROUGE-2 (0.83 Spearman correlation, not shown in the table) in this task and dataset.

Measure	Pyr.	p-value	Resp.	p-value
ROUGE-2	0.96	$p < 0.005$	0.92	$p < 0.005$
JS	0.85	$p < 0.005$	0.74	$p < 0.005$

Table 1: Spearman system rank correlation of content-based measures in TAC 2008 Update Summarization task

Then, we experimented with data from DUC 2004, TAC 2008 Opinion Summarization pilot and with single document summarization in Spanish and French. In spite of the fact that the experiments for French and Spanish corpora use less data points (i.e., less summarizers per task) than for English, results are still quite significant.

For DUC 2004, we computed the JS measure for each peer summary in tasks 2 and 5 and we used JS and the official ROUGE, coverage, and Responsiveness scores to produce systems' rankings. The various Spearman's rank correlation values for DUC 2004 are presented in Tables 2 (for task 2) and 3 (for task 5). For task 2, we have verified a strong correlation between JS and coverage. For task 5, the correlation between JS and coverage is weak, and the correlation between JS and Responsiveness weak and negative.

Measure	Cov.	p-value
ROUGE-2	0.79	$p < 0.0050$
JS	0.68	$p < 0.0025$

Table 2: Spearman system rank correlation of content-based measures with coverage in DUC 2004 Task 2

Although the Opinion Summarization task is a new type of summarization task and its evaluation is a complicated issue, we have decided to compare JS rankings with those obtained using Pyramids and Responsiveness in TAC 2008. Spearman's correlation values are listed in Table 4. As can be seen, there is weak and negative correlation of JS with both Pyramids and Responsiveness. Correlation between Pyramids and Responsiveness rankings is high for this task (0.71 Spearman's correlation value).

Measure	Cov.	p-value	Resp.	p-value
ROUGE-2	0.78	$p < 0.001$	0.44	$p < 0.05$
JS	0.40	$p < 0.050$	-0.18	$p < 0.25$

Table 3: Spearman system rank correlation of content-based measures in DUC 2004 Task 5

Measure	Pyr.	p-value	Resp.	p-value
JS	-0.13	$p < 0.25$	-0.14	$p < 0.25$

Table 4: Spearman system rank correlation of content-based measures in TAC 2008 Opinion Summarization task

For experimentation in Spanish and French, we have run 11 multi-lingual summarization systems over each of the documents in the two corpora, producing summaries at a compression rate close to the compression rate of the provided authors' abstracts. We have computed JS and ROUGE measures for each summary and we have averaged the measure's values for each system. These averages were used to produce rankings per each measure. We computed Spearman's correlations for all pairs of rankings. Results are presented in Tables 5-6. All results show medium to strong correlation between JS and ROUGE measures. However the JS measure based on uni-grams has lower correlation than JSs which use n-grams of higher order.

5 Discussion

The departing point for our inquiry into text summarization evaluation has been recent work on the use of content-based evaluation metrics that do not rely on human models but that compare summary content to input content directly (Louis and Nenkova, 2009). We have some positive and some negative results regarding the direct use of the full document in content-based evaluation. We have verified that in both generic muti-document sum-

Measure	ROUGE-1	p-value	ROUGE-2	p-value	ROUGE-SU4	p-value
JS	0.56	$p < 0.100$	0.46	$p < 0.100$	0.45	$p < 0.200$
JS_2	0.88	$p < 0.001$	0.80	$p < 0.002$	0.81	$p < 0.005$
JS_4	0.88	$p < 0.001$	0.80	$p < 0.002$	0.81	$p < 0.005$
JS_M	0.82	$p < 0.005$	0.71	$p < 0.020$	0.71	$p < 0.010$

Table 5: Spearman system rank correlation of content-based measures with ROUGE in the *Medicina Clinica* Corpus (Spanish)

Measure	ROUGE-1	p-value	ROUGE-2	p-value	ROUGE-2	p-value
JS	0.70	$p < 0.050$	0.73	$p < 0.05$	0.73	$p < 0.500$
JS_2	0.93	$p < 0.002$	0.86	$p < 0.01$	0.86	$p < 0.005$
JS_4	0.83	$p < 0.020$	0.76	$p < 0.05$	0.76	$p < 0.050$
JS_M	0.88	$p < 0.010$	0.83	$p < 0.02$	0.83	$p < 0.010$

Table 6: Spearman system rank correlation of content-based measures with ROUGE in the PISTES Sociological Articles Corpus (French)

marization and in topic-based multi-document summarization in English correlation among measures that use human models (Pyramids, Responsiveness, and ROUGE) and a measure that does not use models (the Jensen Shannon divergence) is strong. We have found that correlation among the same measures is weak for summarization of biographical information and summarization of opinions in blogs. We believe that in these cases content-based measures should consider in addition to the input document, the summarization task (i.e. its text-based representation) to better assess the content of the peers, the task being a determinant factor in the selection of content for the summary. Our multi-lingual experiments in generic single-document summarization confirm a strong correlation among the Jensen-Shannon divergence and ROUGE measures. It is worth noting that ROUGE is in general the chosen framework for presenting content-based evaluation results in non-English summarization. For the experiments in Spanish, we are conscious that we only have one model summary to compare with the peers. Nevertheless, these models are the corresponding abstracts written by the authors of the articles and this is in fact the reason for choosing this corpus. As the experiments in (da Cunha et al., 2007) show, the professionals of a specialized domain (as, for example, the medical domain) adopt similar strategies to summarize their texts and they tend to choose roughly the same content chunks for their summaries. Because of this, the summary of the author of a medical article can be taken as reference for summaries evaluation. It is worth noting that there is still debate on the number of models to be used in summarization evaluation (Owkzarzak and Dang, 2009). In the French corpus PISTES, we suspect the situation is similar to the Spanish case.

6 Conclusions and Future Work

This paper has presented a series of experiments in content evaluation in text summarization to assess the value of content-based measures that do not rely on the use of model summaries for comparison purposes. We have carried out extensive experimentation with different summarization tasks drawing a clearer picture of tasks where the measures could be applied. This paper makes the following contributions:

- We have shown that if we are only interested in ranking summarization systems according to the content of their automatic summaries, there are tasks where models could be substituted by the full document in the computation of the Jensen-Shannon divergence measure obtaining reliable rankings. However, we have also found that the substitution of models by full-documents is not always advisable. We have found weak correlation among different rankings in complex summarization tasks such as the summarization of biographical information and the summa-

Measure	ROUGE-1	p-value	ROUGE-2	p-value	ROUGE-2	p-value
JS	0.83	$p < 0.002$	0.66	$p < 0.05$	0.741	$p < 0.01$
JS_2	0.80	$p < 0.005$	0.59	$p < 0.05$	0.68	$p < 0.02$
JS_4	0.75	$p < 0.010$	0.52	$p < 0.10$	0.62	$p < 0.05$
JS_M	0.85	$p < 0.002$	0.64	$p < 0.05$	0.74	$p < 0.01$

Table 7: Spearman system rank correlation of content-based measures with ROUGE in the RPM2 Corpus (French)

rization of opinions about an "entity".

- We have also carried out large-scale experiments in Spanish and French which show positive medium to strong correlation among system's ranks produced by ROUGE and divergence measures that do not use the model summaries.

- We have also presented a new framework, FRESA, for the computation of measures based on Jensen-Shannon divergence. Following the ROUGE approach, FRESA implements word uni-grams, bi-grams and skip n-grams for the computation of divergences. The framework is being made available to the community for research purposes.

Although we have made a number of contributions, this paper leaves many questions open that need to be addressed. In order to verify correlation between ROUGE and JS, in the short term we intend to extend our investigation to other languages and datasets such as Portuguese and Chinese for which we have access to data and summarization technology. We also plan to apply our evaluation framework to the rest of the DUC and TAC summarization tasks to have a full picture of the correlations among measures with and without human models. In the long term we plan to incorporate a representation of the task/topic in the computation of the measures.

Acknowledgements

We thank three anonymous reviewers for their valuable and enthusiastic comments. Horacio Saggion is grateful to the Programa Ramón y Cajal from the Ministerio de Ciencia e Innovación, Spain and to a Comença grant from Universitat Pompeu Fabra (COMENÇA10.004). This work is partially supported by a postdoctoral grant (National Program for Mobility of Research Human Resources; National Plan of Scientific Research, Development and Innovation 2008-2011) given to Iria da Cunha by the Ministerio de Ciencia e Innovación, Spain.

References

da Cunha, Iria, Leo Wanner, and M. Teresa Cabré. 2007. Summarization of specialized discourse: The case of medical articles in spanish. *Terminology*, 13(2):249–286.

Donaway, Robert L., Kevin W. Drummey, and Laura A. Mather. 2000. A comparison of rankings produced by summarization evaluation measures. In *NAACL-ANLP 2000 Workshop on Automatic Summarization*, pages 69–78, Morristown, NJ, USA. ACL.

Fernandez, Silvia, Eric SanJuan, and Juan-Manuel Torres-Moreno. 2007. Textual Energy of Associative Memories: performants applications of Enertex algorithm in text summarization and topic segmentation. In *MICAI'07*, pages 861–871.

Kullback, S. and R.A. Leibler. 1951. On information and sufficiency. *Annals of Mathematical Statistics*, 22(1):79–86.

Lin, C.-Y. and E. Hovy. 2003. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of HLT-NAACL 2003*, pages 71–78, Morristown, NJ, USA. ACL.

Lin, Chin-Yew, Guihong Cao, Jianfeng Gao, and Jian-Yun Nie. 2006. An information-theoretic approach to automatic evaluation of summaries. In *Conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 463–470, Morristown, NJ, USA. ACL.

Lin, J. 1991a. Divergence measures based on the shannon entropy. *IEEE Transactions on Information Theory*, 37(145-151).

Lin, Jianhua. 1991b. Divergence measures based on the shannon entropy. *IEEE Transactions on Information theory*, 37:145–151.

Lin, Chin-Yew. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In Marie-Francine Moens, Stan Szpakowicz, editor, *Text Summarization Branches Out: ACL-04 Workshop*, pages 74–81, Barcelona, Spain, July.

Louis, Annie and Ani Nenkova. 2009. Automatically Evaluating Content Selection in Summarization without Human Models. In *Conference on Empirical Methods in Natural Language Processing*, pages 306–314, Singapore, August. ACL.

Mani, I., G. Klein, D. House, L. Hirschman, T. Firmin, and B. Sundheim. 2002. Summac: a text summarization evaluation. *Natural Language Engineering*, 8(1):43–68.

Nenkova, Ani and Rebecca Passonneau. 2004. Evaluating Content Selection in Summarization: The Pyramid Method. In *Proceedings of NAACL-HLT 2004*.

Over, Paul, Hoa Dang, and Donna Harman. 2007. Duc in context. *Information Processing & Management*, 43(6):1506–1520.

Owkzarzak, Karolina and Hoa Trang Dang. 2009. Evaluation of automatic summaries: Metrics under varying data conditions. In *Proceedings of the 2009 Workshop on Language Generation and Summarisation (UCNLG+Sum 2009)*, pages 23–30, Suntec, Singapore, August. ACL.

Papineni, K., S. Roukos, T. Ward, , and W. J. Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *ACL'02: 40th Annual meeting of the Association for Computational Linguistics*, pages 311–318.

Pastra, K. and H. Saggion. 2003. Colouring summaries Bleu. In *Proceedings of Evaluation Initiatives in Natural Language Processing*, Budapest, Hungary, 14 April. EACL.

Radev, Dragomir R., Simone Teufel, Horacio Saggion, Wai Lam, John Blitzer, Hong Qi, Arda Çelebi, Danyu Liu, and Elliott Drábek. 2003. Evaluation challenges in large-scale document summarization. In *ACL*, pages 375–382.

Saggion, H., D. Radev, S. Teufel, and W. Lam. 2002. Meta-evaluation of Summaries in a Cross-lingual Environment using Content-based Metrics. In *Proceedings of COLING 2002*, pages 849–855, Taipei, Taiwan, August 24-September 1.

Spärck-Jones, Karen and Julia Rose Galliers, editors. 1996. *Evaluating Natural Language Processing Systems, An Analysis and Review*, volume 1083 of *Lecture Notes in Computer Science*. Springer.

Spiegel, S. and N.J. Castellan, Jr. 1998. *Nonparametric Statistics for the Behavioral Sciences*. McGraw-Hill International.

Torres-Moreno, Juan-Manuel, Patricia Velźquez-Morales, and Jean-Guy Meunier. 2002. Condenśś de textes par des méthodes numŕiques. In *JADT'02*, volume 2, pages 723–734, St Malo, France.

Vivaldi, Jorge, Iria da Cunha, Juan-Manuel Torres-Moreno, and Patricia Velázquez-Morales. 2010. Automatic summarization using terminological and semantic resources. In *LREC'10*, volume 2, page 10, Malta.

Yatsko, V.A. and T.N. Vishnyakov. 2007. A method for evaluating modern systems of automatic text summarization. *Automatic Documentation and Mathematical Linguistics*, 41(3):93–103.

Argument Optionality in the LinGO Grammar Matrix

Safiyyah Saleem
University of Washington
ssaleem@u.washington.edu

Emily M. Bender
University of Washington
ebender@u.washington.edu

Abstract

We present a library of implemented HPSG analyses for argument optionality based on typological studies of this phenomenon in the world's languages, developed in the context of a grammar customization system that pairs a cross-linguistic core grammar with extensions for non-universal phenomena on the basis of user input of typological properties. Our analyses are compatible with multiple intersecting phenomena, including person, number, gender, tense, aspect and morphological rule formulation. We achieve 80-100% coverage on test suites from 10 natural languages.

1 Introduction

The LinGO Grammar Matrix customization system (Bender et al., 2002; 2010) is a web-based tool that creates starter grammars based on users' input to a questionnaire. The system comprises a core grammar covering linguistic phenomena that are posited to be universal (e.g. semantic compositionality) and a set of libraries providing analyses for phenomena that vary across languages (e.g. case). These resources are developed in the context of HPSG (Pollard and Sag, 1994), Minimal Recursion Semantics (Copestake et al., 2005), and the LKB grammar development environment (Copestake, 2002).

Previous to the work reported here, the Grammar Matrix customization system did not handle argument optionality—the possibility of leaving arguments unexpressed in lieu of overt pronouns. This phenomenon, also called pro-drop, argument drop, or null instantiation, is extremely common: according to Dryer (2008), 79% of the 674 languages sampled cannot or do not normally use independent pronouns in subject position. Accordingly, adding it to the customization system improves the system's ability to handle a large class of core sentences in many languages.

For example, in Modern Standard Arabic [arb] (Semitic), overt pronominal subjects are dropped in non-emphatic contexts. Previously, the system was able to model only the longer variant of (1).

(1) (hiyya) naama-t
(3.FEM.SG) sleep.PAST-3.FEM.SG
She slept. [arb]

Furthermore, there was no way to adequately account for languages such as Hausa [hau] (Chadic) which do not allow overt simple pronominal subjects and prohibit overt objects after certain verb forms. The grammar would predict the opposite grammaticality for the examples in (2).

(2) (*nī) nā-san amsā
(*1.SG) 1.SG.COMP-know answer
I know the answer. [hau]

It might seem that these facts could be handled by adding a rule that allows arguments to be dropped if an appropriate option is checked in the customization system. However, the data from Arabic and Hausa suggest that such an approach would be insufficient, as languages place different constraints on the contexts in which overt arguments are required or prohibited.

In §2 we discuss the broad range of typological variation in argument optionality in the world's languages. In §3 we offer a set of HPSG analyses for these patterns. §4 explains how these analyses were incorporated into the Grammar Matrix

Coling 2008: Poster Volume, pages 1068–1076,
Beijing, August 2010

customization system and integrated with the existing libraries. We then present the results of a three-tiered evaluation of the implemented system in §5. The results demonstrate that the system is capable of accurately modeling the attested syntactic argument optionality patterns exhibited by a typologically diverse group of languages as well as the currently unattested but logically possible co-occurrence restrictions on affixes and overt arguments. To our knowledge, this is the first such system. The paper closes with a brief look at how the library could be extended even further to capture the range of semantic distinctions.

2 Typological Patterns

The typological literature shows that argument optionality is extremely common: Dryer (2008) found that of 674 geographically and genetically diverse languages, only 141 normally or obligatorily used independent pronominal subjects. Dryer distinguishes 4 categories in the remaining 533 languages, corresponding to how information about the person, number, and gender (PNG) of the subject is encoded: affixation on the verb, clitics on variable hosts, no encoding, or a mixed strategy. In addition, there are other dimensions in which languages vary, e.g., constraints on contexts in which dropping is done (see (1)–(2)).

Although we were unable to find a similar comprehensive survey of unexpressed objects, there is evidence to suggest that it too may be very widespread. In particular, lexically-licensed object dropping seems to be very common. Even English, which has a very strong preference for overt subjects, can be analyzed as licensing lexically-based object dropping (Fillmore, 1986). As with subject dropping, we also found a number of different co-occurrence restrictions on the presence of verbal affixes and overt objects. Some languages always encode the PNG of an object on the verb, others optionally do so if an overt object is present and obligatorily do so if one is not, while still others do not encode this information at all.

Drawing on work by Dryer and others, Table 1 summarizes the 6 major dimensions along which the rules licensing argument dropping differ. The first constraint is syntactic context. Most languages that license argument dropping do so regardless of tense/aspect, mood, or person. Finnish [fin] and Hebrew [heb] are two notable exceptions (Vainikka and Levy, 1999).

The second constraint, lexically-based licensing, is most commonly found in object dropping. For example, while English usually prohibits argument dropping, it arguably licenses it with verbs such as 'found out', 'agree', and 'promise' (Fillmore, 1986). Lexically-based subject drop is found in Tamil [tam], which generally licenses subject dropping aside from some weather related verbs (Asher, 1985).

The third constraint, noun phrase type, captures the difference between a language such as Hausa which generally prohibits independent pronouns from appearing as subjects and other languages, which allow pronouns in subject position (possibly with emphatic interpretations).

The fourth constraint concerns the position of PNG markers. Of the languages with subject PNG markers and subject dropping, many encode subject PNG as a verbal affix. This pattern is exhibited by such geographically and genetically diverse languages as Spanish [spa], Arabic [arb], West Greenlandic [kal], Tamil [tam], and Nkore-Kiga [nyn]. Other languages such as Chemehuevi [ute], Polish [pol], and Warlpiri [wbp] make use of a clitic which can attach to different types of hosts (Dryer, 2008).

The final two constraints concern co-occurrence restrictions between PNG markers and overt objects. In some Bantu languages such as Nkore-Kiga, a verbal affix is not used unless the object precedes the verb or is pronominal. Object markers are not used when a full NP follows the verb (Taylor, 1985). In written French [fra], verbal affixes[1] are required if an object is dropped and not permitted if it is overt. In Arabic, for most transitive verbs, an object marker is required if an object is dropped and is optional if it is present. Hausa exhibits a more complex pattern: for tenses in which the verbal affix denoting PNG is morphologically separable from the tense marker, the PNG affix is optional if an overt noun phrase is present and required if it is not (Newman, 2000).

[1] See (Miller and Sag, 1997) for convincing arguments that so-called 'clitics' in French are actually affixes.

Constraint	(GF)	Possible Values
Syntactic context	(SUBJ)	{ All, select } tenses/aspects/moods/persons
Lexically-based	(SUBJ, OBJ)	{ All, select } verbs
Noun phrase type	(SUBJ, OBJ)	Independent pronouns { allowed, prohibited }
Placement of PNG marker	(SUBJ)	{ Verb, variable host }
PNG marking w/ dropped argument	(OBJ)	{ Required, optional, not permitted }
PNG marking w/ overt argument	(OBJ)	{ Required, optional, not permitted }

Table 1: Typological variation in licensing argument dropping

Noting these differences led us to posit that when an argument is dropped, there are three possibilities. A verbal affix can be: not permitted, optional, or required. The same three possibilities exist for overt objects as well. Combining what happens when an argument is dropped with what happens when it is present, gives us nine logically possible co-occurrence patterns.

Our review of the typological literature has shown that languages place different constraints on argument dropping. These constraints can be lexical, syntactic, or related to affixation and affix/overt-argument co-occurrence restrictions.

3 Analysis

This section presents HPSG analyses modeling the six dimensions of variation described in §2.

HPSG models natural language by positing lexical entries, lexical rules, and phrase structure rules, all described in terms of feature structures. A central idea, inspired by earlier work in Categorial Grammar (Ajdukiewicz, 1935; Bar-Hillel, 1953), is the notion of valence features. These list-valued features (including SUBJ and COMPS) contain information about the dependents required by a head. The valence lists are projected up the tree within the domain of each head, but shortened as the dependents are realized. A sentence is thus a verbal projection with empty SUBJ and COMPS lists.

In this context, argument dropping is the shortening of a valence list without the overt realization of the argument. Formally, this can be accomplished in at least three different ways: (1) In the mapping of arguments from the ARG-ST (argument structure) feature to the valence lists, one or more arguments can be suppressed, (2) lexical rules can operate on the valence lists, shortening them, or (3) unary (non-branching) phrase structure rules can cancel off valence elements. In this work, we take the third approach, as we find it affords us the most flexibility to deal with variations across languages in constraints on argument optionality, while promoting similarity of analyses across languages.

We control the applicability of the unary-branching rules with the boolean feature OPT, marked on elements of valence lists.[2] For languages which allow subject/object dropping, we instantiate new phrase structure rules: *head-opt-subj-phrase* and/or *head-opt-comp-phrase*. These rules allow the head verb to satisfy a valence requirement without combining with another expression. To undergo these rules, the head daughter (the verb) must specify that the argument that is to be dropped is compatible with [OPT +]. This is sufficient to account for many languages. However, to ensure that languages which have lexical, syntactic context, and affix co-occurrence restrictions do not overgenerate, further additions to the grammar are necessary.

For lexical and affix-co-occurrence restrictions, we prevent overgeneration by manipulating the OPT feature. In languages which only license argument dropping for certain lexical items, we force those verbs which do not allow argument dropping to have arguments that are constrained to be [OPT −]. This prevents them from undergoing the subject/object dropping rules. Verbs are then classified into four different types based on whether or not they allow subject and/or object dropping. Individual lexical items instantiate these types. For those verbs which do not allow a particular argument to be dropped, the only way to satisfy the valence requirement is to combine with an overt argument.

[2]This feature was provided by the core Matrix but was not previously used in the customization system. To our knowledge it is not commonly used within HPSG analyses aside from in grammars that were derived from the Matrix.

Dropped/Overt Argument Affix	Overt Arg Rule	No-Marker-Rule	Marker-Rule	Transitive Verb Lex
required/required	underspecified	none	underspecified	needs lex rule
optional/optional	underspecified	none	underspecified	underspecified
not permitted/not permitted	underspecified	none	none	underspecified
required/optional	OPT −	OPT −	underspecified	needs lex rule
optional/not permitted	OPT −	none	OPT +	underspecified
not permitted/required	OPT −	OPT +	OPT −	needs lex rule
required/not permitted	OPT −	OPT −	OPT +	needs lex rule
optional/required	OPT −	OPT +	underspecified	needs lex rule
not permitted/optional	OPT −	none	OPT −	underspecified

Table 2: Constraints associated with logically possible affix co-occurrence

Languages with complex affix co-occurrence restrictions are modeled by manipulating the OPT feature in a different way: Constraints are placed on lexical and phrase structure rules, as well as on lexical types. In particular, we constrain the rules which combine verbs with overt arguments to check that that argument position is compatible with [OPT −]. This allows the lexical rules attaching the affixes to constrain the optionality of the corresponding argument position. In some of the nine logical possibilities, enforcing these constraints requires sending the verb through "no-marker" lexical rules so that constraints associated with markerless verbs can be enforced. Table 2 summarizes the constraints on the OPT feature on lexical and phrase structure rules, as well as the constraints on lexical types. The first column of this table lists the nine logically possible combinations described in §2. For example, the row labeled "required/required" gives the analysis for a language like West Greenlandic, which allows object dropping and always requires an object marker on the verb regardless of whether or not an overt object is present. In such a language, neither the lexical rules nor the overt-complement phrase structure rule constrain OPT, but the transitive verb lex type is required to undergo some object marking lexical rule.

For licensing that is based on syntactic context (subject dropping only) such as the Finnish and Hebrew examples presented in §2, we place constraints on the daughter of the unary subject drop rule which restrict its application to the right contexts. For example, to account for the argument optionality pattern present in Finnish, we constrain the *head-opt-subj-phrase* rule to require that the item on the head daughter's SUBJ list be specified as non-third-person ([PER *non-third*]). Verbs not meeting this constraint are only allowed to empty their SUBJ lists by combining with an overt subject via the standard, binary *head-subj-phrase* rule. We have not seen a language which licenses subject dropping in syntactic contexts which do not form a natural class according to our feature system. However, our analysis easily lends itself to modeling this type of pattern if it exists by creating multiple different subtypes of the subject drop rule.

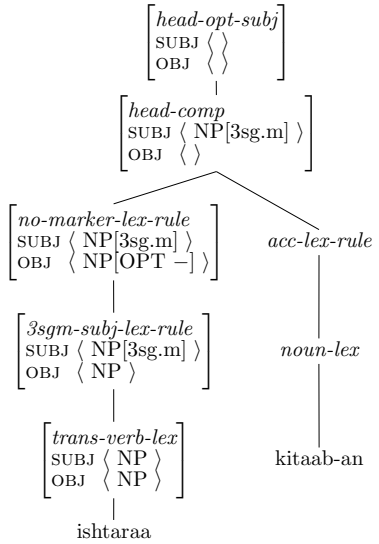


Figure 1: Parse structure for (3)

We close this section by illustrating our analysis with an example from Arabic. The sentence in (3) involves subject drop and an overt object. Since the object is overt, the verb bears only marking of subject PNG. The grammar that our system generates for Arabic assigns (3) the structure sketched in Figure 1.

(3) ishtaraa kitaab-an
3ms.buy.past book-acc
He bought a book

4 Customized Grammar Creation

Before the addition of the argument optionality library, the phenomena covered in the Grammar Matrix customization system included word order, person, number, gender, case, tense/aspect, coordination, matrix yes-no questions, and sentential negation. The user is also allowed to specify lexical items and the morphological rules associated with each of them. Each of the phenomena correspond to a page of the questionnaire.

As the user answers questions, the choices are saved in a 'choices' file. The questionnaire is dynamic and the answers contained in the 'choices' file affect the types of features that the user is able to choose from on subsequent pages. For example, if the user describes the language as having 1st, 2nd, and 3rd persons on the Person page, then on the lexicon page, the user can create separate noun types for each person. Once the 'choices' file contains responses to required sections, the user is able to create the customized starter grammar by clicking on the 'create grammar' button. This invokes the customization script which uses the responses contained in the file to create a grammar that is compatible with the LKB grammar development environment.

Our implementation entailed additions to two major components of the system: the web-based questionnaire and the customization script. To determine which, if any, of the analyses presented in §3 should be included in the customized grammar, we needed to elicit the type of argument optionality pattern the language exhibited. Thus, we added an Argument Optionality page to the questionnaire. The page is divided into two sections—one for subject dropping and one for object dropping. In the section on subject dropping, the user is asked whether subject dropping exists and if so, whether it is context-dependent. For context-dependent subject dropping, the user is allowed to specify the syntactic contexts in which subject dropping is licensed by choosing from a multi-select list of features. There is the option to create multiple contexts. The features that appear in the list are drawn from those that the user chose on previous pages in the questionnaire. The user is also directed to select whether subject dropping is lexically-based, whether affixes are required, optional or not permitted with overt arguments and whether affixes are required, optional or not permitted with dropped arguments. The questions presented in the object dropping section are identical to those in the subject dropping section with the exception that there is no question about context-dependent object dropping.

Since some of the constraints must be placed on individual lexical items and morphological rules, the page also includes instructions to the user on additional steps that need to be taken when completing the Lexicon page. For example, when describing a language where affixes are optional if an argument is dropped and not permitted if an overt argument is present, users are instructed to select 'overt-arg-not-permitted' for those affixes on the Lexicon page.

The changes to the customization script included adding each of the analyses described in §3 along with a mechanism for determining which of the analyses should be included in the grammar depending on the choices related to argument optionality, lexical items, and morphological rules contained in the 'choices' file. The resulting customized grammars include the rules and constraints necessary to allow and prohibit strings that do not contain overt arguments based upon the facts of a particular language as described by the user in the questionnaire.

5 Evaluation

The evaluation was conducted in a three stage process. Each stage involves constructing a set of test suites containing grammatical and ungrammatical strings representing the argument optionality pattern of a set of languages, generating grammars for the languages by answering the Grammar Matrix questionnaire, using the grammars to parse the sentences in the test suite, and hand-verifying the results. The three stages differed in the nature of the languages, the method by which the languages were selected, and the breadth of the customized grammars. The test suites are small, as they are specifically targeted at the phenomenon of argument optionality, but representative in the sense that they cover the space of relevant contrasts in each language.

5.1 Set 1: Pseudo-Languages

In the first stage, we tested the analyses presented in §3 by creating and then using the Grammar Matrix customization system to generate grammars for 38 pseudo-languages (sets of strings with associated grammaticality assignments) which collectively exhaustively exhibit each of the lexical, syntactic context or affix co-occurrence restriction patterns described in Table 1 (§2). All of the possible values identified for these given patterns are present in at least one language, as well as cross-classifications of different dimensions of constraints where appropriate. For example, there are pseudo-languages which share the property of always requiring object markers but differ in that one has lexically licensed object dropping and the other general object dropping. These pseudo-languages test the argument optionality analyses in isolation in that argument optionality is not constrained by other phenomena such as word order.

The customized grammars were able to accurately parse grammatical strings and rule out ungrammatical ones. Coverage on this set of 38 pseudo-languages was 100% with 0% overgeneration and no spurious ambiguity, thus validating the functioning of our analyses across the known typological space.

5.2 Set 2: Illustrative Languages

Next, we tested the system's performance in modeling part of a natural language. For this stage we deliberately chose several languages which exemplified interesting licensing and co-occurrence restriction patterns, including some which were considered during the development of the system. Each test suite included examples of grammatical and ungrammatical strings that were constructed based on the descriptions of the language given in the following sources: Suleiman 1990 (Arabic), Sulkala and Merja 1992 (Finnish), Newman 2000 (Hausa), and Asher 1985 (Tamil). As the test suites were designed to evaluate argument optionality, we restricted the test items to this phenomenon only. Other syntactic phenomena were only included if they affected the argument optionality pattern in the language. For example, gender distinctions were considered only for languages in which this was relevant to affix marking. A brief description of the argument optionality patterns found in these languages follows.

Arabic [arb] (Semitic) Pronominal subjects and objects are generally dropped. Subject affixes are always required whether or not an overt noun phrases is present. Affixes marking object person, number, and gender are required for strictly transitive verbs when an overt noun phrase is not present. Other transitive verbs appear to allow object drop without the object affix.

Finnish [fin] (Uralic) First and second person subjects are freely dropped and markers appear on the verb whether or not an overt noun phrase is present. Third person subjects are not allowed to be dropped with a referential interpretation; however, third person pronouns are obligatorily dropped for what Sulkala and Merja (1992) describe as a generic impersonal meaning. This description fits into what some linguists refer to as the fourth person—a non-referential impersonal syntactic/semantic distinction that is often realized in English as the impersonal pronoun *one*. Since Finnish shows evidence of further syntactic distinctions between generic and referential use of the third person marker, we have analyzed this marker as actually corresponding to two homophonous morphemes. One requires an overt subject and the other requires a dropped subject. There are no verbal affixes for PNG of the object.

Hausa [hau] (Chadic) Hausa generally requires pronominal subjects to be dropped. Simple, unmodified, uncoordinated independent pronouns are ungrammatical in subject position. Subject PNG is marked in a person aspect complex (PAC) along with tense and aspect information. The PAC precedes the lexical verb. When the PNG marker is morphologically segmentable from the tense/aspect, the PNG marker can be omitted if an overt noun phrase is present and is required if the noun phrase is not present. PNG is not marked for objects; however the verb form changes depending on whether a full noun phrase, pronoun, or no object immediately follows the verb.

Tamil [tam] (Dravidian) Subjects and objects can be freely dropped aside from a special class of weather verbs requiring overt subjects. Subject

PNG markers are always required whether a subject is overt or not. PNG is not marked for objects.

Lg.	Items	Grammatical	Ungrammatical	Coverage/ Overgeneration (%)
Arabic	13	10	3	90/0
Finnish	11	9	3	100/0
Hausa	20	8	12	100/0
Tamil	7	5	2	100/0

Table 3: Illustrative Languages Results

As shown in Table 3 we achieved 100% coverage over every test suite in this set except for Arabic. In addition, there was no overgeneration or spurious ambiguity. One Arabic item did not parse because the current implementation of our analyses does not elegantly account for obligatory object marking (with object drop) on some transitive verbs and optional object marking on others. We could have customized a grammar that included another, parallel set of lexical rules that would account for this item. Improvements to this aspect of the argument optionality library depend on upgrades to the morphotactic system.

5.3 Set 3: Held-out Languages

Finally, we tested a set of 'held out' languages not considered during development and chosen for their geographic and genetic diversity without regard for argument dropping patterns. We had previously created the non-argument optionality portions of these test suites and choices files to test the coverage of other libraries in the customization system and thus they include a wider variety of linguistic phenomena than Sets 1 or 2. As before, the construction and grammaticality judgements of the strings were based on descriptive grammars: Chirikba 2003 (Abkhaz), Press 1979 (Chemehuevi), Smirnova 1982 and Newman 2000 (Hausa), Pensalfini 2003 (Jingulu), Asher and Kumari 1997 (Malayalam), Taylor 1985 (Nkore-Kiga), and Fortescue 2003 (W. Greenlandic).

Due to space constraints, we provide only a summary of the argument optionality patterns in these languages (Table 4). All the languages licensed both subject and object dropping and in two of the six, dropping pronominal arguments was strongly preferred. Three languages have word order constraints on how argument optionality is realized: Abkhaz restricts the appearance of one of the third person affixes depending on verb-object order. Nkore-Kiga requires and prohibits the appearance of an object marker depending on where the overt object occurs. Chemehuevi requires that the clitic which is used to mark the subject appear in second position. It is also the only language that has lexical constraints on object dropping. Malayalam was the only language which did not mark person, number, and gender information for the subject.

The customized grammars were able to account for the majority of the patterns demonstrated in these languages (Table 5). We achieved 100% coverage on four languages with zero (Jingulu, Malayalam, West Greenlandic) or moderate (Abkhaz) overgeneration. The main source of errors found in the results is the handling of word order constraints: The grammars were unable to license (Chemehuevi) or restrict (Nkore-Kiga and Abkhaz) argument optionality based on the verb's and argument's positions in the sentence. Once the Grammar Matrix word order library has been improved and is able to account for second position clitics and fine-grained head-complement word order constraints, it will be a simple process to add the new feature(s) to existing lexical rules to account for these patterns. Incorporating the new functionality will not require any major changes to the argument optionality library aside from modifying the questionnaire to elicit the new information from the user.

Language	Items	Grammatical	Ungrammatical	Coverage/ Overgeneration (%)
Abkhaz	10	6	4	100/10
Chemehuevi	8	6	2	83.3/0
Jingulu	9	6	3	100/0
Malayalam	4	4	0	100/0
Nkore-Kiga	10	4	6	100/83.3
W. Greenlandic	5	3	2	100/0

Table 5: Held-out Language Results

In addition, we verified that the addition of argument optionality didn't reduce coverage on any other portion of these testsuites. This indicates that the new argument optionality library is interacting properly with existing libraries. Additional interactions will be tested as we add new libraries to the customization system.

	Object Dropping	Subject Dropping	Word Order Constraints	Lexical Constraints
Abkhaz	opt	opt	yes	none
Chemehuevi	opt	opt	yes	yes
Jingulu	opt	opt	none	none
Malayalam	opt	opt	no	none
Nkore-Kiga	pref	pref	yes	none
W. Greenlandic	pref	pref	none	none

Table 4: Existence of and constraints on argument optionality in six languages

6 Related Work

Subject dropping has been studied extensively within theoretical linguistics under many different frameworks (Rizzi, 1986; Bresnan, 2001; Ackema et al., 2006; Ginzburg and Sag, 2000). Within the context of HPSG, our analysis is similar to the one in the Grammar Matrix-derived Portuguese grammar (Branco and Costa, 2008) and to Müller's (2009) treatment of subject dropping in Maltese. These analyses differ from Ginzburg and Sag's (2000) HPSG analysis which uses language specific variations on the Argument Realization Principle to control whether the subject/object is placed onto the COMPS and/or SUBJ lists.

Language specific analyses have been implemented in deep, broad-coverage grammars for languages such as Japanese (Masuichi et al. (2003), Siegel and Bender (2002)) and Portuguese (Branco and Costa (2008)). Within the ParGram project (Butt et al., 2002), Kim et al. (2003) were able to directly port the argument optionality related rules from a Japanese grammar to Korean. However, to our knowledge, no one has implemented an analysis that has been applied to a large number of typologically, geographically, and genetically diverse languages.

7 Conclusion

Our current work has focused on modeling the variation in syntactic constraints on the licensing and restriction of argument dropping. To our knowledge, this is the first analysis of argument optionality that combines typological breadth with precision analyses that have been implemented and tested on a number of geographically and genetically diverse languages. Although we have tried to account for the patterns found in the typological literature, there may be variants that we are unaware of. We hope to learn of more patterns as the Grammar Matrix customization system is applied to an ever wider set languages.

While the current work focuses on syntactic variation, we intend to expand the argument optionality library to include semantic distinctions as well. A likely starting point would be the proposal given by Bender and Goss-Grubbs (2008) who present a way to model the discourse status (Prince, 1981) of an NP taking into account the differences between definite and indefinite null instantiation described by Fillmore (1986). In addition, ongoing work to improve the word order library may eventually allow us to more accurately model word-order based constraints.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. 0644097. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

Ackema, Peter, Patrick Brandt, Maaike Schoorlemmer, and Fred Weerman, editors. 2006. *Arguments and Agreement*. Oxford University Press, Oxford.

Ajdukiewicz, Kazimierz. 1935. Die syntaktische konexität. *Studia Philosophica*, 1:1–27.

Asher, R.E. and T.C. Kumari. 1997. *Malayalam*. Routledge, NY.

Asher, R.E. 1985. *Tamil*. Croom Helm, London.

Bar-Hillel, Yehoshua. 1953. A quasi-arithmetical notation for syntactic description. *Language*, 29:47–58.

Bender, Emily M. and David Goss-Grubbs. 2008. Semantic representations of syntactically marked discourse status in crosslinguistic perspective. In *Proc. 2008 Conference on Semantics in Text Processing*, pages 17–29.

Bender, Emily M., Dan Flickinger, and Stephan Oepen. 2002. The grammar matrix: An open-source starter-kit for the rapid development of cross-linguistically consistent broad-coverage precision grammars. In *Proc. Workshop on Grammar Engineering and Evaluation at COLING 2002*, pages 8–14.

Bender, Emily M., Scott Drellishak, Antske Fokkens, Michael Wayne Goodman, Daniel P. Mills, Laurie Poulson, and Safiyyah Saleem. 2010. Grammar prototyping and testing with the LinGO Grammar Matrix customization system. In *Proc. ACL 2010 Software Demonstrations*.

Branco, António and Francisco Costa. 2008. A computational grammar for deep linguistic processing of Portuguese: LXGram, version a.4.1. Technical report, University of Lisbon, Dept. of Informatics.

Bresnan, Joan. 2001. *Lexical Functional Syntax*. Blackwell, Boston.

Butt, Miriam, Helge Dyvik, Tracy Holloway King, Hiroshi Masuichi, and Christian Rohrer. 2002. The parallel grammar project. In *Proc. Workshop on Grammar Engineering and Evaluation at COLING 2002*, pages 1–7.

Chirikba, Viachesiav. 2003. *Abkhaz*. LINCOM, Munich.

Copestake, Ann, Dan Flickinger, Carl Pollard, and Ivan A. Sag. 2005. Minimal recursion semantics: An introduction. *Research on Language & Computation*, 3(4):281–332.

Copestake, Ann. 2002. *Implementing Typed Feature Structure Grammars*. CSLI, Stanford.

Dryer, Matthew. 2008. Expression of proniminal subjects. In Haspelmath, Martin, Matthew Dryer, David Gil, and Bernard Comrie, editors, *The World Atlas of Language Structures Online*, chapter 101. Max Planck Digital Library.

Fillmore, Charles. 1986. Pragmatically controlled zero anaphora. In *Proc. 12th annual meeting of the Berkeley Linguistics Society*, pages 95–107.

Fortescue, Michael. 2003. *West Greenlandic*. Croom Helm, London.

Ginzburg, Johnathan and Ivan Sag. 2000. *Interrogative Investigations*. CSLI, Stanford.

Kim, Roger, Mary Dalrymple, Ronald M. Kaplan, Tracy Holloway King, Hiroshi Masuichi, and Tomoko Ohkuma. 2003. Multilingual grammar development via grammar porting. In *ESSLLI 2003 Workshop on Ideas and Strategies for Multilingual Grammar Development*, pages 49–56.

Masuichi, Hiroshi, Tomoko Ohkuma, Hiroki Yoshimura, and Yasunari Harada. 2003. Japanese parser on the basis of the lexical-functional grammar formalism and its evaluation. In Dong Hong Ji, Kim Teng Lua, editor, *Proc. PACLIC17*, pages 298–309.

Miller, Philip H. and Ivan A. Sag. 1997. French clitic movement without clitics or movement. *Natural Language & Linguistic Theory*, 15(3):573–639.

Müller, Stefan. 2009. Towards an HPSG analysis of Maltese. In et al, Bernard Comrie, editor, *Introducing Maltese linguistics. Papers from the 1st International Conference on Maltese Linguistics*, pages 83–112. Benjamins, Amsterdam.

Newman, Paul. 2000. *The Hausa Language: An encyclopedic reference grammar*. Yale University Press, New Haven.

Pensalfini, Rob. 2003. *A Grammar of Jingulu: An Aboriginal language of the Northern Territory*. Pacific Linguistics, Canberra.

Pollard, Carl and Ivan A. Sag. 1994. *Head-Driven Phrase Structure Grammar*. The University of Chicago Press, Chicago, IL.

Press, Margaret. 1979. *Chemehuevi: A grammar and lexicon*. University of California Press, Berkeley.

Prince, Ellen. 1981. Toward a taxonomy of given-new information. In Cole, P., editor, *Radical Pragmatics*, pages 223–255. Academic Press, NY.

Rizzi, Luigi. 1986. Null objects in Italian and the theory of pro. *Linguistic Inquiry*, 17(3):501–557.

Siegel, Melanie and Emily M. Bender. 2002. Efficient deep processing of Japanese. In *Proc. 3rd Workshop on Asian Language Resources and International Standardization at COLING 2002*.

Smirnova, Mirra A. 1982. *The Hausa Language: A Descriptive Grammar*. Routledge, Boston.

Suleiman, Saleh M. 1990. The semantic functions of object deletion in classical arabic. *Language Sciences*, 12(2-3):255 – 266.

Sulkala, Helena and Karjalaninen Merja. 1992. *Finnish*. Routledge, NY.

Taylor, Charles. 1985. *Nkore-Kiga*. Croom Helm, London.

Vainikka, Anne and Yonata Levy. 1999. Empty subjects in Finnish and Hebrew. *Natural Language and Linguistic Theory*, 17:613–671.

Log-linear weight optimisation via Bayesian Adaptation in Statistical Machine Translation

Germán Sanchis-Trilles and **Francisco Casacuberta**
Departamento de Sistemas Informáticos y Computación
Instituto Tecnológico de Informática
Universidad Politécnica de Valencia
{gsanchis,fcn}@dsic.upv.es

Abstract

We present an adaptation technique for statistical machine translation, which applies the well-known Bayesian learning paradigm for adapting the model parameters. Since state-of-the-art statistical machine translation systems model the translation process as a log-linear combination of simpler models, we present the formal derivation of how to apply such paradigm to the weights of the log-linear combination. We show empirical results in which a small amount of adaptation data is able to improve both the non-adapted system and a system which optimises the above-mentioned weights on the adaptation set only, while gaining both in reliability and speed.

1 Introduction

The adaptation problem is a very common issue in statistical machine translation (SMT), where it is frequent to have very large collections of bilingual data belonging to e.g. proceedings from international entities such as the European Parliament or the United Nations. However, if we are currently interested in translating e.g. printer manuals or news data, we will need to find a way in which we can take advantage of such data.

The grounds of modern SMT were established in (Brown et al., 1993), where the machine translation problem was defined as follows: given a sentence $\mathbf{f}$ from a certain source language, an equivalent sentence $\hat{\mathbf{e}}$ in a given target language that maximises the posterior probability is to be found. According to the Bayes decision rule, such statement can be specified as follows:

$$\hat{\mathbf{e}} = \operatorname*{argmax}_{\mathbf{e}} \Pr(\mathbf{e}|\mathbf{f}) \tag{1}$$

Recently, a direct modelling of the posterior probability $\Pr(\mathbf{e}|\mathbf{f})$ has been widely adopted, and, to this purpose, different authors (Papineni et al., 1998; Och and Ney, 2002) proposed the use of the so-called log-linear models, where

$$p(\mathbf{e}|\mathbf{f}) = \frac{\exp \sum_{k=1}^{K} \lambda_k h_k(\mathbf{f}, \mathbf{e})}{\sum_{\mathbf{e}'} \exp \sum_{k=1}^{K} \lambda_k h_k(\mathbf{f}, \mathbf{e}')} \tag{2}$$

and the decision rule is given by the expression

$$\hat{\mathbf{e}} = \operatorname*{argmax}_{\mathbf{e}} \sum_{k=1}^{K} \lambda_k h_k(\mathbf{f}, \mathbf{e}) \tag{3}$$

where $h_k(\mathbf{f}, \mathbf{e})$ is a score function representing an important feature for the translation of $\mathbf{f}$ into $\mathbf{e}$, as for example the language model of the target language, a reordering model or several translation models. K is the number of models (or features) and λ_k are the weights of the log-linear combination. Typically, the weights $\mathbf{\Lambda} = [\lambda_1, \ldots, \lambda_K]^T$ are optimised with the use of a development set.

The use of log-linear models implied an important break-through in SMT, allowing for a significant increase in the quality of the translations produced. In this work, we present a Bayesian technique for adapting the weights of such log-linear models according to a small set of adaptation data.

In this paper, we will be focusing on adapting the weights vector $\mathbf{\Lambda}$, since appropriate values of such vector for a given domain do not necessarily imply a good combination in other domains. One naïve way in which some sort of adaptation can be performed on $\mathbf{\Lambda}$ is to re-estimate these weights

Coling 2008: Poster Volume, pages 1077–1085,
Beijing, August 2010

from scratch only on the adaptation data. However, such re-estimation may not be a good idea, whenever the amount of adaptation data available is not too big. On the one hand, because small amounts of adaptation data may easily yield overtrained values of $\mathbf{\Lambda}$, which may even lead to a degradation of the translation quality. On the other hand, because in some scenarios it is not feasible to re-estimate them because of the time it would take. Moreover, considering a re-estimation of $\mathbf{\Lambda}$ by using both the out-of-domain data and the adaptation set would not be appropriate either. For small amounts of adaptation data, such data would have no impact on the final value of $\mathbf{\Lambda}$, and the time required would be even higher. One such situation may be the Interactive Machine Translation (IMT) paradigm (Barrachina et al., 2009), in which a human translator may start translating a new document, belonging to a specific domain, and the system is required to produce an appropriate output as soon as possible without any prior re-training.

In this paper, a Bayesian adaptation approach solving both problems is presented. Nevertheless, adapting Λ constitutes just a first step towards the adaptation of all the parameters of the SMT model.

The rest of this paper is structured as follows. In next Section, we perform a brief review of current approaches to adaptation and Bayesian learning in SMT. Section 3 describes the typical framework for phrase-based translation in SMT. In Section 4, we present the way in which we apply Bayesian adaptation (BA) to log-linear models in SMT. In Section 5, we describe the practical approximations applied before implementing the BA technique described. In Section 6, experimental design and results are detailed. Conclusions and future work are explained in Section 7.

2 Related work

Adaptation in SMT is a research field that is receiving an increasing amount of attention. In (Nepveu et al., 2004), adaptation techniques were applied to IMT, following the ideas by (Kuhn and Mori, 1990) and adding cache language models (LM) and TMs to their system. In (Koehn and Schroeder, 2007), different ways to combine available data belonging to two different sources was explored; in (Bertoldi and Federico, 2009) similar experiments were performed, but considering only additional source data. In (Civera and Juan, 2007), alignment model mixtures were explored as a way of performing topic-specific adaptation. Other authors (Zhao et al., 2004; Sanchis-Trilles et al., 2009), have proposed the use of clustering in order to extract sub-domains of a large parallel corpus and build more specific LMs and TMs, which are re-combined in test time.

With respect to BA in SMT, the authors are not aware of any work up to the date that follows such paradigm. Nevertheless, there have been some recent approaches towards dealing with SMT from the Bayesian learning point of view. In (Zhang et al., 2008), Bayesian learning was applied for estimating word-alignments within a synchronous grammar.

3 Phrase-based SMT

One of the most popular instantiations of log-linear models in SMT are phrase-based (PB) models (Zens et al., 2002; Koehn et al., 2003). PB models allow to capture contextual information to learn translations for whole phrases instead of single words. The basic idea of PB translation is to segment the source sentence into phrases, then to translate each source phrase into a target phrase, and finally reorder the translated target phrases in order to compose the target sentence. For this purpose, phrase-tables are produced, in which a source phrase is listed together with several target phrases and the probability of translating the former into the latter. PB models were employed throughout this work.

Typically, the weights of the log-linear combination in Equation 3 are optimised by means of Minimum Error Rate Training (MERT) (Och, 2003). Such algorithm consists of two basic steps. First, n-best hypotheses are extracted for each one of the sentences of a given development set. Next, the optimum $\mathbf{\Lambda}$ is computed so that the best hypotheses in the n-best list, according to a reference translation and a given metric, are ranked higher within such n-best list. These two steps are repeated until convergence.

This approach has two main problems. On the

one hand, that it heavily relies on having a fair amount of data available as development set. On the other hand, that it *only* relies on the data in the development set. These two problems have as consequence that, if the development set made available to the system is not big enough, MERT will most likely become unstable and fail in obtaining an appropriate weight vector $\mathbf{\Lambda}$.

However, it is quite common to have a great amount of data available in a given domain, but only a small amount from the specific domain we are interested in translating. Precisely this scenario is appropriate for BA: under this paradigm, the weight vector $\mathbf{\Lambda}$ is *biased* towards the optimal one according to the adaptation set, while avoiding over-training towards such set by not forgetting the generality provided by the training set. Furthermore, recomputing $\mathbf{\Lambda}$ from scratch by means of MERT may imply a computational overhead which may not be acceptable in certain environments, such as SMT systems configured for online translation, IMT or Computer Assisted Translation, in which the final human user is waiting for the translations to be produced.

4 Bayesian adaptation for SMT

The main idea behind Bayesian learning (Duda et al., 2001; Bishop, 2006) is that model parameters are viewed as random variables having some kind of a priori distribution. Observing these random variables leads to a posterior density, which typically peaks at the optimal values of these parameters. Following the notation in Equation 1, previous statement is specified as

$$p(\mathbf{e}|\mathbf{f};T) = \int p(\mathbf{e},\theta|\mathbf{f};T)d\theta \quad (4)$$

where T represents the complete training set and θ are the model parameters.

However, since we are interested in Bayesian *adaptation*, we need to consider one training set T and one adaptation set A, leading to

$$p(\mathbf{e}|\mathbf{f};T,A) \approx \int p(\theta|T,A)p(\mathbf{e}|\mathbf{f},\theta)d\theta \quad (5)$$

In Equation 5, the integral over the complete parametric space forces the model to take into account all possible values of the model parameters, although the prior over the parameters implies that our model will prefer parameter values which are closer to our prior knowledge. Two assumptions have been made: first, that the output sentence $\mathbf{e}$ only depends on the model parameters (and not on the complete training and adaptation data). Second, that the model parameters do not depend on the actual input sentence $\mathbf{f}$. Such simplifications lead to a decomposition of the integral in two parts: the first one, $p(\theta|T,A)$ will assess how good the current model parameters are, and the second one, $p(\mathbf{e}|\mathbf{f},\theta)$, will account for the quality of the translation $\mathbf{e}$ given the current model parameters.

Then, the decision rule given in Equation 1 is redefined as

$$\hat{\mathbf{e}} = \underset{\mathbf{e}}{\operatorname{argmax}} \Pr(\mathbf{e}|\mathbf{f};T,A) \quad (6)$$

Operating with the probability of θ, we obtain:

$$p(\theta|T,A) = \frac{p(A|\theta;T)\,p(\theta|T)}{\int p(A|\theta)\,p(\theta|T)\,d\theta} \quad (7)$$

$$p(A|\theta;T) = \prod_{\forall a\in A} p(\mathbf{f}_a|\theta)\,p(\mathbf{e}_a|\mathbf{f}_a,\theta) \quad (8)$$

where the probability of the adaptation data has been assumed to be independent of the training data and has been modelled as the probability of each bilingual sample $(\mathbf{f}_a,\mathbf{e}_a) \in A$ being generated by our translation model.

Assuming that the model parameters depend on the training data and follow a normal distribution, we obtain

$$p(\theta|T) = \frac{1}{(2\pi)^{d/2}}\exp\left\{-\frac{1}{2}(\theta-\theta_T)^T(\theta-\theta_T)\right\} \quad (9)$$

where θ_T is the set of parameters estimated on the training set and the variance has been assumed to be bounded for all parameters. d is the dimensionality of θ.

Lastly, assuming that our translation model is a log-linear model as described in Equation 3 and that the only parameters we want to adapt are the log-linear weights:

$$p(\mathbf{e}|\mathbf{f},\theta) = \frac{\exp\sum_k \lambda_k\, f_k(\mathbf{f},\mathbf{e})}{\sum_{\mathbf{e}'}\exp\sum_k \lambda_k\, f_k(\mathbf{f},\mathbf{e}')} \quad (10)$$

where the model parameters θ have been instantiated to include only the log-linear weights $\mathbf{\Lambda}$.

Finally, combining Equations 8, 9 and 10, and considering only as model parameters the log-linear weights, we obtain:

$$p(\mathbf{e}|\mathbf{f};T,A) = \mathcal{Z}\int p(A|\mathbf{\Lambda};T)p(\mathbf{\Lambda}|T)p(\mathbf{e}|\mathbf{f},\mathbf{\Lambda})d\mathbf{\Lambda}$$

$$= \mathcal{Z}\int \prod_{\forall a\in A} \frac{\exp\sum_k \lambda_k\ f_k(\mathbf{f}_a,\mathbf{e}_a)}{\sum_{\mathbf{e}'}\exp\sum_k \lambda_k\ f_k(\mathbf{f}_a,\mathbf{e}')}\cdot$$

$$\exp\left\{-\frac{1}{2}(\mathbf{\Lambda}-\mathbf{\Lambda}_T)^T(\mathbf{\Lambda}-\mathbf{\Lambda}_T)\right\}\cdot$$

$$\frac{\exp\sum_k \lambda_k\ f_k(\mathbf{f},\mathbf{e})}{\sum_{\mathbf{e}'}\exp\sum_k \lambda_k\ f_k(\mathbf{f},\mathbf{e}')}\ d\mathbf{\Lambda} \qquad (11)$$

where $\mathcal{Z}$ is the denominator present in the previous equation and may be factored out because it does not depend on the integration variable. It has also been assumed that $p(\mathbf{f}_a|\theta)$ is uniform and can also be factored out.

5 Practical approximations

Although the integral described in Equation 11 is the right thing to do from the theoretical point of view, there are several issues which need to be treated first before implementing it.

Since computing the integral over the complete parametric space is computationally impossible in the case of SMT, we decided to perform a Monte Carlo like sampling of these parameters by assuming that the parameters follow a normal distribution centred in $\mathbf{\Lambda}_T$, the weight vector obtained from the training data. This sampling was done by choosing alternatively only one of the weights in $\mathbf{\Lambda}_T$, modifying it randomly within a given interval, and re-normalising accordingly. Equation 11 is approximated in practise as

$$p(\mathbf{e}|\mathbf{f};T,A) = \sum_{\Lambda_m\in MC(\Lambda_T)} p(A|\mathbf{\Lambda};T)p(\mathbf{\Lambda}|T)p(\mathbf{e}|\mathbf{f},\mathbf{\Lambda})$$

where $MC(\Lambda_T)$ is the set of Λ_m weights generated by the above-mentioned procedure.

There is still one issue when trying to implement Equation 11. The denominator within the components $p(A|\mathbf{\Lambda};T)$ and $p(\mathbf{e}|\mathbf{f},\mathbf{\Lambda})$ contains a sum over all possible sentences of the target language, which is not computable. For this reason, $\sum_{\mathbf{e}'}$ is approximated as the sum over all the hypothesis within a given n-best list. Moreover, instead of performing a full search of the best possible translation of a given input sentence, we will perform a rerank of the n-best list provided by the decoder according to Equation 11.

Typical state-of-the-art PB SMT systems do not guarantee complete coverage of all possible sentence pairs due to the great number of heuristic decisions involved in the estimation of the translation models. Moreover, out-of-vocabulary words may imply that the SMT model is unable to explain a certain bilingual sentence completely. Hence, $p(A|\mathbf{\Lambda};T)$ is approximated as

$$p(A|\mathbf{\Lambda};T) \approx \prod_{\forall a\in A} \frac{\exp\sum_k \lambda_k\ f_k(\mathbf{f}_a,\mathbf{e}_a^*)}{\sum_{\mathbf{e}'}\exp\sum_k \lambda_k f_k(\mathbf{f}_a,\mathbf{e}')} \qquad (12)$$

where $\mathbf{e}^*$ represents the best hypothesis the search algorithm is able to produce, according to a given translation quality measure. As in Equation 11, $p(\mathbf{f}_a|\theta)$ has been assumed uniform.

Once the normalisation factor within Equation 7 has been removed, and the above-mentioned approximations have been introduced, $p(\mathbf{e}|\mathbf{f};T,A)$ is no longer a probability. This fact cannot be underestimated, since it means that the terms $p(A|\mathbf{\Lambda};T)$ and $p(\mathbf{e}|\mathbf{f},\mathbf{\Lambda})$ on the one hand, and $p(\mathbf{\Lambda}|T)$ on the other, may have very different numeric ranges. For this reason, and in order to weaken the influence of this fact, we introduce a leveraging term δ, such that

$$p(\mathbf{e}|\mathbf{f};T,A) = \sum_{\Lambda_m\in MC(\Lambda_T)} \left(p(A|\mathbf{\Lambda};T)p(\mathbf{e}|\mathbf{f},\mathbf{\Lambda})\right)^{\frac{1}{\delta}} p(\mathbf{\Lambda}|T) \qquad (13)$$

Although there are other, more standard, ways of adding this leveraging term, we chose this one for numeric reasons.

6 Experiments

6.1 Experimental setup

Translation quality will be assessed by means of BLEU and TER scores. BLEU measures n-gram precision with a penalty for sentences that are too short (Papineni et al., 2001), whereas TER (Snover et al., 2006) is an error metric that

		Spanish	English
Training	Sentences	731K	
	Run. words	15.7M	15.2M
	Vocabulary	103K	64K
Development	Sentences	2K	
	Run. words	61K	59K
	OoV words	208	127

Table 1: Main figures of the Europarl corpus. *OoV* stands for Out of Vocabulary. K/M stands for thousands/millions of elements.

		Spanish	English
Test 2008	Sentences	2051	
	Run. words	50K	53K
	OoV. words	1247	1201
Test 2010	Sentences	2489	
	Run. words	62K	66K
	OoV. words	1698	1607

Table 2: Main figures of the News-Commentary test sets. *OoV* stands for Out of Vocabulary words with respect to the Europarl corpus.

computes the minimum number of edits required to modify the system hypotheses so that they match the references. Possible edits include insertion, deletion, substitution of single words and shifts of word sequences.

For computing $\mathbf{e}^*$ as described in Equation 12, TER was used, since BLEU implements a geometrical average which is zero whenever there is no common 4-gram between reference and hypothesis. Hence, it is not well suited for our purposes since the complete set of n-best candidates provided by the decoder can score zero.

As a first baseline system, we trained a SMT system on the Europarl Spanish–English training data, in the partition established in the Workshop on SMT of the NAACL 2006 (Koehn and Monz, 2006), using the training and development data provided that year. The Europarl corpus (Koehn, 2005) is built from the transcription of European Parliament speeches published on the web. Statistics are provided in Table 1.

We used the open-source MT toolkit Moses (Koehn et al., 2007)[1] in its default monotonic setup, and estimated the weights of the log-linear combination using MERT on the Europarl development set. A 5-gram LM with interpolation and Kneser-Ney smoothing (Kneser and Ney, 1995) was also estimated.

Since our purpose is to adapt the initial weight vector obtained during the training stage (i.e. the one obtained after running MERT on the Europarl development set), the tests sets provided for the 2008 and 2010 evaluation campaigns of the above-mentioned workshop (Table 2) were also used. These test sets, unlike the one provided in 2006, were extracted from a news data corpus, and can be considered out of domain if the system has been trained on Europarl data.

All the experiments displaying BA results were carried out by sampling a total of 100 random weights, according to preliminary investigation, following the procedure described in Section 5. For doing this, one single weight was added a random amount between 0.5 and -0.5, and then the whole $\mathbf{\Lambda}$ was re-normalised.

With the purpose of providing robustness to the results, every point in each plot of this paper constitutes the average of 10 repetitions, in which the adaptation data was randomly drawn from the News-Commentary test set 2008.

6.2 Comparison between BA and MERT

The effect of increasing the number of adaptation samples made available to the system was investigated. The adaptation data was used either for estimating $\mathbf{\Lambda}$ using MERT, or as adaptation sample for our BA technique. Results can be seen in Figure 1. The δ scaling factor described in Equation 13 was set to 8. As it can be seen, the BA adaptation technique is able to improve consistently the translation quality obtained by the non-adapted system, both in terms of BLEU and TER. These improvements are quite stable even with as few as 10 adaptation samples. This result is very interesting, since re-estimating $\mathbf{\Lambda}$ by means of MERT is only able to yield improvements when provided with at least 100 adaptation samples, displaying a very chaotic behaviour until that point.

In order to get a bit more insight about this chaotic behaviour, confidence interval sizes are shown in Figure 2, at a 95% confidence level, resulting of the repetitions described above. MERT yields very large confidence intervals (as large as 10 TER/BLEU points for less than 100 samples), turning a bit more stable from that point on, where the size of the confidence interval converges slowly to 1 TER/BLEU point. In contrast,

[1] Available from http://www.statmt.org/moses/

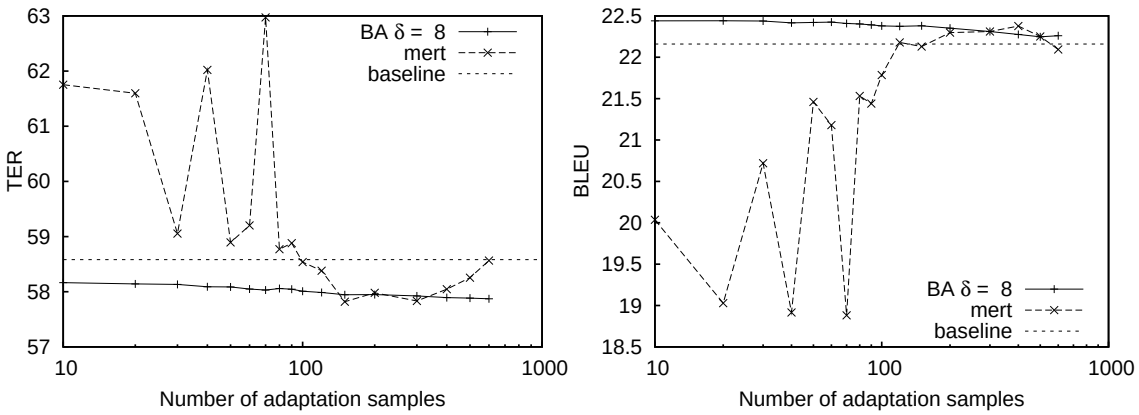


Figure 1: Comparison of translation quality, as measured by BLEU and TER, for baseline system, adapted systems by means of BA and MERT. Increasing number of samples is considered.

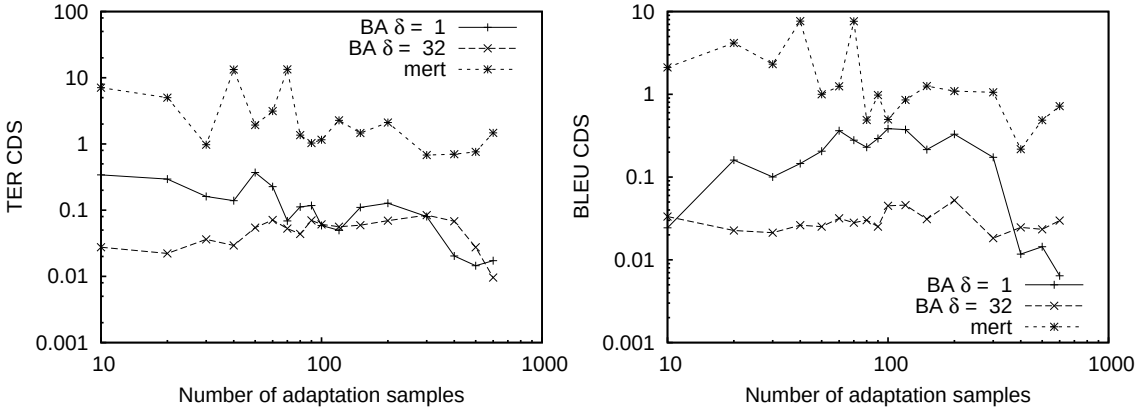


Figure 2: Confidence interval sizes (CDS) for MERT and two BA systems, for different number of adaptation samples. For visibility purposes, both axes are in logarithmic scale.

our BA technique yields very small confidence intervals, about half a TER/BLEU point in the worst case, with only 10 adaptation samples. This is worth emphasising, since estimating $\mathbf{\Lambda}$ by means of MERT when very few adaptation data is available may improve the final translation quality, but may also degrade it to a much larger extent. In contrast, our BA technique shows stable and reliable improvements from the very beginning. Precisely under such circumstances is an adaptation technique useful: when the amount of adaptation data is small. In other cases, the best thing one can do is to re-estimate the model parameters from scratch.

Example translations, extracted from the experiments detailed above, are shown in Figure 5.

6.3 Varying δ

So as to understand the role of scaling factor δ, results obtained varying it are shown in Figure 3. Several things should be noted about these plots:

- Increasing δ leads to smoother adaptation curves. This is coherent with the confidence interval sizes shown in Figure 1.
- Smaller values of δ lead to a slight degradation in translation quality when the amount of adaptation samples becomes larger. The reason for this can be explained by looking at Equation 13. Since $p(A|\mathbf{\Lambda};T)$ is implemented as a product of probabilities, the more adaptation samples the smaller becomes $p(A|\mathbf{\Lambda};T)$, and a higher value of δ is needed to compensate this fact. This suggests the need of a δ which depends on the size of the adaptation sample.
- Larger values of δ do not suffer the problem described above, but yield smaller improvements in terms of translation quality for smaller amount of samples.

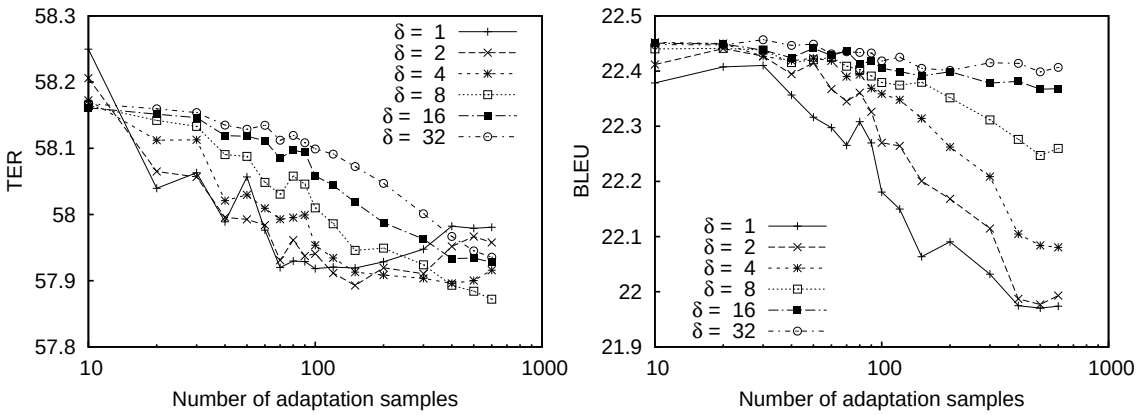


Figure 3: Translation quality comparison for different δ values and number of adaptation samples.

It might seem odd that translation quality as measured by BLEU drops almost constantly as the number of adaptation samples increases. However, it must be noted that the BA technique implemented is set to optimise TER, and not BLEU. Analysing the BLEU scores obtained, we realised that the n-gram precision does increase, but the final BLEU score drops because of a worsening brevity penalty, which is not taken into account when optimising the TER score.

6.3.1 Increasing the n-best order

The effect of increasing the order of n-best considered was also analysed. In order to avoid an overwhelming amount of results, only those obtained when considering 100 adaptation samples are displayed in Figure 4. As it can be seen, TER drops monotonically for all δ values, until about 800, where it starts to stabilise. Similar behaviour is observed in the case of BLEU, although depending on δ the curve shows an improvement or a degradation. Again, this is due to the brevity penalty, which TER does not implement, and which induces this inverse correlation between TER and BLEU when optimising TER.

7 Conclusions and future work

We have presented a Bayesian theoretical framework for adapting the parameters of a SMT system. We have derived the equations needed to implement BA of the log-linear weights of a SMT system, and present promising results with a state-of-the-art SMT system using standard corpora in SMT. Such results prove that the BA framework can be very effective when adapting the mentioned weights. Consistent improvements are obtained over the baseline system with as few as 10 adaptation samples. The BA technique implemented is able to yield results comparable with a complete re-estimation of the parameters even when the amount of adaptation data is sufficient for such re-estimation to be feasible. Experimental results show that our adaptation technique proves to be much more stable than MERT, which relies very heavily on the amount of adaptation data and turns very unstable whenever few adaptation samples are available. It should be emphasised that an adaptation technique, by nature, is only useful whenever few adaptation data is available, and our technique proves to behave well in such context.

Intuitively, the BA technique presented needs first to compute a set of random weights, which are the result of sampling a gaussian distribution whose mean is the best weight vector obtained in training. Then, each hypothesis of a certain test source sentence is rescored according to the following three components:

- The probability of the adaptation corpus under each specific random weight
- The probability of such random weight according to a prior over the weight vector
- The probability of the current hypothesis under those weights

Concerning computational time, our adaptation technique can easily be implemented within the decoder itself, without any significant increase in computational complexity. We consider this im-

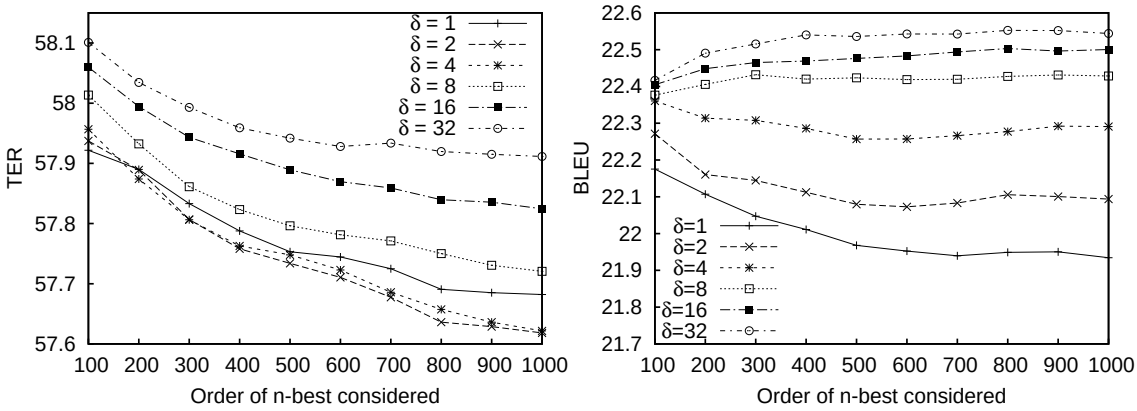


Figure 4: Translation quality for different δ values and n-best sizes considered in the BA system.

source	en afganistán , barack obama espera que se repita el milagro .
reference	barack obama hopes that , in afghanistan , the miracle will repeat itself .
baseline	in afghanistan , barack obama waiting to be repeated the miracle .
BA s10	in afghanistan , barack obama expected to repeat the miracle .
BA s600	in afghanistan , barack obama expected to repeat the miracle .
MERT s10	in afghanistan , barack obama expected to repeat of the miracle .
MERT s600	in afghanistan , barack obama hopes that a repetition of the miracle .
source	al final todo fue más rpido de lo que se pensó .
reference	it all happened a lot faster than expected .
baseline	at the end of all was more quickly than we thought .
BA s10	ultimately everything was more quickly than we thought .
BA s600	ultimately everything was more quickly than we though .
MERT s10	the end all was quicker than i thought .
MERT s600	ultimately everything was quicker than i thought .

Figure 5: Example of translations found in the corpus. `s10` means that only 10 adaptation samples were considered, whereas `s600` means that 600 were considered.

portant, since it implies that rerunning MERT for each adaptation set is not needed, and this is important whenever the final system is set up in an on-line environment.

The derivation presented here can be easily extended in order to adapt the feature functions of the log-linear model (i.e. not the weights). This is bound to have a more important impact on translation quality, since the amount of parameters to be adapted is much higher. We plan to address this issue in future work.

In addition, very preliminary experiments show that, when considering reordering, the advantages described here are larger.

A preliminary version of the present paper was accepted at the Joint IAPR International Workshops on Structural and Syntactic Pattern Recognition and Statistical Techniques in Pattern Recognition 2010. The main contributions of the present paper constitute more extensive experiments, which have been conducted on standard SMT corpora. Furthermore, in this paper we present the results of adding the leveraging term δ, of applying a random, Monte-Carlo like weight sampling (which was not done previously), and an extensive analysis of the effect of varying the order of n-best considered.

We also plan to implement Markov Chain Monte Carlo for sampling the parameters, and analyse the effect of combining the in-domain and out of domain data for MERT. Such results were not included here for time constraints.

Acknowledgments

This paper is based upon work supported by the EC (FEDER/FSE) and the Spanish MICINN under the MIPRCV "Consolider Ingenio 2010" program (CSD2007-00018) and the iTrans2 (TIN2009-14511) project. Also supported by the Spanish MITyC under the erudito.com (TSI-020110-2009-439) project and by the Generalitat Valenciana under grant Prometeo/2009/014.

The authors would like to thank the anonimous reviewers for their constructive comments.

References

Barrachina, S., O. Bender, F. Casacuberta, J. Civera, E. Cubel, S. Khadivi, A. Lagarda H. Ney, J. Tomás, and E. Vidal. 2009. Statistical approaches to computer-assisted translation. *Computational Linguistics*, 35(1):3–28.

Bertoldi, N. and M. Federico. 2009. Domain adaptation in statistical machine translation with monolingual resources. In *Proc. of EACL WMT*.

Bishop, C. M. 2006. *Pattern Recognition and Machine Learning*. Springer.

Brown, P.F., S.A. Della Pietra, V.J. Della Pietra, and R.L. Mercer. 1993. The mathematics of machine translation. In *Computational Linguistics*, volume 19, pages 263–311, June.

Civera, J. and A. Juan. 2007. Domain adaptation in statistical machine translation with mixture modelling. In *Proc. of ACL WMT*.

Duda, R., P. Hart, and D. Stork. 2001. *Pattern Classification*. Wiley-Interscience.

Kneser, R. and H. Ney. 1995. Improved backing-off for m-gram language modeling. *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, II:181–184, May.

Koehn, P. and C. Monz, editors. 2006. *Proc. on the Workshop on SMT*. Association for Computational Linguistics, June.

Koehn, P. and J. Schroeder. 2007. Experiments in domain adaptation for statistical machine translation. In *Proc. of ACL WMT*.

Koehn, P., F. J. Och, and D. Marcu. 2003. Statistical phrase-based translation. In *Proc. HLT/NAACL'03*, pages 48–54.

Koehn et al., P. 2007. Moses: Open source toolkit for statistical machine translation. In *Proc. of the ACL Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic.

Koehn, P. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT Summit*.

Kuhn, R. and R. De Mori. 1990. A cache-based natural language model for speech recognition. *IEEE Transactions on PAMI*, 12(6):570–583.

Nepveu, L., G. Lapalme, P. Langlais, and G. Foster. 2004. Adaptive language and translation models for interactive machine translation. In *Proc. of EMNLP*.

Och, F. and H. Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. In *Proc. of the ACL'02*, pages 295–302.

Och, F.J. 2003. Minimum error rate training for statistical machine translation. In *Proc. of Annual Meeting of the ACL*, July.

Papineni, K., S. Roukos, and T. Ward. 1998. Maximum likelihood and discriminative training of direct translation models. In *Proc. of ICASSP*, pages 189–192.

Papineni, K., A. Kishore, S. Roukos, T. Ward, and W. Jing Zhu. 2001. Bleu: A method for automatic evaluation of machine translation. In *Technical Report RC22176 (W0109-022)*.

Sanchis-Trilles, G., M. Cettolo, N. Bertoldi, and M. Federico. 2009. Online Language Model Adaptation for Spoken Dialog Translation. In *Proc. of IWSLT*, Tokyo.

Snover, Matthew, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proc. of AMTA'06*.

Zens, R., F.J. Och, and H. Ney. 2002. Phrase-based statistical machine translation. In *Proc. of KI'02*, pages 18–32.

Zhang, Hao, Chris Quirk, Robert C. Moore, and Daniel Gildea. 2008. Bayesian learning of non-compositional phrases with synchronous parsing. In *Proceedings of ACL-08: HLT*, pages 97–105, Columbus, Ohio, June. Association for Computational Linguistics.

Zhao, B., M. Eck, and S. Vogel. 2004. Language model adaptation for statistical machine translation with structured query models. In *Proc. of CoLing*.

A Global Relaxation Labeling Approach to Coreference Resolution

Emili Sapena, Lluís Padró and Jordi Turmo*
TALP Research Center
Universitat Politècnica de Catalunya
{esapena, padro, turmo}@lsi.upc.edu

Abstract

This paper presents a constraint-based graph partitioning approach to coreference resolution solved by relaxation labeling. The approach combines the strengths of groupwise classifiers and chain formation methods in one global method. Experiments show that our approach significantly outperforms systems based on separate classification and chain formation steps, and that it achieves the best results in the state of the art for the same dataset and metrics.

1 Introduction

Coreference resolution is a natural language processing task which consists of determining the *mentions* that refer to the same entity in a text or discourse. A mention is a noun phrase referring to an entity and includes named entities, definite noun phrases, and pronouns. For instance, "Michael Jackson" and "the youngest of Jackson 5" are two mentions referring to the same entity.

A typical machine learning-based coreference resolution system usually consists of two steps: (i) classification, where the system evaluates the *coreferentiality* of each pair or group of mentions, and (ii) formation of chains, where given the confidence values of the previous classifications the system forms the coreference chains.

Research supported by the Spanish Science and Innovation Ministry, via the KNOW2 project (TIN2009-14715-C04-04) and from the European Community's Seventh Framework Programme (FP7/2007-2013) under Grant Agreement number 247762 (FAUST)

Regarding the classification step, pioneer systems developed were based on *pairwise* classifiers. Given a pair of mentions, the process generates a feature vector and feeds it to a classifier. The resolution is done by considering each mention of the document as *anaphor*[1] and looking backward until the *antecedent* is found or the beginning of the document is reached (Aone and Bennett, 1995; McCarthy and Lehnert, 1995; Soon et al., 2001).

A first approach towards *groupwise* classifiers is the twin-candidate model (Yang et al., 2003). The model faces the problem as a competition between two candidates to be the antecedent of the anaphor into account. Each candidate mention is compared with all the others in a round robin contest. Following the *groupwise* approach, rankers consider all the possible antecedent mentions at once (Denis and Baldridge, 2008). Rankers can obtain more accurate results due to a more informed context where all candidate mentions are considered at the same time.

Coreference chains are formed after classification. Many systems form the chains by joining each positively-classified pair (i.e. *single-link*) or with simple improvements such as linking an anaphor only to its antecedent with maximum confidence value (Ng and Cardie, 2002).

Some works propose more elaborated methods than single-link for chain formation. The approaches used are Integer Linear Programming

[1] Typically a pair of coreferential mentions m_i and m_j ($i < j$) are called antecedent and anaphor respectively, though m_j may not be anaphoric.

Coling 2008: Poster Volume, pages 1086–1094,
Beijing, August 2010

(ILP) (Denis and Baldridge, 2007; Klenner and Ailloud, 2009; Finkel and Manning, 2008), graph partitioning (Nicolae and Nicolae, 2006), and clustering (Klenner and Ailloud, 2008). The main advantage of these types of *post-processes* is the enforcement of transitivity sorting out the contradictions that the previous classification process may introduce.

Although chain formation processes search for global consistency, the lack of contextual information in the classification step is propagated forward. Few works try to overcome the limitations of keeping classification and chain formation apart. Luo et al. (2004) search the most probable path comparing each mention with the partial-entities formed so far using a Bell tree structure. McCallum and Wellner (2005) propose a graph partitioning cutting by distances, with the peculiarity that distances are learned considering coreferential chains of the labeled data instead of pairs. Culotta et al. (2007) combine a groupwise classifier with a clustering process in a First-Order probabilistic model.

The approach presented in this paper follows the same research line of joining group classification and chain formation in the same step. Concretely, we propose a graph representation of the problem solved by a relaxation labeling process, reducing coreference resolution to a graph partitioning problem given a set of constraints. In this manner, decisions are taken considering the whole set of mentions, ensuring consistency and avoiding that classification decisions are independently taken. Our experimental results on the ACE dataset show that our approach outperforms systems based on separate classification and chain formation steps, and that it achieves the best results in the state of the art for the same dataset and metrics.

The paper is organized as follows. Section 2 describes the graph representation of the task. Section 3 explains the use of relaxation labeling algorithm and the machine learning process. Finally, experiments and results are explained in Section 4 before paper is concluded.

2 Graph Representation

Let $G = G(V, E)$ be an undirected graph where V is a set of vertices and E a set of edges. Let $\mathbf{m} = (m_1, ..., m_n)$ be the set of mentions of a document with n mentions to resolve. Each mention m_i in the document is represented as a vertex $v_i \in V$. An edge $e_{ij} \in E$ is added to the graph for pairs of vertices (v_i, v_j) representing the possibility that both mentions corefer. The list of adjacent vertices of a vertex v_i is $A(v_i)$.

Let C be our set of constraints. Given a pair of mentions (m_i, m_j), a subset of constraints $C_{ij} \subseteq C$ restrict the compatibility of both mentions. C_{ij} is used to compute the weight value of the edge connecting v_i and v_j. Let $w_{ij} \in W$ be the weight of the edge e_{ij}:

$$w_{ij} = \sum_{k \in C_{ij}} \lambda_k f_k(m_i, m_j) \quad (1)$$

where $f_k(\cdot)$ is a function that evaluates the constraint k. And λ_k is the weight associated to the constraint k (λ_k and w_{ij} can be negative).

In our approach, each vertex (v_i) in the graph is a variable (v_i) for the algorithm. Let L_i be the number of different values (labels) that are possible for v_i. The possible labels of each variable are the partitions that the vertex can be assigned. Note that the number of partitions (entities) in a document is unknown, but it is at most the number of vertices (mentions), because in a extreme case, each mention in a document could be referring to a different entity. A vertex with index i can be in the first i partitions (i.e. $L_i = i$).

Each combination of labelings for the graph vertices is a partitioning (Ω). The resolution process searches the partitioning Ω^* which optimizes the goodness function $F(\Omega, W)$, which depends on the edge weights W. In this manner, Ω^* is optimal if:

$$F(\Omega^*, W) \geq F(\Omega, W), \forall \Omega \quad (2)$$

The next section describes the algorithm used in the resolution process.

3 Relaxation Labeling

Relaxation labeling (Relax) is a generic name for a family of iterative algorithms which perform

function optimization, based on local information. The algorithm has been widely used to solve NLP problems such as PoS-tagging (Màrquez et al., 2000), chunking, knowledge integration, and Semantic Parsing (Atserias, 2006).

Relaxation labeling solves our weighted constraint satisfaction problem dealing with the edge weights. In this manner, each vertex is assigned to a partition satisfying as many constraints as possible. To do that, the algorithm assigns a probability for each possible label of each variable. Let $\mathbf{H} = (\mathbf{h}^1, \mathbf{h}^2, \ldots, \mathbf{h}^n)$ be the weighted labeling to optimize, where each $\mathbf{h}^i$ is a vector containing the probability distribution of v_i, that is: $\mathbf{h}^i = (h_1^i, h_2^i, \ldots, h_{L_i}^i)$. Given that the resolution process is iterative, the probability for label l of variable v_i at time step t is $h_l^i(t)$, or simply h_l^i when the time step is not relevant.

The support for a pair variable-label (S_{il}) expresses how compatible is the assignment of label l to variable v_i considering the labels of adjacent variables and the edge weights. Although several support functions may be used (Torras, 1989), we chose the following one, which defines the support as the sum of the edge weights that relate variable v_i with each adjacent variable v_j multiplied by the weight for the same label l of v_j:

$$S_{il} = \sum_{j \in A(v_i)} w_{ij} \times h_l^j \qquad (3)$$

where w_{ij} is the edge weight obtained in Equation 1. In our version of the algorithm, $A(v_i)$ is the list of adjacent vertices of v_i but only including the ones with an index $k < i$. Consequently, the weights only have influence in one direction which is equivalent to using a directed graph. Although the proposed representation is based on a general undirected graph, preliminary experiments showed that using directed edges yields higher perfomance in this particular problem.

The aim of the algorithm is to find a weighted labeling such that global consistency is maximized. Maximizing global consistency is defined as maximizing the average support for each variable. Formally, $\mathbf{H}^*$ is a consistent labeling if:

Initialize:
  $H := H_0$,

Main loop:
  repeat
  For each variable v_i
    For each possible label l for v_i
      $S_{il} = \sum_{j \in A(v_i)} w_{ij} \times h_l^j$
    End for
    For each possible label l for v_i
      $h_l^i(t+1) = \frac{h_l^i(t) \times (1 + S_{il})}{\sum_{k=1}^{L_i} h_k^i(t) \times (1 + S_{ik})}$
    End for
  End for
  Until no more significant changes

Figure 1: Relaxation labeling algorithm

$$\sum_{l=1}^{L_i} h_l^{*i} \times S_{il} \geq \sum_{l=1}^{L_i} h_l^i \times S_{il} \quad \forall \mathbf{h}, \forall i \qquad (4)$$

A partitioning Ω is directly obtained from the weighted labeling $\mathbf{H}$ assigning to each variable the label with maximum probability. The supports and the weighted labeling depend on the edge weights (Equation 3). To satisfy Equation 4 is equivalent to satisfy Equation 2. Many studies have been done towards the demonstration of the consistency, convergence and cost reduction advantages of the relaxation algorithm (Rosenfeld et al., 1976; Hummel and Zucker, 1987; Pelillo, 1997). Although some of the conditions required by the formal demonstrations are not fulfilled in our case, the presented algorithm –that forces a stop after a number of iterations– has proven useful for practical purposes.

Figure 1 shows the pseudo-code of the relaxation algorithm. The process updates the weights of the labels in each step until convergence. The convergence is met when no more significant changes are done in an iteration. Specifically, when the maximum change in an update step ($max_{i,l}(|h_l^i(t+1) - h_l^i(t)|)$) is lower than a parameter ϵ, a small value (0.001 in our experiments), or a fixed number of iterations is reached (2000 in our experiments). Finally, the assigned label for a variable is the one with the highest weight. Figure 2 shows a representation.

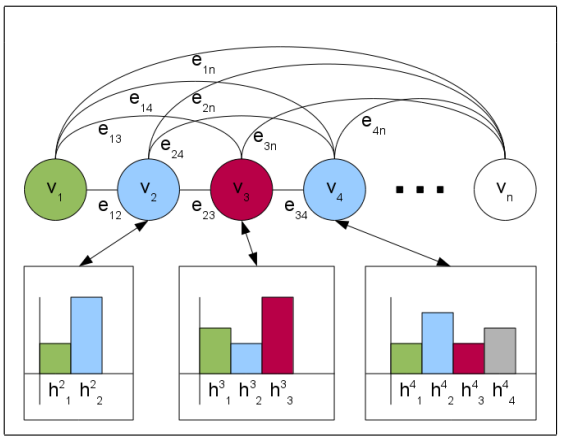


Figure 2: Representation of Relax. The vertices representing mentions are connected by weighted edges e_{ij}. Each vertex has a vector $\mathbf{h}^i$ of probabilities to belong to different partitions. The figure shows $\mathbf{h}^2$, $\mathbf{h}^3$ and $\mathbf{h}^4$.

DIST: Distance between m_i and m_j in sentences: number
DIST_MEN: Distance between m_i and m_j in mentions: number
APPOSITIVE: One mention is in apposition with the other: y,n
I/J_IN_QUOTES: $m_{i/j}$ is in quotes or inside a NP or a sentence in quotes: y,n
I/J_FIRST: $m_{i/j}$ is the first mention in the sentence: y,n
I/J_DEF_NP: $m_{i/j}$ is a definitive NP: y,n
I/J_DEM_NP: $m_{i/j}$ is a demonstrative NP: y,n
I/J_INDEF_NP: $m_{i/j}$ is an indefinite NP: y,n
STR_MATCH: String matching of m_i and m_j: y,n
PRO_STR: Both are pronouns and their strings match: y,n
PN_STR: Both are proper names and their strings match: y,n
NONPRO_STR: String matching like in Soon et al. (2001) and mentions are not pronouns: y,n
HEAD_MATCH: String matching of NP heads: y,n
NUMBER: The number of both mentions match: y,n,u
GENDER: The gender of both mentions match: y,n,u
AGREEMENT: Gender and number of both mentions match: y,n,u
I/J_THIRD_PERSON: $m_{i/j}$ is 3rd person: y,n
PROPER_NAME: Both mentions are proper names: y,n,u
I/J_PERSON: $m_{i/j}$ is a person (pronoun or proper name in a list): y,n
ANIMACY: Animacy of both mentions match (persons, objects): y,n
I/J_REFLEXIVE: $m_{i/j}$ is a reflexive pronoun: y,n
I/J_TYPE: $m_{i/j}$ is a pronoun (p), entity (e) or nominal (n)
NESTED: One mention is included in the other: y,n
MAXIMALNP: Both mentions have the same NP parent or they are nested: y,n
I/J_MAXIMALNP: $m_{i/j}$ is not included in any other mention: y,n
I/J_EMBEDDED: $m_{i/j}$ is a noun and is not a maximal NP: y,n
BINDING: Conditions B and C of binding theory: y,n
SEMCLASS: Semantic class of both mentions match: y,n,u (the same as Soon et al. (2001))
ALIAS: One mention is an alias of the other: y,n,u (only entities, else unknown)

Figure 3: Feature functions used

3.1 Constraints

The performance of the resolution process depends on the edge weights obtained by a set of weighted constraints (Equation 1). Any method or combination of methods to generate constraints can be used. For example, a set of constraints handwritten by linguist experts can be added to another automatically obtained set.

This section explains the automatic constraint generation process carried out in this work, using a set of feature functions and a training corpus. Màrquez et al. (2000) have successfully used similar processes to acquire constraints for constraint satisfaction algorithms.

Each pair of mentions (m_i, m_j) in a training document is evaluated by a set of feature functions (Figure 3). The values returned by these functions form a positive example when the pair of mentions corefer, and a negative one otherwise. Three specialized models are constructed depending on the type of anaphor mention (m_j) of the pair: pronoun, named entity or nominal.

For each specialized model, a decision tree (DT) is generated and a set of rules is extracted with C4.5 rule-learning algorithm (Quinlan, 1993). These rules are our set of constraints. The C4.5rules algorithm generates a set of rules for each path from the learnt tree. It then generalizes the rules by dropping conditions.

The weight assigned to a constraint (λ_k) is its precision over the training data (P_k), but shifted to be zero-centered: $\lambda_k = P_k - 0.5$.

3.2 Pruning

Analyzing the errors of development experiments, we have found two main error patterns that can be solved by a pruning process. First, the contribution of the edge weights for the resolution depends on the size of the document. And second, many weak edge weights may sum up to produce a bias in the wrong direction.

The weight of an edge depends on the weights assigned for the constraints which apply to a pair of mentions according to Equation 1. Each vertex is adjacent to all the other vertices. This produces that the larger the number of adjacencies, the smaller the influence of a constraint is. A consequence is that resolution for large and short documents has different results.

Many works have to deal with similar problems, specially the ones looking backward for antecedents. The larger the document, the more pos-

sible antecedents the system has to classify. This problem is usually solved looking for antecedents in a window of few sentences, which entails an evident limitation of recall.

Regarding the weak edge weights, it is notable that some kind of mention pairs are very weakly informative. For example, the pairs (pronoun, pronoun). Many stories have a few main characters which monopolize the pronouns of the document. This produces many positive training examples for pairs of pronouns matching in gender and person, which may lead the algorithm to produce large coreferential chains joining all these mentions even for stories where there are many different characters. For example, we have found in the results of some documents a huge coreference chain including every pronoun "he". This is because a pair of mentions ("he", "he") is usually linked with a small positive weight. Although the highest adjacent edge weight of a "he" mention may link with the correct antecedent, the sum of several edge weights linking the mention with other "he" causes the problem.

A pruning process is perfomed solving both problems and reducing computational costs from $O(n^3)$ to $O(n^2)$. For each vertex's adjacency list $A(v_i)$, only a maximum of N edges remain and the others are pruned. Concretely, the N/2 edges with largest positive weight and the N/2 with largest negative weight. The value of N is empirically chosen by maximizing performances over training data. On the one hand, the pruning forces the maximum adjacency to be constant and the contribution of the edge weights does not depend on the size of the document. On the other hand, most edges of the less informative pairs are discarded avoiding further confusion. There are no limitations in distance or other restrictions which may cause a loss of recall.

3.3 Initial State

The initial state of the vertices define the *a priori* probabilities for each vertex to be in each partition. There are several possible initial states. In the case where no prior information is available, a random or uniformly distributed state is commonly used. However, a well-informed initial state should drive faster the relaxation process to a better solution. This section describes the well-informed initial state chosen in our approach and the random one. Both are compared in the experiments (Section 4.2).

The well-informed initial state favors the creation of new chains. Variable v_i has $L_i = i$ possible values while variable v_{i+1} has $L_i + 1$. The probability distribution of v_{i+1} is equiprobable for values from 1 to L_i but it is the double for the probability to start a new chain $L_i + 1$.

$$h^i_l = \frac{1}{L_i+1}, \qquad \forall l = 1..L_i - 1$$
$$h^i_{L_i} = \frac{2}{L_i+1}$$

Pronouns do not follow this distribution but a totally equiprobable one, given that they are usually anaphoric.

$$h^i_l = \frac{1}{L_i}, \qquad \forall l = 1..L_i$$

This configuration enables the resolution process to determine as singletons the mentions for which little evidence is available. This small difference between initial probability weights is also introduced in order to avoid exceptional cases where all support values contribute with the same value.

The random initial state is also used in our experiments to test that our proposed configuration is better-informed than random. Given the equiprobability state, we add a random value to each probability to be in a partition:

$$h^i_l = \frac{1}{L_i} + \epsilon_{il}, \qquad \forall l = 0..L_i$$

where ϵ_{il} is a random value $\frac{-1}{2L_i} \leq \epsilon_{il} \leq \frac{1}{2L_i}$. These little random differences may help the algorithm to avoid local minima.

3.4 Reordering

The vertices of the graph would usually be placed in the same order as the mentions are found in the document (*chronological*). In this manner, v_i corresponds to m_i. However, as suggested by Luo (2007), there is no need to generate the model following that order. In our approach, the first variables have a lower number of possible labels. Moreover, an error in the first variables has more influence on the performance than an error in the later ones. Placing named entities at the beginning is reasonably to expect that is helpful for the algorithm, given that named entities are usually the most informative mentions.

	Tokens	Mentions	Entities
bnews train	66627	9937	4408
bnews test	17463	2579	1040
npaper train	68970	11283	4163
npaper test	17404	2483	942
nwire train	70832	10693	4297
nwire test	16772	2608	1137

Figure 4: Statistics about ACE-phase02

Suppose we have three mentions appearing in this order somewhere in a document: "A. Smith", "he", "Alice Smith". For proximity, mention "he" may tend to link with "A. Smith". Then, the third mention "Alice Smith" clearly is the whole name of "A. Smith" but the gender with "he" does not agree. Given that our implementation acts like a directed graph only looking backward (see Section 3), mention "he" won't change its tendency and it may cause a split in the "Alice Smith" coreference chain. However, having named entities in first place and pronouns at the end, enables the mention "he" to determine that "A. Smith" and "Alice Smith" having the same label are not good antecedents.

Reordering only affects on the number of possible labels of the variables and the list of adjacencies $A(v_i)$. The chronological order of the document is taken into account by the constraints regardless of the graph representation. Our experiments confirm (Section 4) that placing first named entity mentions, then nominal mentions and finally the pronouns, the precision increases considerably. Inside of each of these groups, the order is the same order of the document.

4 Experiments and Results

We evaluate our approach to coreference resolution using ACE-phase02 corpus, which is composed of three sections: Broadcast News (BNEWS), Newswire (NWIRE) and Newspaper (NPAPER). Each section is in turn composed of a training set and a test set. Figure 4 shows some statistics about this corpus.

In our experiments, we consider the *true mentions* of ACE. This is because our focus is on evaluating pairwise approach versus the graph partitioning approach and also comparing them to some state-of-the-art approaches which also use true mentions. Moreover, details on mention identifier systems and their performances are rarely published by the systems based on automatic identification of mentions and it difficults the comparison.

To evaluate our system we use CEAF (Luo, 2005) and B^3 (Bagga and Baldwin, 1998). CEAF is computed based on the best one-to-one map between key coreference chains and response ones. We use the mention-based similarity metric which counts the number of common mentions shared by key coreference chains and response ones. As we are using *true mentions* for the experiments, precision, recall and F_1 are the same value and only F_1 is shown. B^3 scorer is used for comparison reasons. B^3 algorithm looks at the presence/absence of mentions for each entity in the system output. Precision and recall numbers are computed for each mention, and the average gives the final precision and recall numbers.

MUC scorer (Vilain et al., 1995) is not used in our experiments. Although it has been widely used in the state of the art, we consider the newer metrics have overcome some MUC limitations (Bagga and Baldwin, 1998; Luo, 2005; Klenner and Ailloud, 2008; Denis and Baldridge, 2008).

Our preprocessing pipeline consists of FreeLing (Atserias et al., 2006) for sentence splitting and tokenization, SVMTool (Gimenez and Marquez, 2004) for part of speech tagging and BIO (Surdeanu et al., 2005) for named entity recognition and classification. No lemmatization neither syntactic analysis are used.

4.1 Baselines

4.1.1 DT with automatic feature selection

The baseline developed in our work is based on Soon et al. (2001) with the improvements of Ng and Cardie (2002), which uses a Decision Tree (DT). Many research works use the same references in order to evaluate possible improvements done by their new models or by the incorporation of new features.

The features used in the baseline are the same than those used in our proposed system (Figure 3). However, some features are noisy and many others have redundancy which causes low performances using DTs. In order to select the best set

	bnews	npaper	nwire	Global			
Metric:	CEAF			CEAF	B^3		
Model	F_1	F_1	F_1	F_1	P	R	F_1
DT	60.6	57.8	60.5	59.7	61.0	74.1	66.9
DT Hill	67.8	61.6	65.0	**64.8**	74.7	69.8	**72.2**

Table 1: Results ACE-phase02. Comparing baselines based on Decision Trees.

	bnews	npaper	nwire	Global			
Metric:	CEAF			CEAF	B^3		
Model	F_1	F_1	F_1	F_1	P	R	F_1
DT	60.6	59.5	64.7	61.7	63.3	74.7	68.5
DT + ILP	62.8	60.3	63.7	**62.5**	72.4	69.2	**70.7**
DT Hill	67.8	63.2	67.2	**66.5**	76.8	71.0	**73.8**
DT Hill + ILP	67.6	63.5	66.7	66.3	80.0	68.3	73.7
Relax	69.5	68.3	73.0	**70.4**	86.5	67.9	**76.1**

Table 2: Results on documents shorter than 200 mentions of ACE-phase02

of features a Hill Climbing process has been performed doing a five-fold cross-validation over the training corpus. A similar feature selection process has been done by Hoste (2005).

The Hill Climbing process starts using the whole set of features. A cross-validation is done (un)masking each feature. The (un)masked feature with more improvement is (added to) removed from the set. The process is repeated until an iteration without improvements is reached.

Note that this optimization process is biased by the metric used to evaluate each feature combination. We use CEAF in our experiments, which encourages precision and consistency.

4.1.2 Integer Linear Programming

The second baseline developed forms the coreference chains given the output of the pair classification of the first baseline. A set of binary variables (x_{ij}) symbolize whether pairs of mentions (m_i, m_j) corefer ($x_{ij} = 1$) or not ($x_{ij} = 0$). An objective function is defined as follows:

$$min \sum_{i<j} -log(Pc_{ij})x_{ij} - log(1 - Pc_{ij})(1 - x_{ij})$$

where Pc_{ij} is the confidence value of mentions m_i and m_j to corefer obtained by the pair classifier. The minimization of the objective function is done by Integer Linear Programming (ILP) in a similar way to (Klenner, 2007; Denis and Baldridge, 2007; Finkel and Manning, 2008). In order to keep consistency in the results, which is the goal of this *post-process*, a set of *triangular* constraints is required. For each three mentions with indexes $i < j < k$ the corresponding variables have to satisfy three constraints:

- $x_{ik} \geq x_{ij} + x_{jk} - 1$
- $x_{ij} \geq x_{ik} + x_{jk} - 1$
- $x_{jk} \geq x_{ij} + x_{ik} - 1$

This implies that this model needs, for a document with n mentions, $\frac{1}{2}n(n-1)$ variables and $\frac{1}{2}n(n-1)(n-2)$ constraints to assure consistency[2]. This is an important limitation with a view to scalability. In our experiments only documents shorter than 200 mentions can be solved by this baseline due to its computational cost.

4.2 Experiments

Four experiments have been done in order to evaluate our proposed approach. This section describes and analyzes the results of each experiment. Finally, our performances are compared with the state of the art.

The first experiment compares the performances of our baselines (Table 1). "DT" is the system based on Decision Tree using all the features of Figure 3 and "DT+Hill" is a DT using the features selected by the Hill Climbing process (Section 4.1.1). There is a significant improvement in the performances (5.1 points with CEAF, 5.3 with B^3) after the automatic feature selection process is done.

[2] $\frac{1}{6}n(n-1)(n-2)$ for each one of the three triangular constraints

	bnews	npaper	nwire	Global			
Metric:	CEAF			CEAF	B^3		
Model	F1	F1	F1	F1	P	R	F1
Relax	67.3	64.4	69.5	67.2	**88.4**	62.7	73.3
Relax pruning	68.6	65.2	70.1	68.0	82.3	**66.9**	73.8
Relax pruning & reorder	**69.5**	**67.3**	**72.1**	**69.7**	85.3	66.8	**74.9**
Relax random IS	68.2	66.1	71.0	68.5	83.5	66.7	74.2
MaxEnt+ILP (Denis, 2007)	-	-	-	66.2	81.4	65.6	72.7
Rankers (Denis, 2007)	65.7	65.3	68.1	67.0	79.8	66.8	72.7

Table 3: Results ACE-phase02.

In the second experiment the ILP chain formation process is applied using the output of both DTs. Results are shown in Table 2. Note that ILP only applies to documents shorter than 200 mentions due to its excessive computational cost (Section 4.1.2). Results for Relax applied to the same documents are also included for comparison. ILP forces consistency of the results producing an increase in precision score with B^3 metric in both cases. However, "DT+Hill" has been optimized for CEAF metric which encourages precision and consistency. For this, a post-process forcing consistency seems unnecessary for a classifier already optimized. Relax significantly outperforms all the baselines.

The third experiment shows the improvements achieved by the use of pruning and reordering techniques (Sections 3.2 and 3.4). Table 3 shows the results. Pruning improves performances with both metrics. B^3 precision is decreased but the global F_1 is increased due to a considerably improvement of recall. Reordering recovers the precision lost by the pruning without loosing recall, which achieves the best performances of 69.7 with CEAF and 74.9 with B^3.

The fourth experiment evaluates the influence of the initial state. A comparison is done with the proposed initial state (Section 3.3) and the random one. The results shown in Table 3 for random initial state are the average of 3 executions. The system called "Relax random IS" is using the same values for pruning and reordering techniques than the best result of previous experiment: "Relax pruning & reorder". As expected, results with a well-informed initial state outperform the random ones.

Finally, Relax performances are compared with the best scores we have found using the same corpora and metrics. We compare our approach with specialized Rankers –groupwise classifier–, and a system using ILP not only forcing consistency but also using information about anaphoricity and named entities. Relax outperforms both systems with both metrics (Table 3).

5 Conclusion

The approach for coreference resolution presented in this paper is a constraint-based graph partitioning solved by relaxation labeling.

The decision to join or not a set of mentions in the same entity is taken considering always the whole set of previous mentions like in groupwise classifiers. Contrarily to the approaches where variables are the linkage of each pair of mentions, in this model consistency is implicitly forced. Moreover, the influence of the partial results of the other mentions at the same time avoids that decisions are independently taken.

The capacity to easily incorporate constraints from different sources and using different knowledge is also remarkable. This flexibility gives a great potencial to the approach. Anaphoricity filtering is not needed given that the necessary knowledge can be also introduced by constraints.

In addition, three tecniques to improve results have been presented: reordering, pruning and feature selection by Hill Climbing. The experiments confirm their utility.

The experimental results clearly outperform the baselines with separate classification and chain formaiton. The approach also outperforms others in the state of the art using same corpora and metrics.

References

Aone, C. and S.W. Bennett. 1995. Evaluating automated and manual acquisition of anaphora resolution strategies. In *Proceedings of the 33rd annual meeting on ACL*, pages 122–129.

Atserias, J., B. Casas, E. Comelles, M. González, L. Padró, and M. Padró. 2006. Freeling 1.3: Syntactic and semantic services in an open-source nlp library. In *Proceedings of the fifth international conference on Language Resources and Evaluation (LREC 2006), ELRA*. Genoa, Italy.

Atserias, J. 2006. *Towards Robustness in Natural Language Understanding.* Ph.D. Thesis, Dept. Lenguajes y Sistemas Informáticos. Euskal Herriko Unibertsitatea. Donosti. Spain.

Bagga, A. and B. Baldwin. 1998. Algorithms for scoring coreference chains. *Proceedings of the Linguistic Coreference Workshop at LREC*, pages 563–566.

Culotta, A., M. Wick, and A. McCallum. 2007. First-Order Probabilistic Models for Coreference Resolution. *Proceedings of NAACL HLT*, pages 81–88.

Denis, P. and J. Baldridge. 2007. Joint Determination of Anaphoricity and Coreference Resolution using Integer Programming. *Proceedings of NAACL HLT*, pages 236–243.

Denis, P. and J. Baldridge. 2008. Specialized models and ranking for coreference resolution. *Proceedings of the EMNLP, Hawaii, USA.*

Denis, P. 2007. *New Learning Models for Robust Reference Resolution*. Ph.D. dissertation, University of Texas at Austin.

Finkel, J.R. and C.D. Manning. 2008. Enforcing transitivity in coreference resolution. In *Proceedings of the 46th Annual Meeting of the ACL HLT: Short Papers*, pages 45–48. Association for Computational Linguistics.

Gimenez, J. and L. Marquez. 2004. Svmtool: A general pos tagger generator based on support vector machines. In *Proceedings of the 4th International Conference on Language Resources and Evaluation*, pages 43–46.

Hoste, V. 2005. *Optimization Issues in Machine Learning of Coreference Resolution*. PhD thesis.

Hummel, R. A. and S. W. Zucker. 1987. On the foundations of relaxation labeling processes. pages 585–605.

Klenner, M. and É. Ailloud. 2008. Enhancing Coreference Clustering. In *Proceedings of the Second Workshop on Anaphora Resolution*. WAR II.

Klenner, M. and E. Ailloud. 2009. Optimization in Coreference Resolution Is Not Needed: A Nearly-Optimal Algorithm with Intensional Constraints. In *Proceedings of the 12th Conference of the EACL.*

Klenner, M. 2007. Enforcing consistency on coreference sets. In *Recent Advances in Natural Language Processing (RANLP)*, pages 323–328.

Luo, X., A. Ittycheriah, H. Jing, N. Kambhatla, and S. Roukos. 2004. A mention-synchronous coreference resolution algorithm based on the bell tree. In *Proceedings of 42nd ACL*, page 135.

Luo, X. 2005. On coreference resolution performance metrics. *Proc. of HLT-EMNLP*, pages 25–32.

Luo, X. 2007. Coreference or not: A twin model for coreference resolution. In *Proceedings of NAACL HLT*, pages 73–80.

Màrquez, L., L. Padró, and H. Rodríguez. 2000. A machine learning approach for pos tagging. *Machine Learning Journal*, 39(1):59–91.

McCallum, A. and B. Wellner. 2005. Conditional models of identity uncertainty with application to noun coreference. *Advances in Neural Information Processing Systems*, 17:905–912.

McCarthy, J.F. and W.G. Lehnert. 1995. Using decision trees for coreference resolution. *Proceedings of the Fourteenth International Conference on Artificial Intelligence*, pages 1050–1055.

Ng, V. and C. Cardie. 2002. Improving machine learning approaches to coreference resolution. *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 104–111.

Nicolae, C. and G. Nicolae. 2006. Best Cut: A Graph Algorithm for Coreference Resolution. *Proceedings of the 2006 Conference on EMNLP*, pages 275–283.

Pelillo, M. 1997. The dynamics of nonlinear relaxation labeling processes. *Journal of Mathematical Imaging and Vision*, 7(4):309–323.

Quinlan, J.R. 1993. *C4.5: Programs for Machine Learning*. Morgan Kaufmann.

Rosenfeld, R., R. A. Hummel, and S. W. Zucker. 1976. Scene labelling by relaxation operations. *IEEE Transactions on Systems, Man and Cybernetics*, 6(6):420–433.

Soon, W.M., H.T. Ng, and D.C.Y. Lim. 2001. A Machine Learning Approach to Coreference Resolution of Noun Phrases. *Computational Linguistics*, 27(4):521–544.

Surdeanu, M., J. Turmo, and E. Comelles. 2005. Named Entity Recognition from Spontaneous Open-Domain Speech. In *Ninth European Conference on Speech Communication and Technology*. ISCA.

Torras, C. 1989. Relaxation and neural learning: Points of convergence and divergence. *Journal of Parallel and Distributed Computing*, 6:217–244.

Vilain, M., J. Burger, J. Aberdeen, D. Connolly, and L. Hirschman. 1995. A model-theoretic coreference scoring scheme. *Proceedings of the 6th conference on Message understanding*, pages 45–52.

Yang, X., G. Zhou, J. Su, and C.L. Tan. 2003. Coreference resolution using competition learning approach. In *ACL '03: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pages 176–183.

"Expresses-an-opinion-about": using corpus statistics in an information extraction approach to opinion mining

Asad B. Sayeed, Hieu C. Nguyen, and Timothy J. Meyer
Department of Computer Science
University of Maryland, College Park
asayeed@cs.umd.edu,
hcnguyen88@gmail.com,
tmeyer1@umd.edu

Amy Weinberg
Institute for Advanced Computer Studies
Department of Linguistics
University of Maryland, College Park
weinberg@umiacs.umd.edu

Abstract

We present a technique for identifying the sources and targets of opinions without actually identifying the opinions themselves. We are able to use an information extraction approach that treats opinion mining as relation mining; we identify instances of a binary "expresses-an-opinion-about" relation. We find that we can classify source-target pairs as belonging to the relation at a performance level significantly higher than two relevant baselines.

This technique is particularly suited to emerging approaches in corpus-based social science which focus on aggregating interactions between sources to determine their effects on socio-economically significant targets. Our application is the analysis of information technology (IT) innovations. This is an example of a more general problem where opinion is expressed using either sub- or supersets of expressive words found in newswire. We present an annotation scheme and an SVM-based technique that uses the local context as well as the corpus-wide frequency of a source-target pair as data to determine membership in "expresses-an-opinion-about". While the presence of conventional subjectivity keywords appears significant in the success of this technique, we are able to find the most domain-relevant keywords without sacrificing recall.

1 Introduction

Two problems in sentiment analysis consist of source attribution and target discovery—who has an opinion, and about what? These problems are usually presented in terms of techniques that relate them to the actual opinion expressed. We have a social science application in which the identification of sources and targets over a large volume of text is more important than identifying the actual opinions particularly in experimenting with social science models of opinion trends. Consequently, we are able to use lightweight techniques to identify sources and targets without using resource-intensive techniques to identify opinionated phrases.

Our application for this work is the discovery of networks of influence among opinion leaders in the IT field. We are interested in answering questions about who the leaders in the field are and how their opinion matches the social and economic success of IT innovation. Consequently, it became necessary for us to construct a system (figure 1) that finds the expressions in text that refer to an opinion leader's activities in promoting or deprecating a technology.

In this paper, we demonstrate an information extraction (Mooney and Bunescu, 2005) approach based in relation mining (Girju et al., 2007) that is effective for this purpose. We describe a technique by which corpus statistics allow us to classify pairs of entities and sentiment analysis targets as instances of an "expresses-an-opinion-about" relation in documents in the IT business press. This genre has the characteristic that many entities and targets are represented within individual sentences and paragraphs. Features based on the

Coling 2008: Poster Volume, pages 1095–1103,
Beijing, August 2010

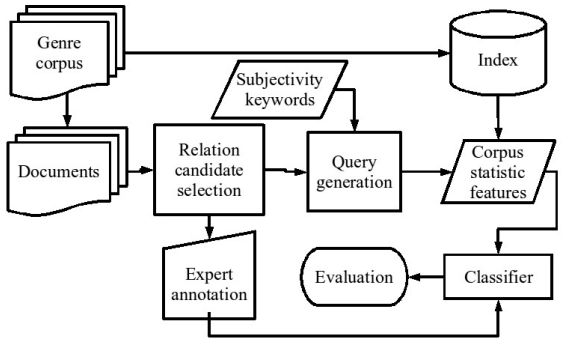


Figure 1: Opinion relation classification system.

frequency counts of query results allow us to train classifiers that allow us to extract "expresses-an-opinion-about" instances, using a very simple annotation strategy to acquire training examples.

In the IT business press, the opinionated language is different from the newswire text for which many extant sentiment tools were developed. We use an existing sentiment lexicon alongside other non-sentiment-specific measures that adapt resources from newswire-developed sentiment analysis projects without imposing the full complexity of those techniques.

1.1 Corpus-based social science

The "expresses-an-opinion-about" relation is a binary relation between opinion sources and targets. Sources include both people—typically known experts, corporate representatives, and other businesspeople—as well as organizations such as corporations and government bodies. The targets are the innovation terms. Therefore, the use of named-entity recognition in this project only focuses on persons and organizations, as the targets are a fixed list.

1.2 Reifying opinion in an application context

A hypothesis implicit in our social science task is that opinion leaders create trends in IT innovation adoption partly by the text that their activities generate in the IT business press. This text has an effect on readers, and these readers act in such a way that in turn may generate more or less prominence for a given innovation—and may also generate further text.

Some of these text-generating activities include expressions of private states in an opinion source (e.g., "I believe that *Web 2.0 is the future*"). These kinds of expressions suggest a particular ontology of opinion analysis involving discourse relations across various types of clauses (Wilson and Wiebe, 2005; Wilson et al., 2005a). However, if we are to track the relative adoption of IT innovations, we must take into account the effect of the text on the reader's opinion about these innovations—there are expressions other than those of private states that have an effect on the reader. These can be considered to be "opinionated acts[1]."

Opinionated acts can include things like purchasing and adoption decisions by organizations. For example:

> And like other top suppliers to **Wal-Mart Stores Inc.**, **BP** has been involved in a mandate to affix *radio frequency identification* tags with embedded electronic product codes to its crates and pallets. (ComputerWorld, January 2005)

In this case, both Wal-Mart and BP have expressed implicit approval for radio frequency identification by adopting it. This may affect the reader's own likelihood of support or adoption of the technology. In this context, we do not directly consider the subjectivity of the opinion source, even though that may be present.

Opinionated acts include things like implications of technology use, not just adoption. We thus define opinion expressions as follows: any expression involving some actor that is likely to affect a reader's own potential to adopt, reject, or speak positively or negatively of a target. This would include "conventional" expressions of private states as well as opinionated acts.

Our definition of "expresses-an-opinion-about" follows immediately. Source A expresses an opinion about target B if an interested third party C's actions towards B may be affected by A's textually recorded actions, in a context where actions

[1] Somasundaran and Wiebe (2009) mention a related category of "pragmatic opinions" that involve world knowledge.

have positive or negative weight (e.g. purchasing, promotion, etc.).

1.3 Domain-specific sentiment detection

We construct a system that uses named-entity recognition and supervised machine learning via SVMs to automatically discover instances of "expresses-an-opinion-about" as a binary relation at reasonably high accuracy and precision.

The advantage of our approach is that, outside of HMM-based named-entity detection (BBN's IdentiFinder), we evade the need for resource-intensive techniques such as sophsticated grammatical models, sequence models, and semantic role labelling (Choi et al., 2006; Kim and Hovy, 2006) by removing the focus on the actual opinion expressed. Then we can use a simple supervised discriminative technique with a joint model of local term frequency information and corpus-wide co-occurrence distributions in order to discover the raw data for opinion trend modelling. The most complex instrument we use from sentiment analysis research on conventional newswire is a sentiment keyword lexicon (Wilson et al., 2005b); furthermore, our techniques allow us to distinguish sentiment keywords that indicate opinion in this domain from keywords that actually indicate that there is no opinion relation between source and target.

While we show that this lightweight technique works well at a paragraph level, it can also be used in conjunction with more resource-intensive techniques used to find "conventional" opinion expressions. Also, the use of topic aspects (Somasundaran and Wiebe, 2009) in conjunction with target names has been associated with an improvement in recall. However, our technique still performs well above the baseline without these improvements.

2 Methodology

2.1 Article preparation

We have a list of IT innovations on which our opinion leader research effort is most closely focused. This list contains common names that refer to these technologies as well as some alternate names and abbreviations. We selected articles at random from the ComputerWorld IT journal that contained mentions of members of the given list. These direct mentions were tagged in the document as XML entities.

Each article was processed by BBN's IdentiFinder 3.3 (Bikel et al., 1999), a named entity recognition (NER) system that tags named mentions of person and organization entities[2].

The articles were then divided into paragraphs. For each paragraph, we generated candidate relations from the entities and innovations mentioned therein. To generate candidates, we paired every entity in the paragraph with every innovation. Redundant pairs are sometimes generated when an entity is mentioned in multiple ways in the paragraph. We eliminated most of these by removing entities whose mentions were substrings of other mentions. For example, "Microsoft" and "Microsoft Corp." are sometimes found in the same paragraph; we eliminate "Microsoft."

2.2 Annotation

We processed 20 documents containing 157 relations in the manner described in the previous section. Then two domain experts (chosen from the authors) annotated every candidate pair in every document according to the following scheme (illustrated in figure 2):

- If the paragraph associated with the candidate pair describes a valid source-target relation, the experts annotated it with `Y`.
- If the paragraph does not actually contain that source-target relation, the experts annotated it with `N`.
- If either the source or the target is misidentified (e.g., errors in named entity recognition), the experts annotated it with `X`.

The Cohen's κ score was 0.6 for two annotators. While this appears to be only moderate agreement, we are still able to achieve good performance in our experiments with this value.

[2] In a separate research effort, we found that IdentiFinder has a high error rate on IT business press documents, so we built a system to reduce the error *post hoc*. We ran this system over the IdentiFinder annotations.

Davis says she has especially enjoyed working with the **PowerPad**'s *bluetooth* interfaces to phones and printers. "It's nice getting into new wireless technology," she says. The *bluetooth* capability will allow couriers to transmit data without docking their devices in their trucks.

Source	Target	Class
Davis	bluetooth	**Y**/N/X
PowerPad	bluetooth	Y/N/**X**

Figure 2: Example paragraph annotation exercise.

We then selected 75 different documents for each annotator and processed and annotated them as above. At this point we have the instances and the classes to which they belong. We labelled 466 instances of Y, 325 instances of N, and 280 instances of X, for a total of 1071 relations.

2.3 Feature vector generation

We have four classes of features for every relation instance. Each type of feature consists of counts extracted from an index of 77,227 ComputerWorld articles from January 1988 to June 2008 generated by the University of Massachusetts search engine Indri (Metzler and Croft, 2004). Each vector is normalized to the unit vector. The index is not stemmed for performance reasons.

The first type of feature consists of simple document frequency statistics for source-target pairs throughout the corpus. The second type consists of document frequency counts of source-target pairs when they are in particularly close proximity to one another. The third type consists of document frequency counts of source target pairs proximate to keywords that reflect subjectivity. The fourth and final type consist of TFIDF scores of vocabulary items in the paragraph containing the putative opinion-holding relation (unigram context features). We use the first three features types to represent the likelihood in the "world" that the source has an opinion about the target and the last feature type to represent the likelihood of the specific paragraph containing an opinion that reflects the source-target relation.

We have a total of 7450 features. Each vector is represented as a sparse array. 806 features represent queries on the Indri index. For all the features, we therefore have 863,226 index queries. We perform the queries in parallel on 25 processors to generate the full feature array, which takes approximately an hour on processors running at 8Ghz. We eliminate all values that are smaller in magnitude than 0.000001 after unit vector normalization.

2.3.1 Frequency statistics

There are two simple frequency statistics features generated from Indri queries. The first is the raw frequency counts of within-document co-occurrences of the source and target in the relation. The second is the mean co-occurrence frequency of the source and target per ComputerWorld document.

2.3.2 Proximity counts

For every relation, we query Indri to check how often the source and the target appear in the same document in the ComputerWorld corpus within four word ranges: 5, 25, 100, and 500. That is to say, if a source and a target appear within five words of one another, this is included in the five-word proximity feature. This generates four features per relation.

2.3.3 Subjectivity keyword proximity counts

We augment the proximity counts feature with a third requirement: that the source and target appear within one of the ranges with a "subjectivity keyword." The keywords are taken from University of Pittsburgh subjectivity lexicon; the utility of this lexicon is supported in recent work (Somasundaran and Wiebe, 2009).

For performance reasons, we did not use all of the entries in the subjectivity lexicon. Instead, we used a TFIDF-based measure to rank the keywords by their prevalence in the ComputerWorld corpus where the term frequency is defined over the entire corpus. Then we selected 200 keywords with the highest score.

For each keyword, we use the same proximity ranges (5, 25, 100, and 500) in queries to Indri where we obtain counts of each keyword-source-target triple for each range. There are threfore 800 subjectivity keyword features.

Positive class	Negative class	System	Prec / Rec / F	Accuracy
Y	N	Random baseline	0.60 / 0.53 / 0.56	0.52
Y	N	Maj.-class (Y) baseline	0.59 / *1.00* / 0.74	0.59
Y	N	Linear kernel	0.70 / 0.73 / 0.72	0.66
Y	N	RBF kernel	**0.72 / 0.76 / 0.75**	**0.69**
Y	N/X	Random baseline	0.44 / 0.50 / 0.47	0.50
Y	N/X	RBF kernel	0.65 / 0.55 / 0.59	0.67

Table 1: Results with all features against majority class and random baselines. All values are mean averages under 10-fold cross validation.

2.3.4 Word context (unigram) features

For each relation, we take term frequency counts of the paragraph to which the relation belongs. We multiply them by the IDF of the term across the ComputerWorld corpus. This yields 6644 features over all paragraphs.

2.4 Machine learning

On these feature vectors, we trained SVM models using Joachims' (1999) svmlight tool. We use a radial basis function kernel with an error cost parameter of 100 and a γ of 0.25. We also use a linear kernel with an error cost parameter of 100 because it is straightforwardly possible with a linear kernel to extract the top features from the model generated by svmlight.

3 Experiments

We conducted most of our experiments with only the Y and N classes, discarding all X; this restricted most of our results to those assuming correct named entity recognition. Y was the positive class for training the svmlight models, and N was the negative class. We also performed experiments with N and X together being the negative class; this represents the condition that we are seeking "expresses-an-opinion-about" even with a higher named-entity error rate.

We use two baselines. One is a random baseline with uniform probability for the positive and negative classes. The other is a majority-class assigner (Y is the majority class).

The best system for the Y vs. N experiment was subjected to feature ablation. We first systematically removed each of the four feature types individually. The feature type whose removal had the largest effect on performance was removed permanently, and the rest of the features were tested without it. This was done once more, at which point only one feature type was present in the models tested.

3.1 Evaluation

All evaluation was performed under 10-fold cross validation, and we report the mean average of all performance metrics (precision, recall, harmonic mean F-measure, and accuracy) across folds.

We define these measures in the standard information retrieval form. If tp represents true positives, tn true negatives, fp false positives, and fn false negatives, then precision is $tp/(tp+fp)$, recall $tp/(tp+fn)$, F-measure (harmonic mean) is $2(prec*rec)/(prec+rec)$, and accuracy is $(tp+tn)/(tp+fp+fn+tn)$.

4 Results and discussion

The results of the experiments with all features are listed in table 1.

4.1 "Perfect" named entity recognition

We achieve best results in the Y versus N case using the radial basis function kernel. We find improvement in F-measure and accuracy at 19% and 17% respectively. Simply assigning the majority class to all test examples yields a very high recall, by definition, but poor precision and accuracy; hence its relatively high F-measure does not reflect high applicability to further processing, as the false positives would amplify errors in our social science application.

The linear kernel has results that are below the RBF kernel for all measures, but are relatively close to the RBF results.

Subjectivity	Proximity	Frequency	Unigram	Prec / Rec / F	Accuracy
✓	✓	✓	✓	0.72 / 0.76 / 0.75	0.69
	✓	✓	✓	0.67 / 0.89 / 0.76	0.67
✓		✓	✓	0.71 / 0.77 / 0.73	0.68
✓	✓		✓	0.70 / 0.78 / 0.74	0.67
✓	✓	✓		0.69 / 0.77 / 0.73	0.67
		✓	✓	0.63 / 0.91 / 0.75	0.64
	✓		✓	0.66 / 0.89 / 0.76	0.67
	✓	✓		0.65 / 0.90 / 0.76	0.66
		✓		0.61 / 0.92 / 0.73	0.60
			✓	0.61 / 0.94 / 0.74	0.60

Table 2: Feature ablation results for RBF kernel on `Y` vs. `N` case. The first line is the RBF result with all features from table 1.

4.2 Introducing erroneous named entities

The case of `Y` versus `N` and `X` together unsurprisingly performed worse than the case where named entity errors were eliminated. However, relative to its own random baseline, it performed well, with a 12% and 17% improvement in F-measure and accuracy using the RBF kernel. This suggests that the errors do not introduce enough noise into the system to produce a large decline in performance.

As `X` instances are about 26% of the total and we see a considerable drop in recall, we can say that some of the `X` instances are likely to be similar to valid `Y` ones; indeed, examination of the named entity recognizer's errors suggests that some incorrect organizations (e.g. product names) occur in contexts where valid organizations occur. However, precision and accuracy have not fallen nearly as far, so that the quality of the output for further processing is not hurt in proportion to the introduction of `X` class noise.

4.3 Feature ablation

Table 2 contains the result of our feature ablation experiments. Overall, the removal of features causes the SVM models to behave increasingly like a majority class assigner. As we mentioned earlier, higher recall at the expense of precision and accuracy is not an optimal outcome for us even if the F-measure is preserved. In our results, the F-measure values are remarkably stable.

In the first round of feature removal, the subjectivity keyword features have the biggest effect with the largest drop in precision and the largest increase in recall; high-TFIDF words from a general-purpose subjectivity lexicon allow the model to assign more items to the negative class.

The next round of feature removal shows that the proximity features have the next largest amount of influence on the classifier, as precision drops by 4%. The proximity features are very similar to the subjectivity features in that they too involve queries over windows of limited word sizes; the subjectivity keyword features only differ in that a subjectivity keyword must be within the window as well. That the proximity features are not more important than the subjectivity features, implies that the subjectivity keywords matter to the classifier, even though they are not specific to the IT domain. However, the proximity of sources and targets also matters, even in the absence of the subjectivity keywords.

Finally, we are left with the frequency features and the unigram context features. Either set of features supports a level of performance greater than the random baseline in table 1. However, the unigram features allow for slightly better recall than the frequency features without loss of precision, but this may not be very surprising, as there are *many* more unigram features than frequency features. More importantly, however, either of these feature types is sufficient to prevent the classifier from assigning the majority class all of the time, although they come close.

Feature type	Range	Keyword
Subjectivity	500	agreement
Subjectivity	500	critical
Subjectivity	500	want
Subjectivity	100	will
Subjectivity	100	able
Subjectivity	500	worth
Subjectivity	500	benefit
Subjectivity	100	trying
Subjectivity	500	large
Subjectivity	500	competitive

Table 3: The 10 most positive features via a linear kernel in descending order.

Feature type	Range	Keyword
Subjectivity	500	low
Subjectivity	500	ensure
Subjectivity	25	want
Subjectivity	100	vice
Subjectivity	500	slow
Subjectivity	100	large
Subjectivity	500	ready
Subjectivity	100	actually
Subjectivity	100	ready
Subjectivity	100	against

Table 4: The 10 most negative features via a linear kernel in descending order.

4.4 Most discriminative features

The models generated by svmlight under a linear kernel allow for the extraction of feature weights by a script written by svmlight's creator. We divided the instances into a single 70%/30% train/test split and trained a classifier with a linear kernel and an error cost parameter of 100, with results similar to those reported under 10-fold cross-validation in table 1. We used all features.

Then we were able to extract the 10 most positive (table 3) and 10 most negative (table 4) features from the model.

Interestingly, all of these are subjectivity keyword features, even the negatively weighted features. The top positive features are often evocative of business language, such as "agreement", "critical", and "competitive". Most of them emerge from queries at the 500-word range, suggesting that their presence in the document itself is evidence that a source is expressing an opinion about a target. That most of them are subjectivity features is reflected in the feature ablation results in the previous section.

It is less clear why "ensure" and "against" should be evidence that a source-target pair is *not* an instance of "expresses-an-opinion-about". On the other hand, words like "ready" (which appears twice) and "actually" can conceivably reflect situations in the IT domain that are not matters of opinion. In either case, this demonstrates one of the advantages of our technique, as these are features that actively assist in classifying some relation instances as not expressing sentiment. For example, contrary to what we would expect, "want" in a 25-word window with a source and a target is actually evidence against an "expresses-an-opinion-about" relation in text about IT innovations (ComputerWorld, July 2007):

> But **Klein**, who is director of information services and technology, didn't want IT to become the *blog* police.

In this example, Klein is expressing a desire, but not about the innovation (blogs) in question.

5 Conclusions and future work

5.1 Summary

We constructed and evaluated a system that detects at paragraph level whether entities relevant to the IT domain have expressed an opinion about a list of IT innovations of interest to a larger social science research program. To that end, we used a combination of co-occurrence statistics gleaned from a document indexing tool and TFIDF values from the local term context. Under these novel conditions, we successfully exceeded simple baselines by large margins.

Despite only moderate annotator agreement, we were able to produce results coherent enough to successfully train classifiers and conduct experiments.

Our feature ablation study suggests that all of the feature types played a role in improving the performance of the system over the random and

majority-class baselines. However, the subjectivity keyword features from an existing lexicon played the largest role, followed by the proximity and unigram features. Subjectivity keyword features dominated the ranks of feature weights under a linear kernel, and the features most predictive of membership in "expresses-an-opinion-about" are words with semantic significance in the context of the IT business press.

5.2 Application to other domains

We used somewhat naïve statistics in a simple machine learning system in order to implement a form of opinion mining for a particular domain. The most direct linguistic guidance we provided our system were the query ranges and the subjectivity lexicon. The generality of this approach yields the advantage that it can be applied to other domains where there are ways of expressing sentiment unique to those domains outside of newswire text and product reviews.

5.3 Improving the features

Our use of an existing sentiment lexicon opens the door in future work for the use of techniques to bootstrap a larger sentiment lexicon that emphasizes domain-specific language in the expression of opinion, including opinionated acts. In fact, our results suggest that terminology in the existing lexicon that is most prominently weighted in our classifier also tends to be domain-relevant. In a further iteration, we might also improve performance by using terms outside the lexicon that tend to co-occur with terms from the lexicon.

5.4 Data generation

Our annotation exercise was a very simple one involving a short reading exercise and the selection of one of three choices per relation instance. This type of exercise is ideally suited to the "crowdsourcing" technique of paying many individuals small amounts of money to perform these simple annotations over the Internet. Previous research (Snow et al., 2008) suggests that we can generate very large datasets very quickly in this way; this is a requirement for expanding to other domains.

5.5 Scalability

In order to classify on the order of 1000 instances, it took nearly a million queries to the Indri index, which took a little over an hour to do in parallel on 25 processors by calling the Indri query engine afresh at each query. While each query is necessary to generate each feature value, there are a number of optimizations we could implement to accelerate the process. Various types of dynamic programming and caching could be used to handle related queries. One way of scaling up to larger datasets would be to use the MapReduce and cloud computing paradigms on which text processing tools have already been implemented (Moreira et al., 2007).

The application for this research is a social science exercise in exploring trends in IT adoption by analysing the IT business press. In the end, the perfect discovery of all instances of "expresses-an-opinion-about" is not as important as finding enough reliable data over a large number of documents. This work brings us several steps closer in finding the right combination of features in order to acquire trend-representative data.

Acknowledgements

This paper is based upon work supported by the National Science Foundation under Grant IIS-0729459.

References

Bikel, Daniel M., Richard Schwartz, and Ralph M. Weischedel. 1999. An algorithm that learns what's in a name. *Mach. Learn.*, 34(1-3).

Choi, Yejin, Eric Breck, and Claire Cardie. 2006. Joint extraction of entities and relations for opinion recognition. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Girju, Roxana, Preslav Nakov, Vivi Nastase, Stan Szpakowicz, Peter Turney, and Deniz Yuret. 2007. Semeval-2007 task 04: classification of semantic relations between nominals. In *SemEval '07: Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 13–18, Morristown, NJ, USA. Association for Computational Linguistics.

Joachims, T. 1999. Making large-scale SVM learning practical. In Schölkopf, B., C. Burges, and

A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*, chapter 11, pages 169–184. MIT Press, Cambridge, MA.

Kim, Soo-Min and Eduard Hovy. 2006. Extracting opinions, opinion holders, and topics expressed in online news media text. In *SST '06: Proceedings of the Workshop on Sentiment and Subjectivity in Text*, pages 1–8, Morristown, NJ, USA. Association for Computational Linguistics.

Metzler, Donald and W. Bruce Croft. 2004. Combining the language model and inference network approaches to retrieval. *Information Processing and Management*, 40(5):735 – 750.

Mooney, Raymond J. and Razvan Bunescu. 2005. Mining knowledge from text using information extraction. *SIGKDD Explor. Newsl.*, 7(1):3–10.

Moreira, José E., Maged M. Michael, Dilma Da Silva, Doron Shiloach, Parijat Dube, and Li Zhang. 2007. Scalability of the nutch search engine. In Smith, Burton J., editor, *ICS*, pages 3–12. ACM.

Rogers, Everett M. 2003. *Diffusion of Innovations, 5th Edition*. Free Press.

Snow, Rion, Brendan O'Connor, Daniel Jurafsky, and Andrew Y. Ng. 2008. Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks. In *EMNLP 2008*, Morristown, NJ, USA.

Somasundaran, Swapna and Janyce Wiebe. 2009. Recognizing stances in online debates. In *ACL-IJCNLP '09: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1*. Association for Computational Linguistics.

Wilson, Theresa and Janyce Wiebe. 2005. Annotating attributions and private states. In *ACL 2005 Workshop: Frontiers in Corpus Annotation II: Pie in the Sky*, pages 53–60.

Wilson, Theresa, Paul Hoffmann, Swapna Somasundaran, Jason Kessler, Janyce Wiebe, Yejin Choi, Claire Cardie, Ellen Riloff, and Siddharth Patwardhan. 2005a. OpinionFinder: A system for subjectivity analysis. In *HLT/EMNLP*.

Wilson, Theresa, Janyce Wiebe, and Paul Hoffmann. 2005b. Recognizing contextual polarity in phrase-level sentiment analysis. In *HLT/EMNLP*.

Sentiment Translation through Multi-Edge Graphs

Christian Scheible, Florian Laws, Lukas Michelbacher, and **Hinrich Schütze**
Institute for Natural Language Processing
University of Stuttgart
{scheibcn, lawsfn, michells}@ims.uni-stuttgart.de

Abstract

Sentiment analysis systems can benefit from the *translation* of sentiment information. We present a novel, graph-based approach using SimRank, a well-established graph-theoretic algorithm, to transfer sentiment information from a source language to a target language. We evaluate this method in comparison with semantic orientation using pointwise mutual information (SO-PMI), an established unsupervised method for learning the sentiment of phrases.

1 Introduction

Sentiment analysis is an important topic in computational linguistics that is of theoretical interest but is also useful in many practical applications. Usually, two aspects are of importance in sentiment analysis. The first is the detection of subjectivity, i.e., whether a text or an expression is meant to express sentiment at all; the second is the determination of sentiment orientation, i.e., what sentiment is to be expressed in a structure that is considered subjective.

Work on sentiment analysis most often covers resources or analysis methods in a single language, usually English. However, the transfer of sentiment analysis between languages can be advantageous by making use of resources for a source language to improve the analysis of the target language.

This paper presents an approach to the transfer of sentiment information between two languages that does not rely on resources with limited availability like parallel corpora. It is built around SimRank, a graph similarity algorithm that has successfully been applied to the acquisition of bilingual lexicons (Laws et al., 2010) and semantic similarity (Michelbacher et al., 2010). It uses linguistic relations extracted from two monolingual corpora to determine the similarity of words in different languages. One of the main benefits of our method is its ability to handle sparse data about the relations between the languages well (i.e., a small seed lexicon). Further, we experiment with combining multiple types of linguistic relations for graph-based translation. Our experiments are carried out using English as a source language and German as a target language. We evaluate our method using a hand-annotated set of German adjectives which we intend to publish.

In the following section, related work is discussed. Section 3.1 gives an introduction to SimRank and its application to lexicon induction, while section 3.2 reviews SO-PMI (Turney, 2002), an unsupervised baseline method for the generation of sentiment lexicons. In section 4, we define our sentiment transfer method which we apply in experiments in section 5.

2 Related Work

Mihalcea et al. (2007) propose two methods for translating sentiment lexicons. The first method simply uses bilingual dictionaries to translate an English sentiment lexicon. A sentence-based classifier built with this list achieved high precision, but low recall on a small Romanian test set. The second method is based on parallel corpora. The source language in the corpus is annotated with sentiment information, and the information is then projected to the target language. Problems arise due to mistranslations.

Banea et al. (2008) use machine translation for multilingual sentiment analysis. Given a corpus annotated with sentiment information in one language, machine translation is used to produce an annotated corpus in the target language, by preserving the annotations. The original annotations

Coling 2008: Poster Volume, pages 1104–1112,
Beijing, August 2010

can be produced either manually or automatically.

Wan (2009) constructs a multilingual classifier using co-training. In co-training, one classifier produces additional training data for a second classifier. In this case, an English classifier assists in training a Chinese classifier.

The induction of a sentiment lexicon is the subject of early work by Hatzivassiloglou and McKeown (1997). They construct graphs from coordination data from large corpora based on the intuition that adjectives with the same sentiment orientation are likely to be coordinated. For example, *fresh and delicious* is more likely than *rotten and delicious*. They then apply a graph clustering algorithm to find groups of adjectives with the same orientation. Finally, they assign the same label to all adjectives that belong to the same cluster.

Corpus work and bilingual dictionaries are promising resources for translating sentiment. In contrast to previous approaches, the work presented in this paper uses corpora that are not annotated with sentiment.

Turney (2002) suggests a corpus-based extraction method based on his pointwise mutual information (PMI) synonymy measure. He assumes that the sentiment orientation of a phrase can be determined by comparing its pointwise mutual information with a positive (*excellent*) and a negative phrase (*poor*). An introduction to this method is given in Section 3.2.

3 Background

3.1 Lexicon Induction via SimRank

We use the extension of the SimRank (Jeh and Widom, 2002) node similarity algorithm proposed by Dorow et al. (2009). Given two graphs $\mathcal{A}$ and $\mathcal{B}$, the similarity between two nodes a in $\mathcal{A}$ and b in $\mathcal{B}$ is computed in each iteration as:

$$S(a,b) = \frac{c}{|N_A(a)||N_B(b)|} \sum_{k \in N_A(a), l \in N_B(b)} S(k,l).$$

$N_X(x)$ is the neighborhood of node x in graph X. To compute similarities between two graphs, some initial links between these graphs have to be given, called seed links. These form the recursion basis which sets $S(a,b) = 1$ if there is a seed link between a and b. At the beginning of each iteration, all known equivalences between nodes are reset to 1.

Multi-Edge Extraction (MEE). MEE is an extension of SimRank that, in each iteration, computes the average node-node similarity of several different SimRank matrices. In our case, we use two different SimRank matrices, one for coordinations and one for adjective modification. See (Dorow et al., 2009) for details. We also used the node degree normalization function $h(n) = \sqrt{n} \times \sqrt{\max_k(|N(k)|)}$ (where n is the node degree, and $N(k)$ the degree of node k) to decrease the harmful effect of high-degree nodes on final similarity values. See (Laws et al., 2010) for details.

3.2 SO-PMI

Semantic orientation using pointwise mutual information (SO-PMI) (Turney, 2002) is an algorithm for the unsupervised learning of semantic orientation of words or phrases. A word has positive (resp. negative) orientation if it is associated with positive (resp. negative) terms more frequently than with negative (resp. positive) terms. Association of terms is measured using their pointwise mutual information (PMI) which is defined for two words w_1 and w_2 as follows:

$$\text{PMI}(w_1, w_2) = log\left(\frac{p(w_1, w_2)}{p(w_1)p(w_2)}\right)$$

Using PMI, Turney defines SO-PMI for a word w as

$$\text{SO-PMI}(w) =$$

$$log\frac{\prod_{p \in P} hits(word \text{ NEAR } p) \times \prod_{n \in N} hits(n)}{\prod_{n \in N} hits(word \text{ NEAR } n) \times \prod_{p \in P} hits(p)}$$

$hits$ is a function that returns the number of hits in a search engine given the query. P is a set of known positive words, N a set of known negative words, and `NEAR` an operator of a search engine that returns documents in which the operands occur within a close range of each other.

4 Sentiment Translation

Unsupervised methods like SO-PMI are suitable to acquire basic sentiment information in a language. However, since hand-annotated resources for sentiment analysis exist in other languages, it seems plausible to use automatic translation of sentiment information to leverage these resources. In order to translate sentiment, we will use multiple sources of information that we represent in a MEE graph as given in Section 3.1.

In our first experiments (Scheible, 2010), coordinated adjectives were used as the sole training source. Two adjectives are coordinated if they are linked with a conjunction like *and* or *but*. The intuition behind using coordinations – based on work by Hatzivassiloglou and McKeown (1997) and Widdows and Dorow (2002) – was that words which are coordinated share properties. In particular, coordinated adjectives usually express similar sentiments even though there are exceptions (e.g., "The movie was both good and bad").

In this paper, we focus on using multiple edge types for sentiment translation. In particular, the graph we will use contains two types of relations, *coordinations* and *adjective-noun modification*. In the sentence "The movie was enjoyable and fun", *enjoyable* and *fun* are coordinated. In *This is an enjoyable movie*, the adjective *enjoyable* modifies the noun *movie*.

We selected these two relation types for two reasons. First, the two types provide clues for sentiment analysis. Coordination information is an established source for sentiment similarity (e.g. Hatzivassiloglou and McKeown (1997)) while adjective-noun relations provide a different type of information on sentiment. For example, nouns with positive associations (*vacation*) tend to occur with positive adjectives and nouns with negative associations (*pain*) tend to occur with negative adjectives. Second, we have successfully used these two types for a similar acquisition task, the acquisition of word-to-word translation pairs (Laws et al., 2010).

In the resulting graph, adjectives and nouns are represented as nodes, each containing a word and its part of speech, and relations are represented as links which are distinguished by their edge types. Two graphs, one in the source language and one in the target language, are needed to translate words between those languages. Figure 1 shows an example for such a setup. Black links in this graph are coordinations, grey links are seed relations.

In order to calculate sentiment for all nodes in the target language, we apply the SimRank algorithm to the graphs which gives us similarities between all nodes in the source graph and all nodes in the target graph. Using the similarity $S(n_s, n_t)$ between a node n_s in the source language graph $\mathcal{S}$ and a node n_t in the target language graph $\mathcal{T}$, the sentiment score ($\text{sent}(n_t)$) is the similarity-weighted average of all sentiment scores in the target language:

$$\text{sent}(n_t) = \sum_{n_s \in \mathcal{S}} \text{sim}_{\text{norm}}(n_s, n_t)\, \text{sent}(n_s)$$

We assume that sentiment scores in the source language are expressed on a numeric scale. The normalized similarity sim_{norm} is defined as

$$\text{sim}_{\text{norm}}(n_s, n_t) = \frac{S(n_s, n_t)}{\sum_{n_s \in \mathcal{S}} S(n_s, n_t)}.$$

The normalization assures that all resulting sentiment values are within $[-1, 1]$, with -1 being the most negative sentiment and 1 the most positive.

5 Experiments

5.1 Data Acquisition

For our experiments, we needed coordination data to build weighted graphs and a bilingual lexicon to define seed relations between those graphs. Coordinations were extracted from the English and German versions of Wikipedia[1] by applying pattern-based search using the Corpus Query Processor (CQP) (Christ et al., 1999). We annotated both corpora with parts of speech using the Tree Tagger (Schmid, 1994). A total of 477,291 English coordinations and 112,738 German coordinations were collected. A sample of this data is given in Figure 2. We restrict these experiments to the use of *and*/*und* since other coordinations

[1] `http://www.wikipedia.org/` (01/19/2009)

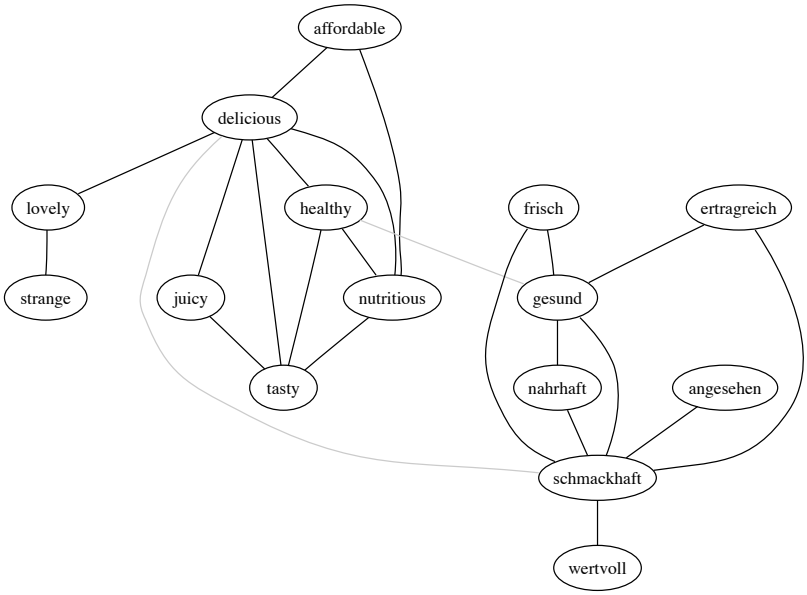


Figure 1: A German and an English graph with coordinated adjectives including seed links

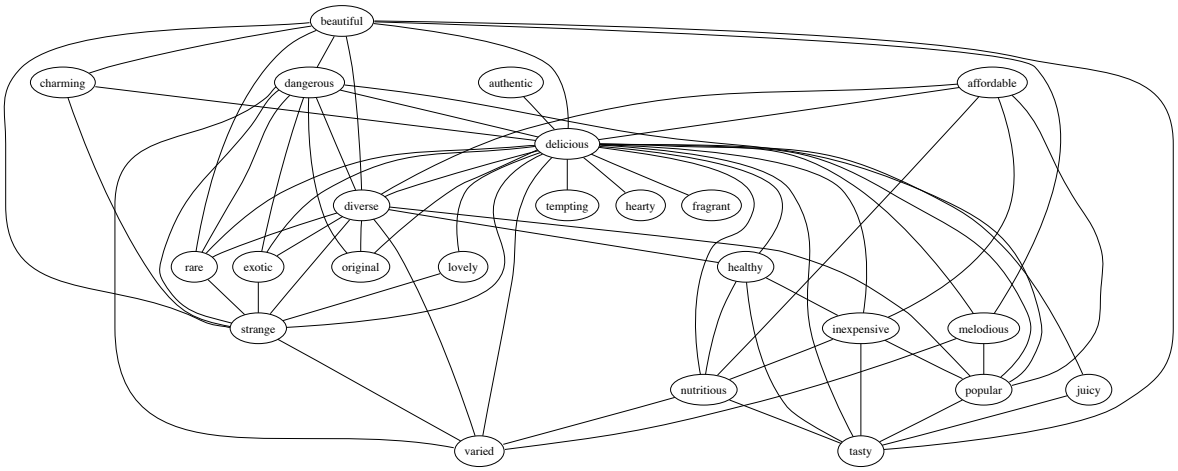


Figure 2: English sample coordinations (adjectives)

behave differently and might even express dissimilarity (e.g. *Was the weather good or bad?*).

The seed lexicon was constructed from the dict.cc dictionary[2]. While the complete dictionary contains 30,551 adjective pairs, we reduced the number of pairs used in the experiments to 1,576.

To produce a smaller seed lexicon which still makes sense from a semantic point of view, we used the General Service List (GSL) (West, 1953) which contains about 2000 words the author considered central to the English language. More specifically, a revised list was used[3].

SO-PMI needs a larger amount of training data. Since Wikipedia does not satisfy this need, we collected additional coordination data from the web using search result counts from Google. In Turney's original paper, he uses the `NEAR` operator, which returns documents that contain two search terms that are within a certain distance of each other, to collect collocations. Unfortunately, Google does not support this operator, so instead, we searched for coordinations using the queries

`+ "w and s"` and
`+ "w und s"`

for English and German, respectively. We added the quotes and the + operator to make sure that both spelling correction and synonym replacements were disabled.

The original experiments were made for English, so we had to construct our own set of seed words. For German, we chose *gut (good)*, *nett (nice)*, *richtig (right)*, *schön (beautiful)*, *ordentlich (neat)*, *angenehm (pleasant)*, *aufrichtig (honest)*, *gewissenhaft (faithful)*, and *hervorragend (excellent)* as positive seed words, and *schlecht (bad)*, *teuer (expensive)*, *falsch (wrong)*, *böse (evil)*, *feindlich (hostile)*, *verhasst (invidious)*, *widerlich (disgusting)*, *fehlerhaft (faulty)*, and *mangelhaft (flawed)* as negative ones.

5.2 Sentiment Lexicon

For our experiments, we used two different polarity lexicons. The lexicon of Wilson et al. (2005) contains sentiment annotations for 8,221 words which are tagged as *positive*, *neutral*, or *negative*. A few words are tagged as *weakneg*, implying weak negativity. These categorial annotations are mapped to the range [-1,1] using the assignment scheme given in Table 1.

annotation	value
positive	1.0
weakpos	0.5
neutral	0.0
weakneg	−0.5
negative	−1.0

Table 1: Assigned values for Wilson et al. set

5.3 Human Ratings

In order to manually annotate a test set, we chose 200 German adjectives that occurred in the Wikipedia corpus and that were part of a coordination. From these words, we removed those which we deemed uncommon, too complicated, or which were mislabeled as adjectives by the tagger. The test set contained 150 adjectives of which seven were excluded after annotators discarded them.

We asked 9 native speakers of German to annotate the adjectives. Possible annotations were *very positive*, *slightly positive*, *neutral*, *slightly negative*, or *very negative*. These categories are the same as the ones used in the training data.

In order to capture the general sentiment, i.e., sentiment that is not related to a specific context, the judges were asked to stay objective and not let their personal opinions influence the annotation. However, some words with strong political implications were annotated by some judges as non-neutral which led to disagreement beyond the usual level. *Nuklear* (*nuclear*) is an example for such a word. We measured the agreement of the judges with Kendall's coefficient of concordance (W) with tie correction (Legendre, 2005), yielding $W = 0.674$ with a high level of significance ($p < .001$); thus, inter-annotator agreement was high (Landis and Koch, 1977).

5.4 Experimental Setup

Given the relations extracted from Wikipedia, we built a German and an English graph by setting

[2] `http://www.dict.cc`
[3] `http://jbauman.com/aboutgsl.html`

Method	r
MEE	0.63
MEE-GSL	0.47
SR	0.63
SR-GSL	0.48
SO-PMI	0.58

Table 2: Correlation with human ratings

the weight of each link to the log-likelihood ratio of the two words it connects according to the corpus frequencies. There are two properties of the graph transfer algorithm that we intend to investigate. First, we are interested in the merits of applying multi edge extraction (MEE) for sentiment transfer. Second, we are interested in how the transfer quality changes when the seed lexicon is reduced in size. This way, a sparse data situation is simulated where large dictionaries are unavailable. Having these two properties in mind, four possible setups are evaluated: (i) using the full seed lexicon with all $30{,}551$ entries, but using only coordination data (*SR*), (ii) reducing the seed lexicon to $1{,}576$ entries from the General Service List (*SR-GSL*), (iii) applying MEE by adding adjective modification data (*MEE*), and (iv) using MEE with a reduced seed lexicon (*MEE-GSL*). SimRank was run for 6 iterations in all experiments. All experiments use the weight function h as described above. We show that this function improves similarities and thus lexicon induction in Laws et al. (2010).

Correlation. First, we will examine the correlation between the automatic methods (SO-PMI and the aforementioned SimRank variations) and the gold standard as done by Turney in his evaluation. For this purpose, the human ratings are mapped to float values following Table 1 and the average rating over all judges for each word is used. The correlation coefficients r are given in Table 2. Judging from these results, the ordering of SR and MEE matches the human ratings better than SO-PMI, however it decreases when using any of the GSL variations instead which can be attributed to using less data.

Classification. The correct identification of the classes *positive*, *neutral*, and *negative* is more important than the correct assignment of values on a scale since the rank ordering is debatable – this becomes apparent when measuring the agreement of human annotators. Since the assignments made by the human judges are not unanimous in most cases, the averages are distributed across the interval [-1,1]; this means that the borders between the three distinct categories are not clear. Since there is no standard evaluation for this particular problem, we need to devise a way to make the range of the neutral category dynamic. In order to find possible borders, we first assume that sentiment is distributed symmetrically around 0. We then define a threshold x which assumes the values $x \in \{\frac{i}{20} | 0 \leq i \leq 20\}$, covering the interval [0,0.5]. Since 0.5 is *slightly positive*, we do not believe that values above it are plausible. Then, each word w is positive if its human rating $\text{score}_h(w) \geq x$, negative if $\text{score}_h(w) \leq -x$, and neutral if $-x < \text{score}_h(w) < x$. The result of this process is a gold standard for the three categories for each of the values for x. The percentiles of the sizes of those categories are mapped to the values produced by the automatic methods. For example, if $x = 0.35$ means that the top 21% of all adjectives are in the positive class, the top 21% of all adjectives as assigned by SO-PMI and the SimRank varieties are positive as well.

The size of the neutral category increases the larger x becomes. Thus, high values for x are unlikely to produce a correct partitioning of the data. Since *slightly positive* was defined as 0.5, we expect the highest plausible value for x to be below that. The size of the neutral category for each value of x is given in Table 3. (Recall that the total size of the set is 143.)

We can then compute the assignment accuracy on the positive, neutral, and negative classes, as well macro- and micro-averages over these classes.

5.5 Results and Discussion

Figures 3 and 4 show the macro- and micro-averaged accuracies over the positive, negative, and neutral class for each automatic method, respectively. Overall, the SimRank variations perform better for x in the interval $[0, 0.3]$. In particular, MEE has a slightly higher accuracy than SR,

x	0.00	0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.45	0.50
# neutral	0	13	35	46	56	64	74	82	92	99	99

Table 3: Size of neutral category given x

word (translation)	**humans**	**SO**	**MEE**	**MEE-GSL**	**SR**	**SR-GSL**
chemisch (chemical)	0.00	-20.20	0.185	0.185	0.186	0.184
auferstanden (resurrected)	0.39	-10.96	-0.075	-0.577	-0.057	-0.493
intelligent (intelligent)	0.94	46.59	0.915	0.939	0.834	0.876
versiert (skilled)	0.67 -5.26 0.953 0.447 0.902 0.404					
mean	-0.04	-9.58	0.003	0.146	0.010	0.142
median	0.00	-15.60	0.110	0.157	0.114	0.157

Table 4: Example adjectives including translation, and their scores

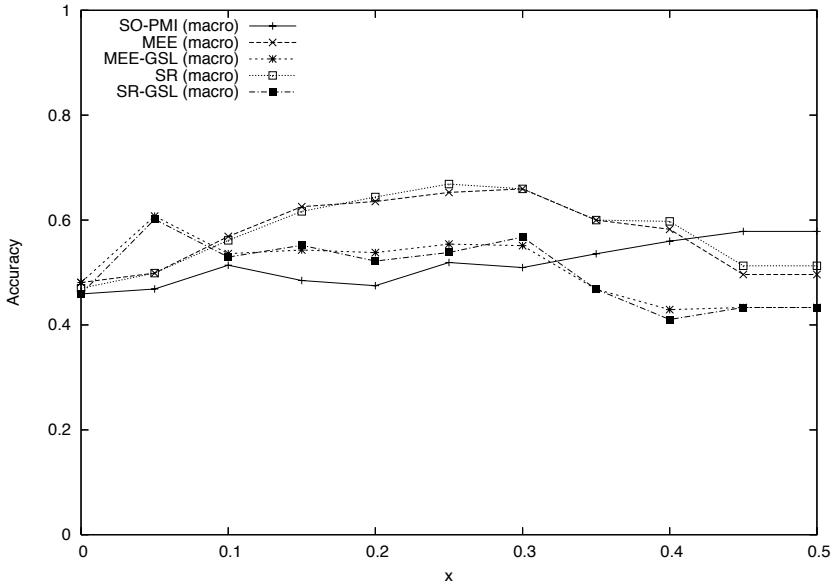


Figure 3: Macro-averaged Accuracy

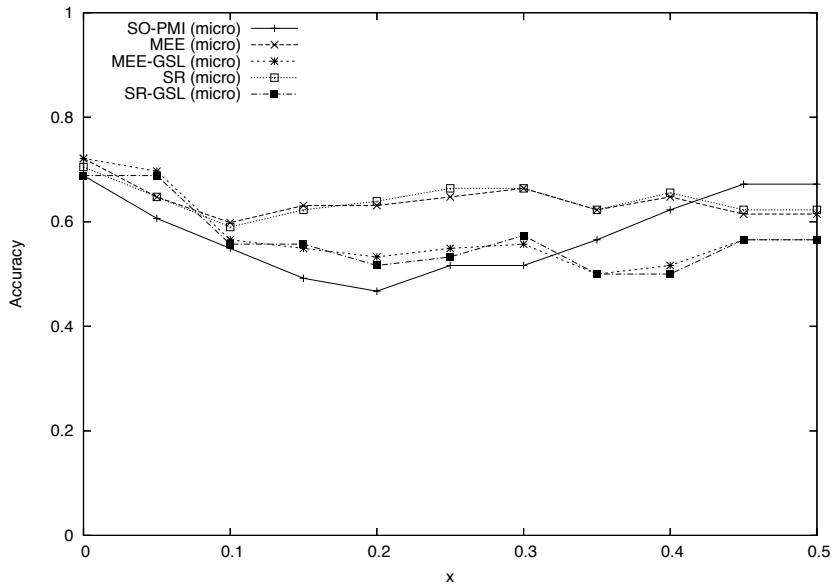


Figure 4: Micro-averaged Accuracy

however, not significantly.

Table 4 shows selected example words with their scores. These values can be understood better together with the means and medians of the respective methods which are given in the table as well. These values give us an idea of where we might expect the neutral point of a particular distribution of polarities.

Chemisch (chemical) is misclassified by SO-PMI since it occurs in negative contexts on the web. SimRank in turn was able to recognize that most words similar to *chemisch* are neutral, the most similar one being its literal translation, *chemical*. *Auferstanden (resurrected)* is an example for misclassification by SimRank which happens because the word is usually coordinated with words that have negative sentiment, e.g. *gestorben (deceased)* and *gekreuzigt (crucified)*. This problem could not be fixed by including adjective-noun modification data since the coordinations produced high log-likelihood values which lead to *dead* being the most similar word to *auferstanden*. *Intelligent* receives a score close to neutral with the original (coordination-only) training method, which could be corrected by applying MEE simply because the ordering of similar words changes through the new weighting method. Nouns modified by *intelligent* include *Leben (life)* and *Wesen (being)* whose translations are modified by positive adjectives. Many words, such as *versiert (skilled)* are classified more accurately due to the new weighting method when compared to our previous experiments (Scheible, 2010) where it received a SimRank polarity of only 0.224.

The inclusion of adjective modifications does not improve the classification results as often as we had hoped. For some cases (cf. *intelligent* mentioned above), the scores do improve, but the overall impact is limited.

6 Conclusion and Outlook

We were able to show that sentiment translation with SimRank is able to classify adjectives more accurately than SO-PMI, an unsupervised baseline method. We demonstrated that SO-PMI is outperformed by SimRank when choosing a reasonable region of neutral adjectives. In addition, we showed that the improvements of SimRank lead to better accuracy in sentiment translation in some cases. In future work, we will apply a sentiment lexicon generated with SimRank in a sentiment classification task for reviews.

The algorithms we compared are different in their purpose of application. While SO-PMI is applicable when large corpora are available for a language, it fails when used in a sparse-data situation, as noted by Turney (2002). We showed that despite reducing the seed lexicon for SimRank to a small fraction of its original size, it still performs better than SO-PMI.

Currently, our experiments are limited by the choice of using adjectives for our test set. While the examination of adjectives is highly important for sentiment analysis (as shown by Pang et al. (2002) who were able to achieve high accuracy even when using only adjectives), the application of our algorithms to a broader set of linguistic units is an important goal for future work.

Acknowledgments. We are grateful to Deutsche Forschungsgemeinschaft for funding this research as part of the WordGraph project.

References

Banea, Carmen, Rada Mihalcea, Janyce Wiebe, and Samer Hassan. 2008. Multilingual subjectivity analysis using machine translation. In *Empirical Methods in Natural Language Processing*, pages 127–135.

Christ, O., B.M. Schulze, A. Hofmann, and E. Koenig. 1999. The IMS Corpus Workbench: Corpus Query Processor (CQP): User's Manual. *University of Stuttgart, March*, 8:1999.

Dorow, Beate, Florian Laws, Lukas Michelbacher, Christian Scheible, and Jason Utt. 2009. A graph-theoretic algorithm for automatic extension of translation lexicons. In *Workshop on Geometrical Models of Natural Language Semantics*, pages 91–95.

Hatzivassiloglou, Vasileios and Kathleen R. McKeown. 1997. Predicting the semantic orientation of adjectives. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, pages 174–181.

Jeh, Glen and Jennifer Widom. 2002. Simrank: A measure of structural-context similarity. In *Proceedings of the Eighth ACM SIGKDD Interna-*

tional Conference on Knowledge Discovery and Data Mining, pages 538–543.

Landis, J.R. and G.G. Koch. 1977. The measurement of observer agreement for categorical data. *Biometrics*, 33(1):159–174.

Laws, Florian, Lukas Michelbacher, Beate Dorow, Christian Scheible, Ulrich Heid, and Hinrich Schütze. 2010. A linguistically grounded graph model for bilingual lexicon extraction. In *Proceedings of the 23nd International Conference on Computational Linguistics*.

Legendre, P. 2005. Species associations: the Kendall coefficient of concordance revisited. *Journal of Agricultural Biological and Environment Statistics*, 10(2):226–245.

Michelbacher, Lukas, Florian Laws, Beate Dorow, Ulrich Heid, and Hinrich Schütze. 2010. Building a cross-lingual relatedness thesaurus using a graph similarity measure. In *Proceedings of the Seventh Conference on International Language Resources and Evaluation*.

Mihalcea, Rada, Carmen Banea, and Janyce Wiebe. 2007. Learning multilingual subjective language via cross-lingual projections. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 976–983.

Pang, Bo, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up? Sentiment classification using machine learning techniques. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing*, pages 79–86.

Scheible, Christian. 2010. Sentiment translation through lexicon induction. In *Proceedings of the ACL 2010 Student Research Workshop*, Uppsala, Sweden. Association for Computational Linguistics.

Schmid, Helmut. 1994. Probabilistic part-of-speech tagging using decision trees. In *Proceedings of the International Conference on New Methods in Language Processing*.

Turney, Peter. 2002. Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 417–424.

Wan, Xiaojun. 2009. Co-training for cross-lingual sentiment classification. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 235–243, Suntec, Singapore, August. Association for Computational Linguistics.

West, Michael. 1953. A general service list of english words.

Widdows, Dominic and Beate Dorow. 2002. A graph model for unsupervised lexical acquisition. In *COLING*.

Wilson, Theresa, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 347–354, October.

Controlled Natural Languages for Knowledge Representation

Rolf Schwitter
Centre for Language Technology
Macquarie University
Rolf.Schwitter@mq.edu.au

Abstract

This paper presents a survey of research in controlled natural languages that can be used as high-level knowledge representation languages. Over the past 10 years or so, a number of machine-oriented controlled natural languages have emerged that can be used as high-level interface languages to various kinds of knowledge systems. These languages are relevant to the area of computational linguistics since they have two very interesting properties: firstly, they look informal like natural languages and are therefore easier to write and understand by humans than formal languages; secondly, they are precisely defined subsets of natural languages and can be translated automatically (and often deterministically) into a formal target language and then be used for automated reasoning. We present and compare the most mature of these novel languages, show how they can balance the disadvantages of natural languages and formal languages for knowledge representation, and discuss how domain specialists can be supported writing specifications in controlled natural language.

1 Introduction

Natural languages are probably the most expressive knowledge representation languages that exist; they are easy for humans to use and understand, and they are so powerful that they can even serve as their own metalanguages. Ironically, it is just this expressive quality that makes natural languages notoriously difficult for a computer to process and understand because a lot of relevant information is usually not stated explicitly in an utterance but only implied by the human author or speaker. There exist – of course – many useful resources and automated techniques that partly compensate for the lack of this background knowledge, and there are many useful applications that require only shallow processing of natural languages. But there exist – without doubt – many potential application scenarios that would benefit from deeper (axiom-based) knowledge that can be created and modified in a human-friendly way.

Formal languages (Monin, 2003) have been suggested and used as knowledge representation languages since they have a well-defined syntax, an unambiguous semantics and support automated reasoning. But these languages are often rather difficult for domain specialists to understand and cause a cognitive distance to the application domain that is not inherent in natural language.

One way to bridge the gap between a natural language and a formal language is the use of a controlled natural language (CNL) that can mediate between these languages. CNLs are engineered subsets of natural languages whose grammar and vocabulary have been restricted in a systematic way in order to reduce both ambiguity and complexity of full natural languages.

Traditionally, CNLs have been grouped into two broad categories: human-oriented CNLs and machine-oriented CNLs (Huijsen, 1998). The main objective of human-oriented CNLs is to improve the readability and comprehensibility of technical documentation (e.g. maintenance doc-

Coling 2008: Poster Volume, pages 1113–1121,
Beijing, August 2010

umentation (ASD Simplified Technical English[1]) and to simplify and standardise human-human communication for specific purposes (e.g. for trade or for air traffic control (see (Pool, 2006) for an overview)). The primary goal of machine-oriented CNLs is to improve the translatability of technical documents (e.g. machine translation (Nyberg and Mitamura, 2000)) and the acquisition, representation, and processing of knowledge (e.g. for knowledge systems (Fuchs et al., 2008) and in particular for the Semantic Web (Schwitter et al., 2008)).

Human- and machine-oriented CNLs have been designed with different goals in mind, and it is not surprising that their coverage can be quite different. O'Brien (2003) shows that there is not much overlap between the rule sets of CNLs in these two categories nor among the rule sets within a category. But since the structure of these CNLs is usually simpler and more predictable than the structure of full natural language, CNLs are in general easier for humans to understand and easier for a computer to process. An ideal CNL for knowledge representation should also be effortless to write and expressive enough to describe the problem at hand.

In this paper, we will survey machine-oriented CNLs that can be used for knowledge representation and can serve as high-level interface languages to knowledge systems. The rest of this paper is structured as follows: In Section 2, we introduce the most mature general-purpose CNLs and discuss the motivation for their design and investigate their characteristics. In Section 3, we discuss some theoretical issues regarding the expressivity and complexity of CNLs. Building on these theoretical considerations, we look in Section 4 at a number of machine-oriented CNLs that have been developed specifically as interface languages to the Semantic Web. In Section 5, we discuss the importance of supporting the writing process of CNLs in an suitable way and compare three different techniques. In Section 6, we discuss different approaches that have been used to evaluate the writability and understandability of CNLs, and finally in Section 7, we present our conclusions.

[1] http://www.asd-ste100.org/

2 General-Purpose CNLs

In this section we focus on a number of machine-oriented CNLs that have been designed to serve as knowledge representation languages. These CNLs are general-purpose languages in the sense that they have not been developed for a specific scenario or a particular application domain. These languages can be used where traditional formal languages are used otherwise. The aim of these languages is to equip domain specialists with an expressive knowledge representation language that is on the one hand easy to learn, use and understand and on the other hand fully processable by a computer.

2.1 Attempto Controlled English (ACE)

ACE (Fuchs et al., 2008) is a CNL that covers a well-defined subset of English that can be translated unambiguously into first-order logic via discourse representation structures (Kamp and Reyle, 1993) and then be used for automated reasoning. ACE is defined by a small set of construction rules that describe its syntax and a small set of interpretation rules that disambiguate constructs that might appear ambiguous in full English. The vocabulary of ACE consists of predefined function words (e.g. determiners, conjunctions, and pronouns), some predefined fixed phrases (e.g. *there is, it is false that*), and content words (nouns, proper names, verbs, adjectives, and adverbs). ACE supports language constructs such as:

- active and passive verbs (and modal verbs);
- strong negation (e.g. *no, does not*) and weak negation (e.g. *is is not provable that*);
- subject and object relative clauses;
- declarative, interrogative, imperative and conditional sentences;
- various forms of anaphoric references to noun phrases (e.g. *he, himself, the man, X*).

It is important to note that the meaning of words in ACE is not predefined; the user is expected to define their meaning by ACE sentences or import these definitions from an existing formal ontology.

Here is a simple example of an ACE text together with a question:

> Every company that buys at least three machines gets a discount. Six Swiss companies each buy one machine. A German company buys four machines. Who gets a discount?

Note that ACE uses disambiguation markers (e.g. *each*) on the surface level and mathematical background knowledge about natural numbers in order to answer the question above. This mathematical knowledge is implemented as a set of Prolog predicates which are executed during the proof (question answering process).

ACE is supported by various tools[2], among them a text editor that helps users to construct correct ACE sentences with the help of hints and error messages, a parser that translates ACE texts into discourse representation structures, a paraphraser that reflects the interpretation of the machine in CNL, and a Satchmo-style reasoning engine that can be used for consistency and redundancy checking as well as for question answering. Applications of ACE include software and hardware specifications, agent control, legal and medical regulations, and ontology construction.

2.2 Processable English (PENG)

PENG (White and Schwitter, 2009) is a CNL that is similar to ACE but adopts a more light-weight approach in the sense that it covers a smaller but fully tractable subset of English. The language processors of ACE and PENG are both based on grammars that are written in a definite clause grammar (DCG) notation. These DCGs are enhanced with feature structures and specifically designed to translate declarative and interrogative sentences into a first-order logic notation via discourse representation structures. In contrast to the original version of ACE that uses the DCG directly and resolves anaphoric references only after a discourse representation structure has been constructed, PENG transforms the DCG into a format that can be processed by a top-down chart parser and resolves anaphoric references during the parsing process while a discourse representation structure is built up. PENG has been designed for an incremental parsing approach and was the first CNL that was supported by a predictive editor (Schwitter et al., 2003). The PENG system provides text- and menu-based writing support that removes some of the burden of learning and remembering the constraints of the CNL from the user and generates a paraphrase that clarifies the interpretation for each sentence that the user enters. PENG's text editor dynamically enforces the grammatical restrictions of the CNL via look-ahead information while a text is written. For each word form that the user enters into the editor, a list of options is generated incrementally by the chart parser to inform the user about how the structure of the current sentence can be continued. The syntactic restrictions ensure that the text follows the rules of the CNL so that it can be translated unambiguously into the formal target language (first-order logic) and be processed by a theorem prover.

In order to illustrate how PENG can be used to reconstruct a problem in controlled natural language, we use an example from the TPTP problem library[3]. The problems in this library are usually used to test the capacity of automated reasoning tools and are translated manually by a human into the formal target language. For reasons of space, we use here one of the simpler problems of the library; the puzzle PUZ012-1 below is also known as "The Mislabeled Boxes":

> There are three boxes a, b, and c on a table. Each box contains apples or bananas or oranges. No two boxes contain the same thing. Each box has a label that says it contains apples or says it contains bananas or says it contains oranges. No box contains what it says on its label. The label on box a says "apples". The label on box b says "oranges". The label on box c says "bananas". You pick up box b and it contains apples. What do the other two boxes contain?

In order to solve this puzzle by a computer, we have to reconstruct it and augment it with the

[2] http://attempto.ifi.uzh.ch/site/tools/

[3] http://www.cs.miami.edu/~tptp/

relevant background knowledge. The main problems that we face here for machine-processing are the following ones: some of the constructions in the problem description are ambiguous (e.g. the antecedent for the personal pronoun *it* is open to two interpretations); the semantic relation between some content words is not explicit (e.g. the relation between the actual things in the box and the names on the labels that describe these things); and some of the constructions are not relevant at all for the solution of the problem (e.g. that the three boxes are on the table). Here is a possible reconstruction of this puzzle in PENG:

> The label of the box a says APPLES. The label of the box b says ORANGES. The label of the box c says BANANAS. APPLES stands for apples. ORANGES stands for oranges. BANANAS stands for bananas. All apples are fruits. All bananas are fruits. All oranges are fruits. Each box contains the apples or contains the bananas or contains the oranges. It is not the case that a box contains fruits and that the label of the box says something that stands for those fruits. It is not the case that a box X contains fruits and that a box Y contains those fruits. The box b contains the apples. What does the box a contain? What does the box c contain?

Note that this reconstruction makes information that is implicit or only assumed in the original problem description explicit in PENG.

PENG has recently been used for the construction of an interface to a situation awareness system (Baader et al., 2009) but the language can be used for similar applications to ACE.

2.3 Computer Processable Language (CPL)

CPL (Clark et al., 2010) is a controlled language that has been developed at Boeing Research and Technology. In contrast to ACE which applies a small set of strict interpretation rules, and in contrast to PENG, which relies on a predictive editor, the CPL interpreter directly resolves various types of ambiguities using heuristic rules for prepositional phrase attachment, word sense disambiguation, semantic role labeling, compound noun interpretation, metonymy resolution, and other language processing activities.

CPL accepts three types of sentences: ground facts, questions, and rules. In the case of ground facts, a basic CPL sentence takes one of the following three forms:

- There is|are *NP*
- *NP verb* [*NP*] [*PP*]*
- *NP* is|are *passive-verb* [by *NP*] [*PP*]*

Verbs can include auxiliaries and particles, and nouns in noun phrases can be modified by other nouns, prepositional phrases, and adjectives. In the case of questions, CPL accepts five forms; the two main forms are:

- What is *NP*?
- Is it true that *Sentence*?

In the case of rules, CPL accepts sentence patterns of the form:

- IF *Sentence* [AND *Sentence*]* THEN *Sentence* [AND *Sentence*]*

Parsing of CPL is performed bottom-up with the help of a broad coverage chart parser that uses preference for common word attachment patterns stored in a manually constructed database. During parsing, a simplified logical form is generated for basic sentences by rules that run in parallel to the grammar rules. There is no explicit quantifier scoping for these basic sentences and some disambiguation decisions (e.g., word sense and semantic relationships) are deferred and handled by the inference engine that makes a "best guess" of word sense assignments using WordNet[4]. The logical form is used to generate ground Knowledge Machine (KM) assertions. KM[5] is a frame-based language with first-order semantics. The KM interpreter employs a sophisticated machinery for reasoning, including reasoning about actions using a situation calculus mechanism. Rules

[4] http://wordnet.princeton.edu/

[5] http://userweb.cs.utexas.edu/users/mfkb/km.html

are entered by the user who writes CPL sentences with the help of rule templates. There exist seven templates for this purpose: three of them create standard logical implications and the rest describe preconditions and effects of actions. Each CPL sentence is interpreted interactively. The system paraphrases its interpretation back to the user, allowing the user to spot and fix misinterpretations. Sentences that express states add facts to a situation, and sentences that express actions trigger rules that update the situation, reflecting the changes that the action has on the situation. The user can ask questions about an emerging situation directly in CPL.

While CPL relies on heuristics, CPL-Lite is a slimmed down version of CPL that can be interpreted deterministically in a similar fashion to PENG. Each CPL-Lite sentence corresponds to a single binary relation between two entities. CPL-Lite distinguishes three types of relations: noun-like relations (e.g. the age of $<x>$ is $<y>$), verb-like relations (e.g. $<x>$ causes $<y>$), and preposition-like relations (e.g. $<x>$ is during $<y>$).

Interestingly, CPL-Lite has the same expressivity as CPL, but CPL-Lite is more verbose and grammatically more restricted. For example, the following two CPL sentences:

1. A man drives a car along a road for 1 hour.
2. The speed of the car is 30 km/h.

can be expressed (or better reconstructed) in an unambiguous way in CPL-Lite:

3. A person drives a vehicle.
4. The path of the driving is a road.
5. The duration of the driving is 1 hour.
6. The speed of the driving is 30 km/h.

Note that the user used here the noun *person* instead of *man* and *vehicle* instead of *car* during this reconstruction process because only these words were available in the system's ontology.

CPL and CPL-Lite have been mainly used to encode general and domain specific commonsense knowledge and to allow knowledge engineers to pose queries in a comprehensible way.

2.4 Other General-Purpose CNLs

Common Logic Controlled English (CLCE)[6] is a proposal for a CNL – similar to ACE and PENG – that has been designed as a human interface language for the ISO standard Common Logic (CL)[7]. However, CLCE itself is not part of this standard but uses Common Logic semantics. CLCE supports full first-order logic with equality supplemented with an ontology for sets, sequences, and integers. The primary syntactic restrictions are the use of present tense verbs, singular nouns, and variables instead of pronouns. Despite these limitations, CLCE can express the kind of English used in software specifications, mathematics textbooks, and definitions and axioms found in formal ontologies.

Formalized-English (Martin, 2002) is another proposal for a CNL that can be used as a general knowledge representation language. This language has a relatively simple structure and is derived from a conventional knowledge representation language. Formalized-English contains a number of formal-looking language elements and is therefore not a strict subset of standard English.

3 Theoretical Considerations

During the design of a CNL one has to pay attention to two important theoretical issues: the expressive power of the envisaged language and its computational complexity. E2V (Pratt-Hartmann, 2003) is a CNL that mainly grew out of theoretical studies about the expressivity and complexity of natural language fragments. E2V corresponds to the decidable two-variable fragment of first-order logic ($\mathcal{L}^2$). This fragment is interesting since it has the so-called finite model property. That means if a formula of $\mathcal{L}^2$ is satisfiable, then it is satisfiable in a finite model. E2V includes determiners (*every, no, a*), nouns, transitive verbs, verb phrase negation, relative, reflexive, and personal pronouns. Without any writing support it is difficult to decide if a sentence is in E2V or not. For example, one reading of sentence (7) is in E2V, the other one is not:

[6] http://www.jfsowa.com/clce/specs.htm
[7] ISO/IEC24707:2007

7. Every artist who employs a carpenter despises every beekeeper who admires him.

On the syntactic level, E2V is a subset of ACE with the exception that pronouns (e.g. *him*) always refer to the closest (acceptable) noun in the syntax tree (e.g. *artist*) and not to the closest (acceptable) noun that occurs in the surface structure (e.g. *carpenter*). This is because the E2V interpretation relies on the two-variable fragment of first-order logic. Note that sentence (7) has the following two possible representations (8 and 9) in first-order logic:

8. $\forall x_1$ (artist(x_1) & $\exists x_2$ (carpenter(x_2) & employ(x_1,x_2)) -> $\forall x_3$ (beekeeper(x_3) & admire(x_3,x_1) -> despise(x_1,x_3)))

9. $\forall x_1$ $\forall x_2$ (artist(x_1) & (carpenter(x_2) & employ(x_1,x_2) -> $\forall x_3$ (beekeeper(x_3) & admire(x_3,x_2) -> despise(x_1,x_3)))

Although there are three variables in the formula (8) that correspond to the three nouns in sentence (7), the variables x_2 and x_3 never occur free in the same sub-formula. Therefore, the number of variables can be reduced by replacing x_3 through x_2. This technique can not be applied to the variables in formula (9).

E2V has been extended in various ways (Pratt-Hartmann and Third, 2006) and one extension includes counting determiners (e.g. *at least three, at most five, exactly four*). These determiners will not in general translate into the two-variable fragment of first-order logic, but into the fragment $\mathcal{C}^2$, which adds counting quantifiers to the two-variable fragment. The satisfiability problem of this fragment is still decidable and its expressivity and computational complexity is similar to those description logic languages that build the foundation of the Semantic Web.

4 CNLs for the Semantic Web

Recently, a number of CNLs have been developed that can serve as front-end to those formal languages that are used in the context of the Semantic Web[8]. These CNLs can be used by domain specialists who prefer familiar natural language-like notations over formal ones for authoring and verbalising formal ontologies.

ACE View (Kaljurand, 2007) is a CNL editor that supports a defined subset of ACE that can be used as an alternative syntax for the Semantic Web languages OWL and SWRL. ACE View integrates two mappings: one from ACE to OWL/SWRL and one from OWL to ACE. These mappings are not bidirectional in a strict sense since the OWL to ACE mapping also covers OWL axioms and expression types that the ACE to OWL mapping does not generate.

Sydney OWL Syntax (SOS) (Cregan et al., 2007) is a proposal for a CNL that builds upon PENG and provides a syntactically bidirectional mapping to OWL-DL. SOS is strictly bidirectional: each statement can be translated into OWL functional-style syntax and vice versa. The bidirectional translation is achieved with the help of a definite clause grammar that generates the target notation during the parsing process. In contrast to ACE, syntactic constructs of OWL are always carried over one-to-one to SOS. Thus, semantically equivalent OWL statements that use different syntactical constructs are always mapped to different SOS statements.

Rabbit (Hart et al., 2008) is a CNL designed for a scenario where a domain expert and an ontology engineer work together to build an ontology. The construction process is supported by a text-based ontology editor. The editor accepts Rabbit sentences, helps to resolve possible syntax errors, and translates well-formed sentences into OWL. The semantics of some Rabbit constructs is controversial (e.g. exclusive interpretation of disjunction) and hard to align with the semantics of OWL.

Lite Natural Language (Bernardi et al., 2007) is a CNL based on Categorial Grammar; it has the same expressivity as the description logic DL-Lite. DL-Lite is a tractable fragment of OWL and has polynomial time complexity for the main reasoning tasks. DL-Lite is expressive enough to capture relational databases and UML (Unified Modeling Language) diagrams.

[8] http://www.w3.org/TR/owl2-overview/

CLOnE (Funk et al., 2007) is a CNL that is built on top of the natural language processing framework GATE[9]. CLOnE is a simple ontology authoring language that consists of eleven sentence patterns which roughly correspond to eleven OWL axiom patterns. It is unclear whether CLOnE can be extended in a systematic way to cover larger fragments of OWL.

The three controlled languages ACE, SOS, and Rabbit are compared in more detail in Schwitter et al. (2008). There exist three other CNL research streams that are closely related to the Semantic Web: CNLs for querying Semantic Web content (Bernstein and Kaufmann, 2006); CNLs for maintaining semantic wikis (Kuhn, 2009; Kuhn, 2010); and CNLs for describing rules and policies (De Coi et al., 2009).

5 Writing Support for CNLs

Writing a specification in CNL is not an easy task since the author has to stick to the rules of the controlled language. Writing in CNL is in essence a normative process that prescribes how humans should use language to communicate effectively with a computer in order to achieve a particular goal. The challenge here is to develop interface techniques that make the writing process as unobtrusive and effortless as possible. Three main techniques have been suggested to support the writing process of CNLs: the use of error messages, conceptual authoring, and predictive feedback.

Error messages seem to be the most obvious way to support the writing of a text in CNL, and many CNLs (among them (Clark et al., 2010; Fuchs et al., 2008)) use this technique. The user is supposed to learn and remember the restrictions of the CNL and then to write the text following the memorised rules. If the parsing process fails, then the CNL system tries to identify the cause of the error and provides one or more suggestions for how to fix the error. The problem with this technique is that the input might be an unrestricted sentence and a useful error message would require in the worst case knowledge of the sort that is needed for processing full natural language.

[9] http://gate.ac.uk/

Conceptual authoring (Power et al., 2009) is a technique that allows authors to edit a knowledge base on the semantic level by refining specific categories and properties that occur in CNL sentences via a hierarchy of menu options. The selection of an option by the author results in an update of the underlying model and triggers the generation of a new sentence that can then be further refined. This method relies on natural language generation techniques and makes the analysis of CNL sentences unnecessary. The problem with this technique is that it does not allow the author to specify new knowledge that is not already encoded in the knowledge base; it is basically a technique for knowledge authoring and visualization and does not provide an independent knowledge representation language.

Predictive feedback (Schwitter et al., 2003; Kuhn and Schwitter, 2008) is a technique that informs the authors during the writing process about the approved structures of the CNL. This technique relies on interfaces that are aware of the grammar and can look-ahead within this grammar. Using this technique the author receives immediate feedback while a text is written and cannot enter sentences that are not in the scope of the grammar. The grammar of the language PENG has been designed from the beginning to be used in a predictive editor and is processed by a chart parser that is able to generate the look-ahead information. The following example illustrates how a predictive editor works:

- A [adjective | common noun]
- A man [verb | who | 'does not']
- A man works ['.' | preposition | adverb]

In this example the look-ahead information consists of syntactic categories, word forms and punctuation marks; all these elements are implemented as hypertext links. Selecting a hypertext link for a syntactic category displays approved word forms and selecting a word form or a punctuation mark directly adds this element to the text. Kuhn (2010) shows in an number of experiments that predictive editors are easy for untrained users to use and argues that predictive feedback is the best way to support the writing process of CNLs.

6 Evaluating CNLs

Over the past years, a number of different user experiments have been designed to measure various usability aspects of CNLs (see (Kuhn, 2010) for an introduction). These experiments can be grouped into three different categories: task-based experiments, paraphrase-based experiments, and graph-based experiments.

In task-based experiments (for example, (Kaufmann and Bernstein, 2007)), human subjects receive a certain task that requires them to use a CNL as an interface language to a knowledge base together with a tool that potentially supports the writing process. These experiments test how easy or difficult it is to write in these controlled languages using the given tool, but they do not test the understandability of these languages.

Paraphrase-based experiments (for example, (Hart et al., 2008)) aim to evaluate the understandability of a CNL in a tool-independent way. Human subjects receive a statement in CNL and a choice of paraphrases in full natural language, and then have to select the correct paraphrase. These experiments scale well with the expressivity of the CNL but it is difficult to guarantee that the paraphrases are understood in the intended way.

Graph-based experiments (for example, (Kuhn, 2010)) try to overcome the problems of paraphrase-based experiments. In order to test the understandability of CNLs and formal languages, a graph-based notation is used to describe a situation accompanied with statements in the language to be tested. The human subjects have to decide which of these statements are true and which ones are false with respect to the situation illustrated by the graph notation.

The reported results of these experiments in the literature provide strong evidence that CNLs are easier to write and easier to understand for domain specialists than formal languages.

7 Conclusions

It is an exciting time to work on controlled natural languages. In this paper, we surveyed a number of machine-oriented controlled natural languages that can be used instead of formal languages for representing knowledge. These controlled natural languages look like English but correspond to a formal target language. Anyone who can read English has already the basic skills to understand these controlled natural languages. Writing a specification in controlled natural language is a bit harder: it requires that the author either learns the language in order to be able to stay within its syntactic and semantic restrictions or that he uses an intelligent authoring tool that supports the writing process and enforces the restrictions of the language.

Machine-oriented controlled natural languages can be translated automatically (and often deterministically) into a formal target language (e.g. into full first-order logic or into a version of description logics). These languages can be used to express the kind of information that occurs in software specifications, formal ontologies, business rules, and legal and medical regulations.

In summary, an ideal machine-oriented controlled natural language should fulfill at least the following requirements: (a) it should have a well-defined syntax and a precise semantics that is defined by an unambiguous mapping into a logic-based representation; (b) it should look as natural as possible and be based on a subset of a certain natural language; (c) it should be easy for humans to write and understand and easy for a machine to process; and (d) it should have the necessary expressivity that is required to describe a problem in the respective application domain.

Of course these requirements can be in conflict with each other and therefore careful compromises need to be made when a new controlled natural language is designed. This design process offers many interesting research challenges for researchers in the area of computational linguistics and artificial intelligence. This research is driven by the overall goal to close the gap between natural and formal languages and to allow for true collaboration between humans and machines in the near future.

Acknowledgments

I would like to thank to three anonymous reviewers of Coling 2010 for their valuable feedback and to Robert Dale for comments and suggestions on previous versions of this paper.

References

Baader, Franz, Andreas Bauer, Peter Baumgartner, Anne Cregan, Alfredo Gabaldon, Krystian Ji, Kevin Lee, Dave Rajaratnam and R. Schwitter. 2009. A Novel Architecture for Situation Awareness Systems, In: *Proceedings of TABLEAUX 2009*, LNAI 5607, pp. 77–92.

Bernardi, Raffaella, Diego Calvanese, and Camilo Thorne. 2007. Lite Natural Language. In: *Proceedings of IWCS-7*.

Bernstein, Abraham and Esther Kaufmann. 2006. GINO – a guided input natural language ontology editor. In: *Proceedings of ISWC 2006*, LNCS 4273, pp. 144–157.

Clark, Peter, Phil Harrison, William R. Murray, and John Thompson. 2010 Naturalness vs. Predictability: A Key Debate in Controlled Languages. In: *Proceedings 2009 Workshop on Controlled Natural Languages (CNL'09)*.

Cregan, Anne, Rolf Schwitter, and Thomas Meyer. 2007. Sydney OWL Syntax – towards a Controlled Natural Language Syntax for OWL 1.1. In: *Proceedings of OWLED 2007*, CEUR, vol. 258.

De Coi, Juri L., Norbert E. Fuchs, Kaarel Kaljurand, Tobias Kuhn. 2009. Controlled English for Reasoning on the Semantic Web. In: *LNCS*, vol. 5500, pp. 276–308.

Fuchs, Norbert E., Kaarel Kaljurand, and Tobias Kuhn. 2008. Attempto Controlled English for Knowledge Representation. In: *Reasoning Web*, LNCS, vol. 5224, pp. 104–124.

Funk, Adam, Valentin Tablan, Kalina Bontcheva, Hamish Cunningham, Brian Davis, and Siegfried Handschuh. 2007. CLOnE: Controlled Language for Ontology Editing. In: *Proceedings of ISWC 2007*.

Hart, Glen, Martina Johnson, and Catherine Dolbear. 2008. Rabbit: Developing a controlled natural language for authoring ontologies. In: *Proceedings of ESWC 2008*, LNCS, vol. 5021, pp. 348–360.

Huijsen, Willem-Olaf. 1998. Controlled Language – An Introduction. In: *Proceedings of CLAW 98*, pp. 1–15.

Kaljurand, Kaarel. 2007. Attempto Controlled English as a Semantic Web Language. *PhD Thesis*. Faculty of Mathematics and Computer Science, University of Tartu.

Kamp, Hans and Uwe Reyle. 1993. *From Discourse to Logic*. Kluwer, Dordrecht.

Kaufmann, Esther and Abraham Bernstein. 2007. How Useful Are Natural Language Interfaces to the Semantic Web for Casual End-Users? In: *Proceedings of ISWC/ASWC 2007*, NLCS, vol. 4825, pp. 281–294.

Kuhn, Tobias and Rolf Schwitter. 2008. Writing Support for Controlled Natural Languages. In: *Proceedings of ALTA 2008*, pp. 46–54.

Kuhn, Tobias. 2009. How controlled English can improve semantic wikis. In: *Proceedings of SemWiki 2009*, CEUR, vol. 464.

Kuhn, Tobias. 2010. Controlled English for Knowledge Representation. *Doctoral Thesis*. Faculty of Economics, Business Administration and Information Technology of the University of Zurich.

Martin, Philippe. 2002. Knowledge representation in CGLF, CGIF, KIF, Frame-CG and Formalized-English. In: *Proceedings of ICCS 2002*, LNAI, vol. 2393, pp. 77–91.

Monin, Jean-François. 2003. *Understanding Formal Methods.* Springer-Verlag, London.

Nyberg, Eric H. and Teruko Mitamura. 2000. The KANTOO Machine Translation Environment. In: *Proceedings of AMTA 2000*, LNCS, vol. 1934, pp. 192–195.

O'Brien, Sharon. 2003. Controlling controlled english – an analysis of several controlled language rule sets. In: *Proceedings of EAMT-CLAW 03*, Dublin City University, Ireland, pp. 105–114.

Pool, Jonathan. 2006. Can Controlled Languages Scale to the Web? In: *Proceedings of the 5th Int. Workshop on Controlled Language Applications*.

Power, Richard, Robert Stevens, Donia Scott, and Alan Rector. 2009. Editing OWL through generated CNL. In: *Pre-Proceedings of the Workshop on CNL 2009*, CEUR, vol. 448.

Pratt-Hartmann, Ian. 2003. A two-variable fragment of English. In: *Journal of Logic, Language and Information*, 12(1), pp. 13–45.

Pratt-Hartmann, Ian and Allan Third. 2006. More fragments of language: the case of ditransitive verbs. In: *Notre Dame Journal of Formal Logic*, 47(2), pp. 151–177.

Schwitter, Rolf, Anna Ljungberg, and David Hood. 2003. ECOLE – A Look-ahead Editor for a Controlled Language. In: *Proceedings of EAMT-CLAW03*, pp. 141–150.

Schwitter, Rolf, Kaarel Kaljurand, Anne Cregan, Catherine Dolbear, and Glen Hart. 2008. A comparison of three controlled natural languages for OWL 1.1. In: *Proceedings of OWLED 2008*, CEUR, vol. 496.

White, Colin and Rolf Schwitter. 2009. An Update on PENG Light. In: *Proceedings of ALTA 2009*, pp. 80–88.

Informed ways of improving data-driven dependency parsing for German

Wolfgang Seeker
University of Stuttgart
Inst. für Maschinelle Sprachverarbeitung
seeker@ims.uni-stuttgart.de

Bernd Bohnet
University of Stuttgart
Inst. für Maschinelle Sprachverarbeitung
Bernd.Bohnet@ims.uni-stuttgart.de

Lilja Øvrelid
University of Potsdam
Institut für Linguistik
ovrelid@uni-potsdam.de

Jonas Kuhn
University of Stuttgart
Inst. für Maschinelle Sprachverarbeitung
jonas@ims.uni-stuttgart.de

Abstract

We investigate a series of targeted modifications to a data-driven dependency parser of German and show that these can be highly effective even for a relatively well studied language like German if they are made on a (linguistically and methodologically) informed basis and with a parser implementation that allows for fast and robust training and application. Making relatively small changes to a range of very different system components, we were able to increase labeled accuracy on a standard test set (from the CoNLL 2009 shared task), ignoring gold standard part-of-speech tags, from 87.64% to 89.40%. The study was conducted in less than five weeks and as a secondary project of all four authors. Effective modifications include the quality and combination of auto-assigned morphosyntactic features entering machine learning, the internal feature handling as well as the inclusion of global constraints and a combination of different parsing strategies.

1 Introduction

The past years have seen an enormous surge of interest in dependency parsing, mainly in the data-driven paradigm, and with a particular emphasis on covering a whole set of languages with a single approach. The reasons for this interest are manifold; the availability of shared task data from various CoNLL conferences (among others (Buchholz and Marsi, 2006; Hajič et al., 2009)), comprising collections of languages based on a single representation format, has certainly been instrumental. But likewise, the straightforward usefulness of dependency representations for a number of tasks plays an important role. The relative language independence of the representations makes dependency parsing particularly attractive for multilingually oriented work, including machine translation.

As data-driven approaches to dependency parsing have reached a certain level of maturity, it may appear as if further improvements of parsing performance have to rely on relatively advanced tuning procedures, such as sophisticated automatic feature selection procedures or combinations of different parsing approaches with complementary strengths. It is indeed still hard to pinpoint the structural properties of a language (or annotation scheme) that make the parsing task easier for a particular approach, so it may seem best to leave the decision to a higher-level procedure.

This paper starts from the suspicion that while sophisticated tuning procedures are certainly helpful, one should not underestimate the potential of relatively simple modifications of the experimental set-up, such as a restructuring of aspects of the dependency format, a targeted improvement of the quality of automatically assigned features, or a simplification of the feature space for machine learning – the modifications just have to be made in an informed way. This

Coling 2008: Poster Volume, pages 1122–1130,
Beijing, August 2010

presupposes two things: (i) a thorough linguistic understanding of the issues at hand, and (ii) a relatively powerful and robust experimental machinery which allows for experimentation in various directions and which should ideally support a fast turn-around cycle.

We report on a small pilot study exploring the potential of relatively small, informed modifications as a way of improving parsing accuracy even for a language that has received considerable attention in the parsing literature, including the dependency parsing literature, namely German. Within a timeframe of five weeks and spending only a few hours a day on the project (between a group of four people), we were able to reach some surprising improvements in parsing accuracy.

By way of example, we experimented with modifications in a number of rather different system areas, which we will discuss in the course of this paper after a brief discussion of related work and the data basis in Section 2. Based on a second-order maximum spanning tree algorithm, we used a hash kernel to facilitate the mapping of the features onto their weights for a very large number of features (Section 3); we modified the dependency tree representation for prepositional phrases, adding hierarchical structure that facilitates the picking up of generalizations (Section 4). We take advantage of a morphological analyzer to train an improved part-of-speech tagger (Section 5), and we use knowledge about the structure of morphological paradigms and the morphology-syntax interface in the feature design for machine learning (Section 6). As is known from other studies, the combination of different parsing strategies is advantageous; we include a relatively simple parser stacking procedure in our pilot study (Section 7), and finally, we apply Integer Linear Programming in a targeted way to add some global constraints on possible combinations of arc labels with a single head (Section 8). Section 9 offers a brief conclusion.

2 Related Work and Data Basis

We quickly review the situation in data-driven dependency parsing in general and on applying it to German specifically.

The two main approaches to data-driven dependency parsing are transition based dependency parsing (Nivre, 2003; Yamada and Matsumoto, 2003; Titov and Henderson, 2007) and maximum spanning tree based dependency parsing (Eisner, 1996; Eisner, 2000; McDonald and Pereira, 2006). Transition based parsers typically have a linear or quadratic complexity (Attardi, 2006). Nivre (2009) introduced a transition based non-projective parsing algorithm that has a worst case quadratic complexity and an expected linear parsing time. Titov and Henderson (2007) combined a transition based parsing algorithm, using beam search, with a latent variable machine learning technique.

Maximum spanning tree based dependency parsers decompose a dependency structure into *factors*. The factors of the first order maximum spanning tree parsing algorithm are edges consisting of the head, the dependent (child) and the edge label. This algorithm has a quadratic complexity. The second order parsing algorithm of McDonald and Pereira (2006) uses a separate algorithm for edge labeling. In addition to the first order factors, this algorithm uses the edges to those children which are closest to the dependent and has a complexity of O(n^3). The second order algorithm of Carreras (2007) uses in addition to McDonald and Pereira (2006) the child of the dependent occurring in the sentence between the head and the dependent as well as the edge from the dependents to a grandchild. The edge labeling is an integral part of the algorithm which requires an additional loop over the labels. This algorithm therefore has a complexity of O(n^4). Johansson and Nugues (2008) reduced the required number of loops over the edge labels by considering only the edges that existed in the training corpus for a distinct head and child part-of-speech tag combination.

Predating the surge of interest in data-based dependency parsing, there is a relatively long tradition of dependency parsing work on German, including for instance Menzel and Schröder (1998) and Duchier and Debusmann (2001). German was included in the CoNLL shared tasks in 2006 (Multilingual Dependency Parsing, (Buchholz and Marsi, 2006)) and in 2009 (Syntactic and Semantic Dependencies in Multiple Languages, (Hajič et al., 2009)) with data based on the TIGER

corpus (Brants et al., 2002) in both cases. Since the original TIGER treebank is in a hybrid phrase-structural/dependency format with a relatively flat hierarchical structure, conversion to a pure dependency format involves some non-trivial steps. The 2008 ACL Workshop on Parsing German included a specific shared task on dependency parsing of German (Kübler, 2008), based on two sets of data: again the TIGER corpus – however with a different conversion routine than for the CoNLL tasks – and the TüBa-D/Z corpus (Hinrichs et al., 2004).

In the 2006 CoNLL task and in the 2008 ACL Workshop task, the task was dependency parsing with given gold standard part-of-speech tags from the corpus. This is a valid way of isolating the specific subproblem of parsing, however it is clear that the task does not reflect the application setting which includes noise from automatic part-of-speech tagging. In the 2009 CoNLL task, both gold standard tags and automatically assigned tags were provided. The auto-tagged version was created with the standard model of the TreeTagger (Schmid, 1995) (i.e., with no domain-specific tagger training).

In our experiments, we used the data set from the 2009 CoNLL task, for which the broadest comparison of recent parsing approaches exists. The highest-scoring system in the shared task was Bohnet (2009) with a labeled accuracy (LAS) of 87.48%, on auto-tagged data. The highest-scoring (in fact the only) system in the dependency parsing track of the 2008 ACL Workshop on parsing German was Hall and Nivre (2008) with an LAS of 90.80% on gold-tagged data, and with a data set that is not comparable to the CoNLL data.[1]

3 Hash Kernel

Our parser is based on a second order maximum spanning tree algorithm and uses MIRA (Crammer et al., 2006) as learning technique in combination with a hash kernel. The hash kernel has a higher accuracy since it can use additional features found during the creation of the dependency tree in addition to the features extracted from the training examples. The modification to MIRA is simple: we replace the feature-index mapping that maps the features to indices of the weight vector by a random function. Usually, the feature-index mapping in the support vector machine has two tasks: The mapping maps the features to an index and it filters out features that never occurred in a dependency tree. In our approach, we do not filter out these features, but use them as additional features. It turns out that this choice improves parsing quality. Instead of the feature-index mapping we use the following hash function:[2]

$h \leftarrow |(l\ xor(l \vee \text{0xffffffff00000000} >> 32))\%\ \text{size}|$

The Hash Kernel for structured data uses the hash function $h : J \rightarrow \{1...n\}$ to index ϕ where ϕ maps the observations X to a feature space. We define $\phi(x, y)$ as the numeric feature representation indexed by J. The learning problem is to fit the function F so that the errors of the predicted parse tree y are as low as possible. The scoring function of the Hash Kernel is defined as:[3]

$$F(x, y) = \overrightarrow{w} * \overline{\phi}(x, y)$$

For different j, the hash function $h(j)$ might generate the same value k. This means that the hash function maps more than one feature to the same weight which causes weight collisions. This procedure is similar to randomization of weights (features), which aims to save space by sharing values in the weight vector (Blum, 2006; Rahimi and Recht, 2008). The Hash Kernel shares values when collisions occur that can be considered as an approximation of the kernel function, because a weight might be adapted due to more than one feature. The approximation works very well with a weight vector size of 115 million values.

With the Hash Kernel, we were able to improve on a baseline parser that already reaches a quite high LAS of 87.64% which is higher than the top score for German (87.48%) in the CoNLL Shared task 2009. The Hash Kernel improved that value by 0.42 percentage points to 88.06%. In addition to that, we obtain a large speed up in terms of parsing time. The baseline parser spends an average of 426 milliseconds to parse a sentence of the test

[1] To get an idea of how the data sets compare, we trained the version of our parser described in Section 3 (i.e., without most of the linguistically informed improvements) on this data, achieving labeled accuracy of 92.41%, compared to 88.06% for the 2009 CoNLL task version.

[2] $>> n$ shifts n bits right, and % is the modulo operation.

[3] $\overrightarrow{w}$ is the weight vector and the size of $\overrightarrow{w}$ is n.

set and the parser with Hash Kernel only takes 126 milliseconds which is an increase in speed of 3.4 times. We get the large speed up because the memory access to a large array causes many CPU cache misses which we avoid by replacing the feature-index mapping with a hash function. As mentioned above, the speedup influences the experimenters' opportunities for explorative development since it reduces the turnaround time for experimental trials.

4 Restructuring of PPs

In a first step, we applied a treebank transformation to our data set in order to ease the learning for the parser. We concentrated on prepositional phrases (PP) to get an idea how much this kind of transformation can actually help a parser. PPs are notoriously flat in the TIGER Treebank annotation (from which our data are derived) and they do not embed a noun phrase (NP) but rather attach all parts of the noun phrase directly at PP level. This annotation was kept in the dependency version and it can cause problems for the parser since there are two different ways of annotating NPs: (i) for normal NPs where all dependents of the noun are attached as daughters of the head noun and (ii) for NPs in PPs where all dependents of the noun are attached as daughters to the preposition thus being sisters to their head noun. We changed the annotation of PPs by identifying the head noun in the PP and attaching all of its siblings to it. To find the correct head, we used a heuristic in the style of Magerman (1995). The head is chosen by taking the rightmost daughter of the preposition that has a category label according to the heuristic and is labeled with NK (noun kernel element).

Table 1 shows the parser performance on the data after PP-restructuring.[4] The explanation for the benefit of the restructuring is of course that now there is only one type of NP in the whole corpus which eases the parser's task to correctly learn and identify them.

	dev. set		test set	
	LAS	UAS	LAS	UAS
hash kernel	87.40	89.79	88.06	90.24
+restructured	87.49	89.97	88.30	90.44

Table 1: Parser performance on restructured data

Since restructuring parts of the corpus seems beneficial, there might be other structures where more consistent annotation could help the parser, e. g., coordination or punctuation (like in the 2008 ACL Workshop data set, cp. Footnote 1).

5 Part-of-Speech Tagging

High quality part-of-speech (PoS) tags can greatly improve parsing quality. Having a verb wrongly analyzed as a noun and similar mistakes are very likely to mislead the parser in its decision process. A lot of the parser's features include PoS tags and reducing the amount of errors during PoS tagging will therefore reduce misleading feature values as well. Since the quality of the automatically assigned PoS tags in the German CoNLL '09 data is not state-of-the-art (see Table 2 below), we decided to retag the data with our own tagger which uses additional information from a symbolic morphological analyzer to direct a statistical classifier.

For the assignment of PoS tags, we apply a standard maximum entropy classification approach (see Ratnaparkhi (1996)). The classes of the classifier are the PoS categories defined in the *Stuttgart-Tübingen Tag Set (STTS)* (Schiller et al., 1999). We use standard binarized features like the word itself, its last three letters, whether the word is capitalized, contains a hyphen, a digit or whether it consists of digits only. As the only non-binary feature, word length is recorded. These standard features are augmented by a number of binary features that support the classification process by providing a preselection of possible PoS tags. Every word is analyzed by DMOR, a finite state morphological analyzer, from whose output analyses all different PoS tags are collected and added to the feature set. For example, DMOR assigns the PoS tags NN (common noun) and ADJD (predicative adjective) to the word *gegan-*

[4] Note that we are evaluating against a gold standard here (and in the rest of the paper) which has been restructured as well. With a different gold standard one could argue that the absolute figures we obtain are not fully comparable with the original CoNLL shared task. However, since we are doing dependency parsing, the transformation does neither add nor remove any nodes from the structure nor do we change any labels. The only thing that is done during the transformation is the reattachment of some daughters of a PP. This is only a small modification, and it is certainly linguistically warranted.

gen (gone). From these analyses two features are generated, namely *possible-tag:NN* and *possible-tag:ADJD*, which are strong indicators for the classifier that one of these classes is very likely to be the correct one. The main idea here is to use the morphological analyzer as a sort of lexicon that preselects the set of possible tags beforehand and then use the classifier to do the disambiguation (see Jurish (2003) for a more sophisticated system based on Hidden-Markov models that uses roughly the same idea). Since the PoS tags are included in the feature set, the classifier is still able to assign every class defined in *STTS* even if it is not in the preselection. Where the morphological analyzer does not know the word in question we add features for every PoS tag representing a productive word class in German, making the reasonable assumption that the morphology knows about all closed-class words and word forms. Finally, we add word form and possible tag features for the previous and the following word to the feature set thus simulating a trigram tagger. We used the method of Kazama and Tsujii (2005) which uses inequality constraints to do a very efficient feature selection[5] to train the maximum entropy model.

We annotated the entire corpus with versions of our own tagger, i.e., the training, development and test data. In order to achieve a realistic behavior (including remaining tagging errors, which the parser may be able to react to if they are systematic), it was important that each section was tagged without any knowledge of the gold standard tags. For the development and test portion, this is straightforward: we trained a model on the gold PoS of the training portion of the data and applied it to retag these two portions. Retagging the training portion was a bit trickier since we could not use a model trained on the same data, but at the same time, we wanted to use a tagger of similarly high quality – i.e. one that has seen a similar amount of training data. The training set was therefore split into 20 different parts and for every split, a tagging model was trained on the other 19 parts which then was used to retag the remaining 20th part. Table 2 shows the quality of our tagger evaluated on the German CoNLL '09 data in terms of accuracy and compares it to the originally annotated PoS tags which have been assigned by using the TreeTagger (Schmid, 1995) together with the German tagging model provided from the TreeTagger website. Tagging accuracy improves consistently by about 2 percentage points which equates to an error reduction of 44.55 % to 49.0 %.

	training	development	test
original	95.69	95.51	95.46
retagged	97.61	97.71	97.52
error red.	44.55%	49.00%	45.37%

Table 2: Tagging accuracy

Table 3 shows the parser performance when trained on the newly tagged data. The considerable improvements in tagging accuracy visibly affect parsing accuracy, raising both the labeled and the unlabeled attachment score by 0.66 percentage points (LAS) and 0.51 points (UAS) for the development set and by 0.45 points (LAS) and 0.64 points (UAS) for the test set.

	dev. set		test set	
	LAS	UAS	LAS	UAS
restructured	87.49	89.97	88.30	90.44
+retagged	88.15	90.48	88.75	91.08

Table 3: Parser performance on retagged data

6 Morphological Information

German, as opposed to English, exhibits a relatively rich morphology. Predicate arguments and nominal adjuncts are marked with special case morphology which allows for a less restricted word order in German. The German case system comprises four different case values, namely nominative, accusative, dative and genitive case. Subjects and nominal predicates are usually marked with nominative case, objects receive accusative or dative case and genitive case is usually used to mark possessors in possessive constructions. There are also some temporal and spatial nominal adjuncts which require certain case values. Since case is used to mark the function of a noun phrase in a clause, providing case information to a parser might improve its performance.

The morphological information in the German CoNLL '09 data contains much more information than case alone and previous models (baseline,

[5] We used a width factor of 1.0.

hash kernel, retagged) have used all of it. However, since we aim to improve a syntactic parser, we would like to exclude all morphological information from the parsing process that is not obviously relevant to syntax, e. g. mood or tense. By reducing the morphological annotations to those that are syntactically relevant, we hope to reduce the noise that is introduced by irrelevant information. (One might expect that machine learning and feature selection should "filter out" irrelevant features, but given the relative sparsity of unambiguous instances of the linguistically relevant effects, drawing the line based on just a few thousand sentences of positive evidence would be extremely hard even for a linguist.)

We annotated every case-bearing word in the corpus with its case information using DMOR. With case-bearing words, we mean nouns, proper nouns, attributive adjectives, determiners and all kinds of pronouns. Other types of morphological information was discarded. We did not use the manually annotated and disambiguated morphological information already present in the corpus for two reasons: the first one is the same as with the PoS tagging. Since it is unrealistic to have gold-standard annotation in a real-world application which deals with unseen data, we want the parser to learn from and hopefully adapt to imperfectly annotated data. The second reason is the German-inherent form syncretism in nominal paradigms. The German noun inflection system is with over ten different (productive and non-productive) inflectional patterns quite complicated, and to make matters worse, there are only five different morphological markers to distinguish 16 different positions in the pronoun, determiner and adjective paradigms and eight different positions in the noun paradigms. Some positions in the paradigm will therefore always be marked in the same way and we would like the parser to learn that some word forms will always be ambiguous with respect to their case value.

We also conducted experiments where we annotated number and gender values in addition to case. The idea behind this is that number and gender might help to further disambiguate case values. The downside of this is the increase in feature values. Combining case and number features means a multiplication of their values creating eight new feature values instead of four. Adding gender annotation raises this number to 24. Beside the disambiguation of case, there is also another reason why we might want to add number and gender: Inside a German noun phrase, all parts have to agree on their case and number feature in order to produce a well-formed noun phrase. Furthermore, the head noun governs the gender feature of the other parts. Thus, all three features can be relevant to the construction of a syntactic structure.[6] Table 4 shows the results of our experiments with morphological features.

	dev. set		test set	
	LAS	UAS	LAS	UAS
retagged	88.15	90.48	88.75	91.08
no morph.	87.78	90.18	88.60	90.92
+case	88.04	90.48	88.77	91.13
+c+n	88.21	90.62	88.88	91.13
+c+n+g	87.96	90.33	88.73	90.99

Table 4: Parser performance with morph. information (c=case, n=number, g=gender)

The *no morph* row in Table 4 shows, that using no morphological information at all decreases parser performance. When only case values are annotated, the parser performance does not change much in comparison to the retagged model, so there is no benefit here. Adding number features on the other hand improves parsing results significantly. This seems to support our intuition that number helps in disambiguating case values. However, adding gender information does not further increase this effect but hurts parser performance even more than case annotation alone. This leaves us with a puzzle here. Annotating case and number helps the parser, but case alone or having case, number and gender together affects performance negatively. A possible explanation might be that the effect of the gender information is masked by the increased number of feature values (24) which confuses the parsing algorithm.

7 Parser Stacking

Nivre and McDonald (2008) show how two different approaches to data-driven dependency pars-

[6] Person would be another syntactically relevant information. However, since we are dealing with a newspaper corpus, first and second person features appear very rarely.

ing, the graph-based and transition-based approaches, may be combined and subsequently learn to complement each other to achieve improved parsing results for different languages.

MaltParser (Nivre et al., 2006) is a language-independent system for data-driven dependency parsing which is freely available.[7] It is based on a deterministic parsing strategy in combination with treebank-induced classifiers for predicting parsing actions. MaltParser employs a rich feature representation in order to guide parsing. For the training of the Malt parser model that we use in the stacking experiments, we use learner and parser settings identical to the ones optimized for German in the CoNLL-X shared task (Nivre et al., 2006). Furthermore, we employ the technique of pseudo-projective parsing described in Nilsson and Nivre (2005) and a split prediction strategy for predicting parse transitions and arc labels (Nivre and Hall, 2008).[8] In order to obtain automatic parses for the whole data set, we perform a 10-fold split. For the parser stacking, we follow the approach of Nivre and McDonald (2008), using MaltParser as a guide for the MST parser with the hash kernel, i.e., providing the arcs and labels assigned by MaltParser as features. Table 5 shows the scores we obtain by parser stacking. Although our version of MaltParser does not quite have the same performance as for instance the version of Hall and Nivre (2008), its guidance leads to a small improvement in the overall parsing results.

	dev. set		test set	
	LAS	UAS	LAS	UAS
MaltParser	82.47	85.78	83.84	86.8
our parser	88.21	90.62	88.88	91.13
+stacking	88.42	90.77	89.28	91.40

Table 5: Stacked parser performance with guidance by MaltParser

[7] http://maltparser.org

[8] The feature models make use of information about the lexical form (FORM), the predicted PoS (PPOS) and the dependency relation constructed thus far during parsing (DEP). In addition, we make use of the predicted values for other morphological features (PFEATS). We employ the arc-eager algorithm (Nivre, 2003) in combination with SVM learners, using LIBSVM with a polynomial kernel.

8 Relabeling

In the relabeling step, we pursue the idea that some erroneous parser decisions concerning the distribution of certain labels might be detected and repaired in post-processing. In German and in most other languages, there are syntactic restrictions on the number of subjects and objects that a verb might select. The parser will learn this behavior during training. However, since it is using a statistical model with a limited context, it can still happen that two or more of the same grammatical functions are annotated for the same verb. But having two subjects annotated for a single verb makes this particular clause uninterpretable for subsequently applied tasks. Therefore, we would like to detect those doubly annotated grammatical functions and correct them in a controlled way.

The detection algorithm is simple: Running over the words of the output parse, we check for every word whether it has two or more daughters annotated with the same grammatical function and if we find one, we relabel all of its daughters.[9] For the relabeling, we applied a dependency-version of the function labeler described in Seeker et al. (2010) which uses a maximum entropy classifier that is restrained by a number of hard constraints implemented as an Integer Linear Program. These constraints model the aforementioned selectional restrictions on the number of certain types of verbal arguments. Since these are hard constraints, the labeler is not able to annotate more than one of those grammatical functions per verb. If we count the number of sentences that contain doubly annotated grammatical functions in the best parsing results from the previous section, we get 189 for the development set and 153 for the test set. About two thirds of the doubly annotated functions are subjects and the biggest part of the remaining third are accusative objects which are the most common arguments of German verbs.

Table 6 shows the final results after relabeling the output of the best performing parser configuration from the previous section. The improvements on the overall scores are quite small, which

[9] The grammatical functions we are looking for are SB (subject), OA (accusative object), DA (dative), OG (genitive object), OP (prepositional object), OC (clausal object), PD (predicate) and OA2 (second accusative object).

	dev. set		test set	
	LAS	UAS	LAS	UAS
stacking	88.42	90.77	89.28	91.40
+relabeling	88.48	90.77	89.40	91.40

Table 6: Parse quality after relabeling

is partly due to the fact that the relabeling affects only a small subset of all labels used in the data. Furthermore, the relabeling only takes place if a doubly annotated function is detected; and even if the relabeling is applied we have no guarantee that the labeler will assign the labels correctly (although we are guaranteed to not get double functions). Table 7 shows the differences in precision and recall for the grammatical functions between the original and the relabeled test set. As one can see, scores stay mostly the same except for SB, OA and DA. For OA, scores improve both in recall and precision. For DA, we trade a small decrease in precision for a huge improvement in recall and vice versa for SB, but on a much smaller scale. Generally spoken, relabeling is a local repair strategy that does not have so much effect on the overall score but can help to get some important labels correct even if the parser made the wrong decision. Note that the relabeler can only repair incorrect label decisions, it cannot help with wrongly attached words.

	original		relabeled	
	rec	prec	rec	prec
DA	64.2	83.2	74.7	79.6
OA	88.9	85.8	90.7	88.2
OA2	0.0	NaN	0.0	NaN
OC	95.2	93.5	95.1	93.7
OG	33.3	66.7	66.7	80.0
OP	54.2	80.8	54.2	79.9
PD	77.1	76.8	77.1	76.8
SB	91.0	90.6	90.7	93.7

Table 7: Improvements on grammatical functions in the relabeled test set

9 Conclusion

We presented a sequence of modifications to a data-driven dependency parser of German, departing from a state-of-the-art set-up in an implementation that allows for fast and robust training and application. Our pilot study tested what can be achieved in a few weeks if the data-driven technique is combined with a linguistically informed approach, i.e., testing hypotheses of what should be particularly effective in a very targeted way. Most modifications were relatively small, addressing very different dimensions in the system, such as the handling of features in the Machine Learning, the quality and combination of automatically assigned features and the ability to take into account global constraints, as well as the combination of different parsing strategies. Overall, labeled accuracy on a standard test set (from the CoNLL 2009 shared task), ignoring gold standard part-of-speech tags, increased significantly from 87.64% (baseline parser without hash kernel) to 89.40%.[10] We take this to indicate that a targeted and informed approach like the one we tested can have surprising effects even for a language that has received relatively intense consideration in the parsing literature.

Acknowledgements

We would like to thank Sandra Kübler, Yannick Versley and Yi Zhang for their support. This work was partially supported by research grants from the Deutsche Forschungsgemeinschaft as part of SFB 632 "Information Structure" at the University of Potsdam and SFB 732 "Incremental Specification in Context" at the University of Stuttgart.

References

Attardi, G. 2006. Experiments with a Multilanguage Non-Projective Dependency Parser. In *Proceedings of CoNLL*, pages 166–170.

Blum, A. 2006. Random Projection, Margins, Kernels, and Feature-Selection. In *LNCS*, pages 52–68. Springer.

Bohnet, B. 2009. Efficient Parsing of Syntactic and Semantic Dependency Structures. In *Proceedings of CoNLL 2009)*.

Brants, Sabine, Stefanie Dipper, Silvia Hansen, Wolfgang Lezius, and George Smith. 2002. The TIGER treebank. In *Proceedings of the Workshop on Treebanks and Linguistic Theories*, Sozopol.

Buchholz, Sabine and Erwin Marsi. 2006. CoNLL-X Shared Task on Multilingual Dependency Parsing. In *In Proc. of CoNLL*, pages 149–164.

Carreras, X. 2007. Experiments with a Higher-order Projective Dependency Parser. In *EMNLP/CoNLL*.

[10] $\alpha = 0.01$, measured with a tool by Dan Bikel from www.cis.upenn.edu/~dbikel/download/compare.pl

Crammer, K., O. Dekel, S. Shalev-Shwartz, and Y. Singer. 2006. Online Passive-Aggressive Algorithms. *Journal of Machine Learning Research*, 7:551–585.

Duchier, Denys and Ralph Debusmann. 2001. Topological dependency trees: a constraint-based account of linear precedence. In *Proceedings of ACL 2001*, pages 180–187, Morristown, NJ, USA. Association for Computational Linguistics.

Eisner, J. 1996. Three New Probabilistic Models for Dependency Parsing: An Exploration. In *Proceedings of Coling 1996*, pages 340–345, Copenhaen.

Eisner, J., 2000. *Bilexical Grammars and their Cubic-time Parsing Algorithms*, pages 29–62. Kluwer Academic Publishers.

Hajič, J., M. Ciaramita, R. Johansson, D. Kawahara, M. Antònia Martí, L. Màrquez, A. Meyers, J. Nivre, S. Padó, J. Štěpánek, P. Straňák, M. Surdeanu, N. Xue, and Y. Zhang. 2009. The CoNLL-2009 Shared Task: Syntactic and Semantic Dependencies in Multiple Languages. In *Proceedings of the 13th CoNLL-2009, June 4-5*, Boulder, Colorado, USA.

Hall, Johan and Joakim Nivre. 2008. A dependency-driven parser for German dependency and constituency representations. In *Proceedings of the Workshop on Parsing German*, pages 47–54, Columbus, Ohio, June. Association for Computational Linguistics.

Hinrichs, Erhard, Sandra Kübler, Karin Naumann, Heike Telljohann, and Julia Trushkina. 2004. Recent developments in linguistic annotations of the tüba-d/z treebank. In *Proceedings of the Third Workshop on Treebanks and Linguistic Theories*, pages 51–62, Tübingen, Germany.

Johansson, R. and P. Nugues. 2008. Dependency-based Syntactic–Semantic Analysis with PropBank and NomBank. In *Proceedings of the Shared Task Session of CoNLL-2008*, Manchester, UK.

Jurish, Bryan. 2003. A hybrid approach to part-of-speech tagging. Technical report, Berlin-Brandenburgische Akademie der Wissenschaften.

Kazama, Jun'Ichi and Jun'Ichi Tsujii. 2005. Maximum entropy models with inequality constraints: A case study on text categorization. *Machine Learning*, 60(1):159–194.

Kübler, Sandra. 2008. The PaGe 2008 shared task on parsing german. In *Proceedings of the Workshop on Parsing German*, pages 55–63, Columbus, Ohio, June. Association for Computational Linguistics.

Magerman, David M. 1995. Statistical decision-tree models for parsing. In *Proceedings of ACL 1995*, pages 276–283, Morristown, NJ, USA. Association for Computational Linguistics Morristown, NJ, USA.

McDonald, R. and F. Pereira. 2006. Online Learning of Approximate Dependency Parsing Algorithms. In *In Proc. of EACL*, pages 81–88.

Menzel, Wolfgang and Ingo Schröder. 1998. Decision procedures for dependency parsing using graded constraints. In *Proceedings of the COLING-ACL '98 Workshop on Processing of Dependency-Based Grammars*, pages 78–87.

Nilsson, Jens and Joakim Nivre. 2005. Pseudo-projective dependency parsing. In *Proceedings of ACL 2005*, pages 99–106.

Nivre, Joakim and Johan Hall. 2008. A dependency-driven parser for German dependency and constituency representations. In *Proceedings of the ACL Workshop on Parsing German*.

Nivre, J. and R. McDonald. 2008. Integrating Graph-Based and Transition-Based Dependency Parsers. In *ACL-08*, pages 950–958, Columbus, Ohio.

Nivre, Joakim, Jens Nilsson, Johan Hall, Gülşen Eryiğit, and Svetoslav Marinov. 2006. Labeled pseudo-projective dependency parsing with Support Vector Machines. In *Proceedings of CoNLL 2006*.

Nivre, J. 2003. An Efficient Algorithm for Projective Dependency Parsing. In *8th International Workshop on Parsing Technologies*, pages 149–160, Nancy, France.

Nivre, J. 2009. Non-Projective Dependency Parsing in Expected Linear Time. In *Proceedings of the 47th Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP*, pages 351–359, Suntec, Singapore.

Rahimi, A. and B. Recht. 2008. Random Features for Large-Scale Kernel Machines. In Platt, J.C., D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems*, volume 20. MIT Press, Cambridge, MA.

Ratnaparkhi, Adwait. 1996. A maximum entropy model for part-of-speech tagging. In *Proceedings of EMNLP 1996*, volume 1, pages 133–142.

Schiller, Anne, Simone Teufel, and Christine Stöckert. 1999. Guidelines für das Tagging deutscher Textcorpora mit STTS (Kleines und großes Tagset). Technical Report August, Universität Stuttgart.

Schmid, Helmut. 1995. Improvements in part-of-speech tagging with an application to German. In *Proceedings of the ACL SIGDAT-Workshop*, volume 11.

Seeker, Wolfgang, Ines Rehbein, Jonas Kuhn, and Josef Van Genabith. 2010. Hard Constraints for Grammatical Function Labelling. In *Proceedings of ACL 2010*, Uppsala.

Titov, I. and J. Henderson. 2007. A Latent Variable Model for Generative Dependency Parsing. In *Proceedings of IWPT*, pages 144–155.

Yamada, H. and Y. Matsumoto. 2003. Statistical Dependency Analysis with Support Vector Machines. In *Proceedings of IWPT*, pages 195–206.

Using Clustering to Improve Retrieval Evaluation without Relevance Judgments

Zhiwei Shi
Institute of Computing Technology
Chinese Academy of Science
shizhiwei@ict.ac.cn

Peng Li
Institute of Computing Technology
Chinese Academy of Science
lipeng01@ict.ac.cn

Bin Wang
Institute of Computing Technology
Chinese Academy of Science
wangbin@ict.ac.cn

Abstract

Retrieval evaluation without relevance judgments is a hard but also very meaningful work. In this paper, we use clustering technique to improve the performance of judgment free retrieval evaluation. By using one system to represent all the systems that are similar to it, we can largely reduce the negative effect of similar retrieval results in Retrieval evaluation. Experimental results demonstrated that our method outperformed all the previous judgment free evaluation methods significantly. Its overall average performance outperformed the best previous result by 20.5%. Besides, our work is a general framework that can be applied to any other judgment free evaluation method for performance improvement.

1 Introduction

Generally, to compare the effectiveness of information retrieval systems, we need to prepare a test collection composed of a set of documents, a set of query topics, and a set of relevance judgments indicating which documents are relevant to which topics. Among these requirements, relevance judgment is the most human resource exhausting and time consuming part. It even becomes infeasible when the test collection is extremely large. To address this problem, the TREC conferences used a pooling technology (Voorhees and Harman, 1999), where the top *n* (e.g., *n*=100) documents retrieved by each participating system are collected into a pool and then only the documents in the pool are judged for system comparison. Zobel (1998) has shown that this pooling method leads to reliable results in term of determining the effectiveness of retrieval systems and their relative rankings. Yet, the relevance determination process is still very resource intensive especially when the test collection reaches or exceeds terabyte, or much more queries are included. More seriously, when we change to a new document collection, we have to redo the entire evaluation process.

There are two possible solutions to the problem above, evaluation with incomplete relevance judgments and evaluation without relevance judgments. The former is well studied. Many well designed ranking methods with incomplete judgments were carried out. Two of them, Minimal Test Collection (MTC) method (Carterette et al., 2006) and Statistical evaluation (statMAP) method (Aslam et al., 2006), even got practical application in the Million Query (1MQ) track in TREC 2007 (Allan et al., 2007), and achieved satisfactory evaluation performance. The latter is comparatively less studied. Only a few papers concentrate on the issue of evaluating retrieval systems without relevance judgments. In Section 2 of this paper, we will briefly review some representative methods. We will see what they are and how they work.

Coling 2008: Poster Volume, pages 1131–1139,
Beijing, August 2010

In this paper, we focus our effort on the retrieval evaluation without relevance judgments. Although 'blind' evaluation is really a hard problem and its evaluation performance is far less than that of methods with incomplete judgments, it is undeniable that non-judgment evaluation has its own advantages. In some cases, relevance judgments are non-attainable. For example, when researchers compare their novel retrieval algorithms to existing methods, or search for optimal parameters of their algorithms, or conduct data fusion in a dynamic environment, relevance judgment usually seems impossible. Besides, to construct a good evaluation method without relevance judgments, researchers need to mine the retrieval results thoroughly, and try to find laws that indicate the correlation between the effectiveness of a system and features of its retrieval result. These laws are not only useful for 'blind' evaluation methods but also valuable for evaluation methods with incomplete judgments.

One of the useful laws for 'blind' evaluation methods is Authority Effect (Spoerri, 2005). Yet it always ruined by multiple similar results.

In this work, we use clustering technique to solve this problem. By selecting one system to represent all the systems that are similar to it, we can largely reduce the negative effect of similar retrieval results. Details of this method will be presented Section 3. Experimental results, which are reported in Section 4, also verified that our idea is feasible and effective. Our method outperformed all the previous judgment free evaluation methods on every test bed. The overall average performance outperformed the best previous result by 20.5%. Finally, we conclude our work in Section 5.

2 Related Work

In 2001, Soboroff et al. (2001) firstly proposed the concept of evaluating retrieval systems in the absence of relevance judgments. They generated a set of pseudo-relevance judgments by randomly selecting and declaring some documents from the pool of top 100 documents as relevant. This set of pseudo-relevance judgments (instead of a set of human relevance judgments) was then used to determine the effectiveness of the retrieval systems. Four versions of this random pseudo-relevance method were designed and tested on data from the ad hoc track in TREC 3, 5, 6, 7 and 8. They were simple random pseudo-relevance method, the variant with duplicate documents, the variant with Shallow pools and the variant with Exact-fraction sampling. All their resulting system assessments and rankings were well correlated with actual TREC rankings, and the variant with duplicate documents in pools got the best performance, with an average Kendall's tau value 0.50 over the data of TREC 3, 5, 6, 7 and 8.

Soboroff et al.'s idea came from two results in retrieval evaluation. One is that incomplete judgments do not harm evaluation results greatly. Zobel's (1998) research had showed that the results obtained using pooling technology were quite reliable given a pool depth of 100. He also found that even though the pool depth was limited to 10, the relative performance among systems changed little, although actual precision scores did change for some systems. The other is that partially incorrect relevance judgments do not harm evaluation results greatly. Voorhees (1998) ascertained that despite a low average overlap between assessment sets, and wide variation in overlap among particular topics, the relative rankings of systems remained largely unchanged across the different sets of relevance judgments. These two points are bases of Soboroff et al.'s random pseudo-relevance method, and give explanation to the result that their rankings were positively related to that of the actual TRECs. As a matter of fact, the two points are bases of all the retrieval evaluation methods without or with incomplete relevance judgments.

Aslam and Savell (2003) devised a method to measure the relative retrieval effectiveness of systems through system similarity computation. In their work, the similarity between two retrieval systems was the ratio of the number of documents in their intersection and union. Each system was scored by the average similarity between it and all other systems. This measurement produced results that were highly correlated with the random pseudo-relevance method. Aslam and Savell hypothesized that this was caused by 'tyranny of the masses' effect, and these two related methods were assessing the systems based on 'popularity' instead of 'performance'. The analysis by Spoerri (2005) sug-

gested that the 'popularity' effect was caused by considering all the runs submitted by a retrieval system, instead of only selecting one run per system. Our later experimental results will show that this point of view is partially correct. The 'popularity' effect could not be avoided completely by only selecting one run per system. This is indeed a hard problem for all the evaluation methods without relevance judgments.

Wu and Crestani (2003) developed multiple 'reference count' based methods to rank retrieval systems. They made the distinction between an 'original' document and its duplicates in all other lists, called the 'reference' documents, when computing a document's score. A system's score is the (weighted) sum of the scores of its 'original' documents. Several versions of reference count method were carried out and tested. The basic method (Basic) scored each 'original' document by the number of its 'reference' documents. The first variant (V1) assigned different weights to 'reference' documents based on their ranking positions. The second variant (V2) assigned different weights to the 'original' document based on its ranking position. The third variant (V3) assigned different weights to both the 'original' documents and the 'reference' documents based on their ranking positions. The fourth variant (V4) was similar to V3, except that it normalized the weights to 'reference' documents. Wu and Crestani's method output similar evaluation performance to that of the random pseudo-relevance method. Their work also showed that the similarity between the multiple runs submitted by the same retrieval system affected the ranking process. If only one run was selected for any of the participant system for any query, for 3-9 systems, V3 outperformed random pseudo-relevance method by 45.6%; for 10-15 systems, random pseudo-relevance method outperformed V3 by 6.5%.

Nuray and Can (2006) introduced a method to rank retrieval systems automatically using data fusion. Their method consists of two parts. One is selecting systems for data fusion, and the other is selecting documents as pseudo relevant documents as the fusion result. In the former part, they hypothesized that systems returning documents different from the majority could provide better discrimination among the documents and systems. In return, this could lead to a more accurate pseudo relevant documents and more accurate rankings. To find proper systems, they introduced the 'bias' concept for system selection. In their work, bias was 1 minus the similarity between a system and the majority, where the similarity is a normalized dot product of two vectors. In the latter part, Nuray and Can tested three criterions, namely Rank position, Borda count and Condorcet. Experimental results on data from TREC 3, 5, 6 and 7 showed that bias plus Condorcet got the best evaluation results and it outperformed the reference count method and random pseudo relevance method greatly.

More recently, Spoerri (2007) proposed a method using the structure of overlap between search results to rank retrieval systems. This method provides us a new view on how to rank retrieval systems without relevance judgments. He used local statistics of retrieval results as indicators of relative effectiveness of retrieval systems. Concretely, if there are N systems to be ranked, N groups are constructed randomly with the constraint that each group contains five systems and each system will appear in five groups; then the percentages of a system's documents not found by other systems (Single%) as well as the difference between the percentages of documents found by a single system and all five systems (Single%-AllFive%) are calculated as indicators of relative effectiveness respectively. Spoerri found that these two local statistics were highly and negatively correlated with the mean average precision and precision at 1000 scores of the systems. By utilizing the two statistics to rank systems from subsets of TREC 3, 6, 7 and 8, Spoerri obtained appealing evaluation results. The overlap structure of the top 50 documents were sufficient to rank retrieval systems and produced the best results, which outperformed previous attempts to rank retrieval systems without relevance judgments significantly.

So far, we have reviewed 5 representatives of non-judgment evaluation methods. All these methods faced the same serious problem: similar runs harmed the effectiveness of ranking process. Different methods handled this problem differently. Aslam and Savell (2003) called this the 'tyranny of the masses' and provided no solution. Wu and Crestani (2003) addressed this problem by selecting only one run for any of the participant system for any query. Nuray and Can (2006) selected systems that were less simi-

lar to the majority for data fusion. Spoerri (2007) performed his method on a selected subset of all the systems. All these treatments led to evaluation performance improvement. Yet we will say it could be improved more. In the next section, we will present a new solution to this problem. Its performance is examined in Section 4.

3 Using Clustering to Improve Retrieval Evaluation without Relevance Judgments

3.1 Problem

As we reviewed in Section 2, previous research had shown that incomplete relevance judgments and partially incorrect relevance judgments do not harm retrieval evaluation greatly. This is why pooling technique can lead to reliable retrieval evaluation results. It is also the theoretical foundation of evaluation without relevance judgments.

Besides, non-judgments methods armed with more laws inside retrieval results. These laws indicate the correlation between retrieval effectiveness of a system and features in its retrieval results. One of the most important laws used in non-judgments evaluation is Authority Effect (Spoerri, 2005): document, which is retrieval by more systems, is more likely being relevant. Unfortunately, similar retrieval results ruined this law. Aslam and Savell (2003) called this the 'tyranny of the masses'. So, how to alleviate the negative effect of similar retrieval results is a big issue in non-judgments evaluation.

3.2 Solution

Generally, our solution to the 'tyranny of the masses' is removing similar systems by clustering. The whole process is as follows:

Firstly, all systems to be evaluated are clustered into several subsets.

Secondly, for each subset, one system is selected as a representative.

Thirdly, all the information used for system evaluation comes from these representatives.

Finally, score every system according to the information collected in the previous step.

This is the general framework of our methodology. Notice that, in the third step, only selected systems contribute to the information required for system evaluation. So we can eliminate the negative effect caused by similar retrieval results.

This solution can be applied to any method of retrieval evaluation without relevance judgments. To illustrate how to apply it to a retrieval evaluation method, we will describe using clustering to improve Average System Similarity, which is proposed by Aslam and Savell (2003), in detail as an example.

3.3 Average System Similarity Based on Clustering

In Aslam and Savell's (2003) method, each system is evaluated based on a criterion named Average System Similarity. The average system similarity of a given system S_0 is calculated according to formula (1).

$$\mathrm{AvgSysSim}(S_0) = \frac{1}{n-1}\sum_{S \neq S_0} SysSim(S, S_0) \quad (1)$$

where n is the number of systems to be evaluated, and similarity between two systems S and S_0, SysSim(S, S_0), is calculated based on formula (2).

$$\mathrm{SysSim}(S_1, S_2) = \frac{\left|\mathrm{Ret}_1 \cap \mathrm{Ret}_2\right|}{\left|\mathrm{Ret}_1 \cup \mathrm{Ret}_2\right|} \quad (2)$$

where Ret_i indicates the set of documents returned by System i (i = 1, 2).

When applying clustering technique to the system similarity method, we need to define an equivalence relation first.

Definition 1 (**System Equivalence**): Suppose that all systems are clustered into m clusters namely C_1, C_2, …, C_m. Two systems S_1 and S_2 are equivalent if and only if there exists k ($1 \le k \le m$) so that $S_1 \in C_k$ and $S_2 \in C_k$.

$$S_1 = S_2 \quad iff \quad \exists k, 1 \le k \le m, S_1 \in C_k, S_2 \in C_k \quad (3)$$

Given the definition of System Equivalence, we get the average system similarity based on clustering as follows:

$$\mathrm{AvgSysSim}(S_0) = \frac{1}{m-1}\sum_{R \neq S_0} SysSim(R, S_0) \quad (4)$$

where m is the number of clusters and R is the representative system of a cluster.

Replacing formula (1) with formula (4), we get the retrieval evaluation method Average System Similarity Based on Clustering, shortly ASSBC.

There are two important issues for ASSBC that need to be addressed. Issue 1: How to select representative system from a cluster? Issue 2: How to decide the number of clusters we need?

Before we address Issue 1, we introduce another definition, Cluster Similarity.

Definition 2 (**Cluster Similarity**): for any given two clusters C_1 and C_2, with their respective representative systems S_1 and S_2, the cluster similarity between C_1 and C_2 is the system similarity between S_1 and S_2.

$$\text{ClusterSim}(C_1, C_2) = \text{SysSim}(S_1, S_2) \quad (5)$$

Now we come to selecting representative systems for clusters. Here, we utilize a hierarchical bottom up clustering technique. The entire clustering process is as follows.

Initially, each system forms a cluster.
Loop Until the number of clusters is m
Two most similar clusters merge, and one of their representatives with higher average system similarity survives as the representative of the new cluster.
End Loop.

In the initial step, since every cluster contains only one system, the representative system is unquestionable. Within each loop, two representative systems of the old clusters are candidates of the new cluster, and the one with higher score, which means higher retrieval performance, becomes the representative of the new cluster.

For Issue 2, technically, how to decide the number of clusters is always a problem for clustering. Yet, we do not have to rush in the decision. Let us examine the evaluation performance on different values of *m* first.

4 Experiments

In this section, we will illustrate the evaluation performance of Average System Similarity Based on Clustering vs. different values of *m*. Before we come to the experimental results, we would like to make some details clear first.

4.1 Some Clarification

4.1.1 Dataset

We perform our experiments on the ad hoc tasks of TREC-3, -5, -6 and -7. Most existing works on retrieval evaluation without judgments are tested on these tasks. To make a direct comparison with these work mentioned in Section 2 later, we also choose these tasks as our test bed.

4.1.2 Performance Measurement

One of the measures of retrieval effectiveness used by TREC is mean non-interpolated average precision (MAP). Since average precision is based on much more information than other effectiveness measures such as R-precision or P(10) and known to be a more powerful and more stable effectiveness measure (Buckley and Voorhees, 2000), we utilize MAP as the effective measurement of retrieval systems in our experiments.

The correlation of the ranking with our proposed methods, as well as other methods, to the TREC official rankings is measured using the Spearman's rank correlation coefficient. One reason is that it suits better for evaluating correlation between ratio sequences, e.g. MAP, than Kendall's tau. The other reason is that we can directly compare our results with those of previous attempts reviewed in Section 2, since most of them provided Spearman's rank correlation coefficient results.

4.1.3 Substitute for Number of Clusters

TREC	Runs
3	40
5	61
6	74
7	103

Table 1. Number of TREC runs

As we know, the number of systems (runs) varies in different TREC dataset (see Table 1 for details). Instead of examining the evaluation performance variation when absolute number of clusters *m* changes, we illustrate the evaluation performance vs. the percentage of *m*. Actually, for the sake of convenience, we will plot the correlation of our method to the TREC official rankings vs. the percentage of systems removed from the representative group in the following subsection.

4.2 Experimental results

Figure 1-4 show the plots of the correlation of our method to the TREC official rankings vs. the percentage of systems removed from the representative group on TREC-3, -5, -6 and -7 respectively. The percentage of systems removed goes from 0 to 85%, where 0 means no system removed and represents the original Average System Similarity method, and 85% is an up bound in our experiments. The horizontal line indicates the original performance. The tagged number on the curve says when the performance curve reaches its peak and the peak value.

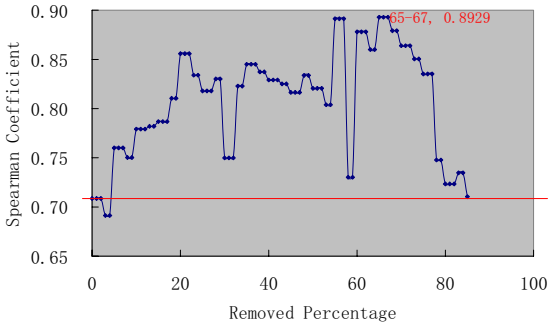


Figure 1. Spearman Coefficient of ASSBC vs. different percentage of removed systems on TREC -3.

In Figure 1, the Spearman coefficients of ASSBC vs. different percentage of removed systems on TREC-3 are presented. Except for the beginning, almost all the points are above the horizontal line. The curve reaches its top at 65%-67%, where the Spearman coefficient is 0.8929.

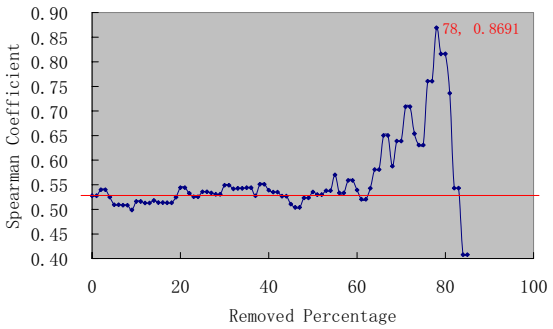


Figure 2. Spearman Coefficient of ASSBC vs. different percentage of removed systems on TREC -5.

Figure 2 depicts the evaluation performance on TREC-5. From 0 to 63%, the performance curve fluctuates around the horizontal line. This means deficient clustering does not bring substantial performance variation. After 63%, the curve begins to rise and reaches its peak at 78%, where the performance is 0.8691. Then it drops dramatically as more systems removed from the representative group.

The situation on TREC-6 is plotted in Figure 3. In this case, the curve rises gently in the interval between 0 and 70% except for some fluctuation. After 70%, the curve starts to climb and reaches the peak at 75% with the peak value of 0.8576. It remains high performance until 80%, and then decline quickly.

Figure 3. Spearman Coefficient of ASSBC vs. different percentage of removed systems on TREC -6.

Figure 4 presents the evaluation performance on TREC-7. The trend in this figure is pretty much like that in Figure 2. The curve fluctuates first, and then climbs the hill, where the peak value is 0.6557 and 75% systems are removed. The only difference is in this figure the curve is gentler. This means on TREC-7 ASSBC does not obtain as much improvement as on TREC-5.

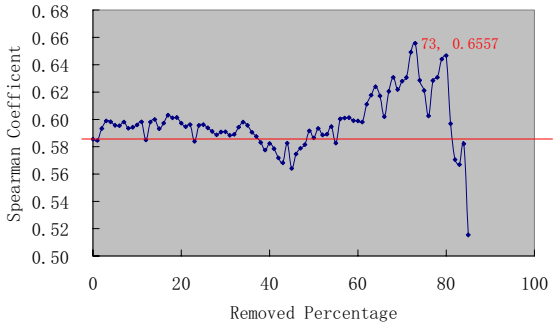


Figure 4. Spearman Coefficient of ASSBC vs. different percentage of removed systems on TREC -7.

According Figure1-4, we can say that clustering systems does bring us evaluation performance improvement. Generally, obvious improvement occurs in the interval between 65% and 80%. TREC-3 is an exception. The curve on TREC-3 reaches its peak at 65%. Notice that in TREC-3 there are only 40 systems (runs), and

65% indicates 26 systems removed and 14 systems left as representatives. Interestingly, for other TRECs, 78% (the biggest peak position) means at least 14 systems left as well. So, this can be interpreted as the minimum number of clusters.

To examine the general effect on evaluation performance of cluster number, we also plot the average performance of TREC -3, -5, -6 and -7 vs. the percentage of systems removed from the representative group in Figure 5. With slight fluctuation, the average performance curve climbs stably, and reaches its peak 0.7754 at the position 78%. Then it drops dramatically.

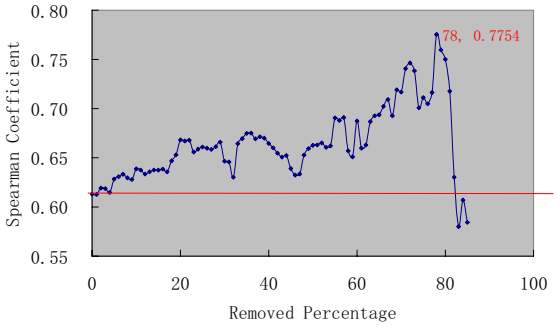


Figure 5. Average Spearman Coefficient of ASSBC vs. different percentage of removed systems on TREC -3, -5, -6 and -7.

To make the result more intuitive, we present a comparison of the performance of original Average System Similarity (ASS) and the best performance of Average System Similarity Based on Clustering (ASSBC) in Table 2. According to the table, we can see that clustering systems improve the evaluation performance significantly.

	ASS	ASSBC	Improvement
Trec3	0.7086	0.8929	26.0%
Trec5	0.5277	0.8691	64.7%
Trec6	0.6300	0.8576	36.1%
Trec7	0.5855	0.6557	12.0%
Avg	0.6129	0.7754	26.5%

Table 2. Spearman coefficients of original Average System Similarity (ASS) and the best performance of Average System Similarity Based on Clustering (ASSBC) on TREC -3, -5, -6, -7 and the over all average.

4.3 Comparison with All Previous Attempts

Meanwhile, we also provide a comparison among the ASSBC method and all the existing non-judgment evaluation methods mentioned in Section 2. The result is given in Table 3.

	RS	RC	CB	Single%	ASS	ASSBC optimal (78% Removed)
Trec3	0.627	0.587	0.867	0.824	0.709	0.893
Trec5	0.429	0.421	0.657*	0.563	0.528	0.869
Trec6	0.436	0.384	0.717	0.618	0.630	0.854
Trec7	0.411	0.382	0.453	0.550	0.585	0.631
Avg	0.476	0.444	0.674	0.639	0.613	0.812

Table 3. Spearman coefficients for best results from different evaluation methods

In Table 3, RS represents the result of random pseudo relevance method, where relevance ratio is set to 10% rather than the actual ratio in its original version; RC is the best result produced by reference count method; BC accounts for the best result of Bias plus Condorcet method, a data fusion based method. Results of these three methods are cited from Nuray and Can's (2006) paper. For the number with a '*' (BC on TREC 5), in their original paper, same result in different tables conflict, and we pick the higher value presenting in Table 3. Single% is the representative of Spoerri's overlap structure based method. Different from its original version, the result in Table 3 is gained on all the systems opposite to on a selected subset, except that runs submitted by the same system are counted only once. ASS is short for Average System Similarity. ASSBC optimal is the best result of our method. Here we utilize both 78%

as the percentage of removed systems and 14 as the minimum number of clusters[1]. Clearly, our method outperforms all the previous attempts on every TREC. The overall average performance outperforms the best previous result (from CB) by 20.5%.

5 Conclusion

Retrieval evaluation without relevance judgments is a hard problem. Meanwhile it is also an important problem that we can not avoid it in many research areas and applications.

One of the main factors that depress the performance of judgments free evaluation is: similar retrieval results ruined the Authority Effect, which is one of the important bases for all the judgment free evaluation methods.

In this paper, we use clustering technique to address this problem. By using one system to represent all the systems that are similar to it, we can largely reduce the negative effect of similar retrieval results. Experimental results also verified our idea. Our method outperforms all the previous judgment free evaluation methods on every test bed. The overall average performance outperforms the best previous result by 20.5%.

Besides, improving judgment free evaluation via clustering is more than just a method. It is a general framework that can be applied to any judgment free evaluation method. The Average System Similarity Based on Clustering method is an example. It works well means that the framework is feasible and successful. We will apply it to other judgment free evaluation methods in our future work.

Acknowledgement This work is supported by the National Science Foundation of China under Grant No. 60776797, the Major State Basic Research Project of China (973 Program) under Grant No. 2007CB311103 and the National High Technology Research and Development Program of China (863 Program) under Grant No. 2006AA010105.

[1] Since we add a terminal criterion for clustering with 14 as the minimum number of clusters, the average performance in Table 3 gains an improvement compared to that presented in Figure 5 and Table 2.

References

Allan J., Carterette B., Aslam J. A., Pavlu V., Dachev B., and Kanoulas E. 2007 Overview of the TREC 2007 Million Query Track, Proceedings of TREC.

Aslam J. A., Pavlu V. and Yilmaz E. 2006 A statistical method for system evaluation using incomplete judgments, Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval, August 06-11, 2006, Seattle, Washington

Aslam J. A. and Savell R. 2003 On the effectiveness of evaluating retrieval systems in the absence of relevance judgments, Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval, July 28-August 01, 2003, Toronto, Canada

Buckley, C. and Voorhees, E. M. 2000 Evaluating evaluation measure stability, Proceedings of the 23rd ACMSIGIR conference pp. 33 – 40

Carterette B., Allan J. and Sitaraman R. 2006 Minimal test collections for retrieval evaluation, Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval, August 06-11, 2006, Seattle, Washington, USA

Nuray R. and Can F. 2006 Automatic ranking of information retrieval systems using data fusion, Information Processing and Management: an International Journal, v.42 n.3, p.595-614, May 2006

Soboroff I., Nicholas C. and Cahan P. 2001 Ranking retrieval systems without relevance judgments, Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval, p.66-73, September 2001, New Orleans, Louisiana, United States

Spoerri A. 2005 How the overlap between search results correlates with relevance. In: Proceedings of the 68th annual meeting of the American Society for Information Science and Technology (ASIST 2005).

Spoerri A. 2007 Using the structure of overlap between search results to rank retrieval systems without relevance judgments, Information Processing and Management: an International Journal, v.43 n.4, pp.1059-1070, July, 2007

Voorhees E. M. 1998 Variations in relevance judgments and the measurement of retrieval effectiveness, Proceedings of the 21st annual international ACM SIGIR conference on Research and devel-

opment in information retrieval, p.315-323, August 24-28, 1998, Melbourne, Australia

Voorhees E. M. and Harman, D. 1999 Overview of the eighth text retrieval conference (TREC-8). The eighth text retrieval conference (TREC-8), Gaithersburg, MD, USA, 1999. U.S. Government Printing Office, Washington

Wu S. and Crestani F. 2003 Methods for ranking information retrieval systems without relevance judgments, Proceedings of the 2003 ACM symposium on Applied computing, March 09-12, 2003, Melbourne, Florida

Zobel J. 1998 How reliable are the results of large-scale information retrieval experiments?, Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval, p.307-314, August 24-28, 1998, Melbourne, Australia

A Method for Automatically Generating a Mediatory Summary to Verify Credibility of Information on the Web

Hideyuki Shibuki and **Takahiro Nagai** and **Masahiro Nakano**
Rintaro Miyazaki and **Madoka Ishioroshi** and **Tatsunori Mori**
Graduate School of Environment and Information Sciences,
Yokohama National University
{shib, nagadon, nakano, rintaro, ishioroshi, mori}@forest.eis.ynu.ac.jp

Abstract

In this paper, we propose a method for mediatory summarization, which is a novel technique for facilitating users' assessments of the credibility of information on the Web. A mediatory summary is generated by extracting a passage from Web documents; this summary is generated on the basis of its relevance to a given query, fairness, and density of keywords, which are features of the summaries constructed to determine the credibility of information on the Web. We demonstrate the effectiveness of the generated mediatory summary in comparison with the summaries of Web documents produced by Web search engines.

1 Introduction

Many pages on the Web contain incorrect or unverifiable information. Therefore, there is a growing demand for technologies that can enable us to obtain reliable information. However, it would be almost impossible to automatically ascertain the accuracy of information presented on the Web. Hence, the second-best approach is the development of a supporting method for judging the credibility of information on the Web.

Presently, when we wish to judge the credibility of information on the Web, we often read some relevant Web documents retrieved via Web search engines. However, Web search engines do not provide any suggestions in cases where the content of some documents conflicts with the content of other documents. Furthermore, the retrieved documents are too many to read and may not be ranked according to the credibility of the information they provide. In other words, information retrieval is not sufficient to support users' assessments of the credibility of information, and therefore, additional techniques are required for the same.

Several previous researches have been conducted for developing such techniques. Juffinger et al. (2009) ranked blogs in terms of their concurrence with well-verified information from sources such as a news corpus. Miyazaki et al. (2009) devised a method for extracting the description of the information sender, namely, the person or organization providing texts in Web pages. Ohshima et al. (2009) proposed a method for reranking Web pages according to users' regionality, which depends on two factors: uniformity and proximity. While the abovementioned studies mainly facilitate users' assessments of the credibility of individual Web pages, the following studies deal with the issues of credibility of information in multiple documents on the Web. Murakami et al. (2009) proposed a method to analyze semantic relationships such as agreement, conflict, or evidence between texts on the Web. Kawahara et al. (2009) reported a method for presenting overviews of evaluative information such as positive/negative opinions.

Although the above techniques facilitate users' assessments of credibility of information on the Web, there is still room for methods that support users' judgment. For example, when the truth of a statement[1] "Diesel engines are harmful to the environment" is to

[1] In this paper, a *statement* is defined as text such as an opinion, evaluation, or objective fact.

Coling 2008: Poster Volume, pages 1140–1148,
Beijing, August 2010

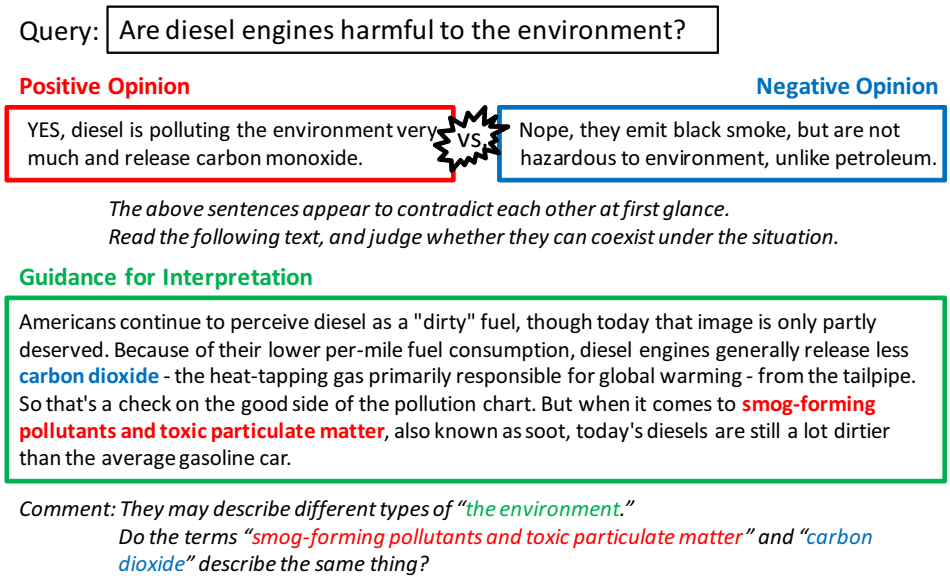


Figure 1: An example of the mediatory summary.

be verified, the following two contradictory groups of Web documents are obtained: one stating that "Diesel engines are harmful to the environment" and the other stating that "Diesel engines are not harmful to the environment." How does one resolve this conflict? Are the contents of one group that contains a less reasonable description wrong? On the other hand, if contents of both these groups are correct, then why do they appear to be contradictory to each other? A display of only the overview of statements in Web documents and the relationships between these statements does not always provide sufficient information to answer these questions.

In order to direct users to a reasonable interpretation written by one author, Kaneko et al. (2009) proposed the notion of a mediatory summary for *pseudo conflicts*, which are relationships between statements that appear to contradict each other at first glance but can coexist under a certain situation. However, Kaneko et al. did not describe any algorithms for automatically generating the mediatory summary. Therefore, in this paper, we propose a method for automatically generating the mediatory summary and demonstrate the effectiveness of generated mediatory summaries.

The rest of this paper is organized as follows. In Section 2, we describe the concept of a mediatory summary and the features required for automatically generating it. In Section 3, we describe an algorithm for generation of the mediatory summary. In Section 4, we present experimental results to demonstrate the effectiveness of the generated mediatory summary. Section 5 provides the conclusion.

2 Mediatory Summary

The generation of the mediatory summary is a type of informative summarization based on passage extraction. Figure 1 shows an example of the ideal mediatory summary for the query "Are diesel engines harmful to the environment?" The text in boxes with thick lines is extracted from Web documents, and the italicized text is generated through templates. The user is shown both positive and negative responses to the query and appropriately guided on how to interpret them. One of the most difficult issues in the automated generation of the mediatory summary is the extraction of the most suitable passage that is used for interpretation.[2]

[2]Owing to space limitations, we have omitted the discussion on generation through templetes.

Nakano et al. (2010) constructed a text summarization corpus for determining the credibility of information on the Web. Their corpus contains six query statements, the Web document collections retrieved for each query, and 24 summaries made by four persons per query. We analyzed these summaries from the viewpoint of mediatory summary generation and observed that mediatory summaries usually display the following three features.

The first feature is the high relevance to a given query. We can approximately determine whether the text displays this feature by examining whether or not it contains content words in the query such as "diesel," as in Figure 1. The second feature is fairness, or in other words, evenly describing both positive and negative opinions. We can approximately determine whether the text displays this feature by examining whether or not it contains words for both opinions and having different, typically opposite meanings such as "lot" and "less," as in Figure 1. It should be noted that words with opposite meanings are not limited to antonyms. In the case of the query "Are diesel engines harmful to the environment?," "carbon dioxide" and "smog-forming pollutants" should be also regarded as words with opposite meanings. The third feature is the high density of words of the above two types in a text. In addition to appropriateness of a text with high density as a summary, such a text is likely to be a text contrasting both sides of contents.

3 Proposed Method

3.1 Outline

We propose a method for generating mediatory summaries by extracting passages that display the features described in the previous section. First, we define the content words related to the topic of a query as *topic keywords*.[3] Next, we define the content words corresponding to the positive and negative opinions as *positive keywords* and *negative keywords*, respectively. Although topic keywords appear in the given query, positive/negative keywords hardly appear in the given query. Therefore, positive/negative keywords have to be extracted from other text.

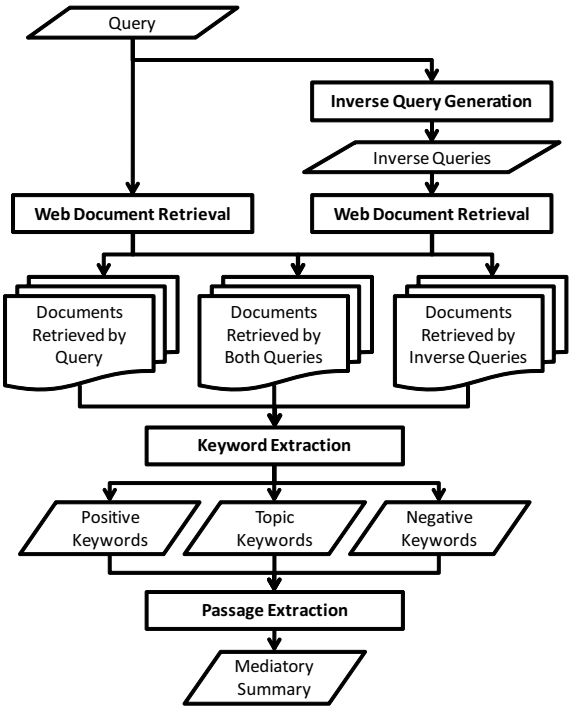


Figure 2: Outline of the proposed method.

Figure 2 shows the outline of the proposed method. First, in order to find contents opposed to a given query, *inverse queries* are generated. We define an inverse query as a query generated by replacing a word in the given query with its antonym. Next, the given query and the inverse queries are used to retain three sets of Web documents, which are likely to contain more positive keywords, negative keywords, and neither. The topic, positive, and negative keywords are then extracted from these sets. Finally, the extracted keywords are used to extract passages from the three sets of Web documents; these passages are ranked in order of the score described in Section 3.5 as a mediatory summary.

[3] Although topic words are not restricted to words in the given query, we use only the words in the given query as topic words in this paper. Inclusion of words other than those in the given query is a topic for future research.

3.2 Inverse Query Generation

Inverse queries are generated by simply replacing a word in the given query with an antonym of the word using a dictionary for antonyms. For example, if the query is "Is safety of LASIK operation high?," and if "risk" and "low" are input into the dictionary as antonyms of "safety" and "high," respectively, then the two generated inverse queries are "Is risk of LASIK operation high?" and "Is safety of LASIK operation low?" Positive keywords are words in the given query that are to be replaced with their opposite words in the inverse queries. On the other hand, the newly introduced opposite words in the inverse queries are regarded as negative keywords. In the above example, "safety" and "high" are positive keywords, whereas "risk" and "low" are negative ones. If there is no replaceable word, the inverse query is not generated.

3.3 Web Document Retrieval

Using a given query and the corresponding inverse queries, Web documents are retrieved via TSUBAKI (Shinzato et al., 2008); TSUBAKI is an open-search engine with aninfrastructure based on deep Japanese natural language processing, and it can accept a natural-language sentence as a query. The number of documents retrieved per query is tentatively set to 100 in our experiment described in Section 4.

The retrieved documents are classified into the following three document sets: one containing documents retrieved by the given query but not retrieved by the inverse queries, one containing documents retrieved by the inverse queries but not retrieved by the given query, and one containing documents retrieved by both the given query and the inverse queries. These three document sets are termed D_{query}, $D_{inverse}$, and D_{both}, respectively. Words appearing frequently in D_{query} are likely to be positive keywords, and those appearing frequently in $D_{inverse}$, negative keywords.

TSUBAKI has an optional function for automatically expanding a submitted query by including the negative form of the main verb in the query. For example, if "Is safety of LASIK operation high?" is the submitted query, then the expanded query is "Is safety of LASIK operation not high?" The documents retrieved for the expanded query derived from the given query are regarded as documents retrieved for inverse queries. Similarly, the documents retrieved for the expanded query derived from the inverse queries are regarded as documents retrieved for the given query. Hence, even if there is no inverse query, possibly conflicting documents can be retrieved.

Table 1: An example of extracted keywords.

word	$rank_{tf}$	$rank_{POS}$	$rank_{NEG}$	polarity
LASIK	1	1	1	other
operation	2	2	2	other
eyesight	3	3	3	other
examination	19	13	60	positive
glasses	47	58	75	other
blindness	61	5,045	20	negative
effect	66	72	111	positive
complications	77	206	58	negative

3.4 Keyword Extraction

Positive and negative keywords are extracted from the retrieved Web documents as follows. First, the positive score $sc_{POS}(w)$ and negative score $sc_{NEG}(w)$ of a word w are calculated by Equations (1) and (2), respectively:

$$sc_{POS}(w) = \frac{df(w, D_{query}) \cdot tf(w)}{df(w, D_{inverse}) + 1} \quad (1)$$

$$sc_{NEG}(w) = \frac{df(w, D_{inverse}) \cdot tf(w)}{df(w, D_{query}) + 1} \quad (2)$$

where $tf(w)$ is the frequency of w in all retrieved documents, and $df(w, D)$ is the number of documents containing w in D. The $sc_{POS}(w)$ is higher and $sc_{NEG}(w)$ is lower if the word w appears more frequently in the documents retrieved by the given query and less frequently in the documents retrieved by the inverse queries. The frequency $tf(w)$ is

used to express these scores in order to consider the global importance of w in the entire retrieved document set.

Because words such as "LASIK" in the query "Is safety of LASIK operation high?" have high scores in terms of both $sc_{POS}(w)$ and $sc_{NEG}(w)$, such words should not be extracted as positive or negative keywords. Because the number of documents in D_{query} is different from that in $D_{inverse}$, $sc_{POS}(w)$ cannot be directly compared with $sc_{NEG}(w)$. Hence, $sc_{POS}(w)$ and $sc_{NEG}(w)$ are normalized, and $rank_{POS}(w)$ and $rank_{NEG}(w)$ are compared. The ranking functions $rank_{POS}(w)$ and $rank_{NEG}(w)$ are defined as the nth place ranks when all words are ranked in the descending order of $sc_{POS}(w)$ and $sc_{NEG}(w)$, respectively. We consider the top C_{rank} words on the $tf()$ ranking as candidates for possible keywords. If $rank_{POS}(w) - rank_{NEG}(w)$ or $rank_{NEG}(w) - rank_{POS}(w)$ is greater than C_{dif}, then we regard w as a positive or negative keyword, respectively. Table 1 shows an example of the extracted keywords when the given query is "Is safety of LASIK operation high?" In this paper, C_{rank} and C_{dif} are tentatively set to 100 and 20, respectively, in a preliminary experiment using several queries except the ones described in Section 4.[4] Finally, the positive/negative keywords, mentioned in Section 3.2, are respectively added to the positive/negative keyword sets described above.

The topic keywords are the words in the given query excluding the positive/negative keywords.

3.5 Passage Extraction

Passages suitable for a mediatory summary are extracted through the following four stages.

For all sentences in the document sets described in Section 3.3, the first stage involves the recognition of sentences that are useless for mediatory summary. We regard insufficient or incomplete sentences as useless sentences. We simply consider a sentence to be sufficient when the sentence has, at least, one verb phrase and one noun phrase and when the sentence contains more than two verb/noun phrases. We simply consider a sentence to be insufficient if it is not a sufficient sentence. When the expression "...," which indicates an omission, appears at the end of a sentence, then we consider the sentence incomplete. This recognition of sentences plays an important role in the calculation of scores in subsequent stages.

The second stage involves the calculation of the score of each sentence. When KW is defined as a set of all the topic, positive, and negative keywords, the basic score $sc_{BAS}(s)$ of sentence s is calculated by Equation (3).

$$sc_{BAS}(s) = \frac{\sum_{w \in KW} appear(w, s)}{|KW|} \qquad (3)$$

where $appear(w, s)$ is a function whose value is 1 if w appears in s and 0 otherwise. If s contains many different keywords, $sc_{BAS}(s)$ acquires a high value. If s is recognized as a useless sentence, then $sc_{BAS}(s)$ is multiplied by $C_{useless}$ as a penalty. In this paper, $C_{useless}$ is tentatively set to 0.5 in the case of ungrammatical sentences and 0 in the case that the end of the sentences is omitted. When the score of a sentence is calculated, the fairness described in Section 2 is determined in the following manner. If s contains positive/negative keywords besides topic keywords, the $sc_{BAS}(s)$ is multiplied by C_{basic}. The negative form of positive/negative keywords is considered, and C_{basic} is tentatively set to two if either a positive or negative expression appears in s and to three if both positive and negative expressions appear in s.

The third stage involves the application of a smoothing method to raw scores of a sentence in order to suppress over-fragmentation of passages. As given in Equation (4), the smoothed score $sc_{SMO}(s_i)$ for the ith sentence s_i in a document is calculated using the Hann

[4]The parameters used in this paper are set tentatively. Determining the optimal parameters is a topic for future research.

function, whose window length is L.

$$sc_{SMO}(s_i) = \sum_{j=-\frac{L}{2}}^{\frac{L}{2}} (sc_{BAS}(s_{i+j}) \cdot hf(j)) \quad (4)$$

$$hf(k) = 0.5 + 0.5 \cos 2\pi \frac{k}{L} \quad (5)$$

The value of L is tentatively set as 5 in this study. Insufficient sentences may convey useful information to readers when they are embedded in an appropriate context. On the other hand, incomplete sentences do not. Therefore, $sc_{SMO}(s)$ of such omitted sentences is set as 0 even after smoothing. If all types of keywords appear in the Hann window, then $sc_{SMO}(s)$ is multiplied by C_{smooth} because a passage containing s is likely to be a part of a mediatory summary. The value of C_{smooth} is tentatively set as 2 in this study.

The fourth stage involves the extraction and ranking of passages. Every series of sentences with $sc_{SMO}(s)$ greater than $\frac{1}{N}$ of the maximum score in the document is extracted as a passage. N is tentatively set as 3. The score $sc_{PAS}(p)$ of passage p is the highest score $sc_{SMO}(s)$ of sentence s in the passage. If all types of keywords appear in p, the $sc_{PAS}(p)$ is multiplied by $C_{passage}$, as in the third stage. Note that the length of the extracted passages is not set to L sentences and that passages that contain sentences multiplied by C_{smooth} are not always multiplied by $C_{passage}$. $C_{passage}$ is tentatively set as 3 in this study. Because of summarization, the passages whose lengths are nearer to the ideal length C_{length} are ranked higher. The final score $sc_{FIN}(p)$ is calculated by Equation (6).

$$sc_{FIN}(p) = \exp\left(sc_{PAS}(p) - \alpha \cdot er(p)\right) \quad (6)$$

$$er(p) = |C_{length} - nc(p)| \quad (7)$$

where $nc(p)$ is the number of characters in p, and α is a coefficient. C_{length} and α are tentatively set to 300 and 0.02, respectively.

4 Experiment

4.1 Conditions

Because the proposed method is the first method for automated generation of mediatory summary, there is no existing method to directly compare our proposed method with. Therefore, we compare the proposed method with the following three methods. The first method, $KWtf$, uses frequent words instead of the three types of keywords described in Section 3.4. The second method, Lin_{TSU}, uses an existing summarization module for summarizing the top documents in order of the score in which TSUBAKI retrieves them. The third method, Lin_{scFIN}, uses the same existing summarization module for summarizing documents containing the top passages in order of the score $sc_{FIN}()$ described in Section 3.5. We employ Lingua::JA::Summarize::Extract[5] as a summarization module, which extracts sentences containing more characteristic words; this extraction is based on the word frequency and word bigram frequency in a given document.

$KWtf$ is compared with the proposed method in order to investigate the effectiveness of keywords in terms of polarity. One of the functions of the proposed method is the classification of the top C_{rank} words on the $tf()$ ranking into positive keywords, negative keywords, and others in order to determine the fairness described in Section 2. $KWtf$ is simply based on frequent words and does not classify them. In other words, all of the top C_{rank} words on the $tf()$ ranking are used as keywords without polarity. It should be noted that no rewards can be obtained using C_{basic}, C_{smooth}, and $C_{passage}$ in the passage extraction described in Section 3.5, although penalties can be obtained using $C_{useless}$ and C_{length}.

Lin_{TSU} is compared with the proposed method in order to clarify the difference between the summarization by our method and that by a method used for general purposes. In other words, we investigate whether or not the extraction of sentences containing more characteristic words is sufficient for generating mediatory summaries.

Lin_{scFIN} is compared with the proposed method in order to investigate the appropri-

[5] http://search.cpan.org/~yappo/Lingua-JA-Summarize-Extract-0.02/

Table 2: Average precision of appropriateness of summaries generated by each query.

	Top 3	Top 5	Top 10		Top 3	Top 5	Top 10
Are diesel engines harmful to the environment?				Is safety of LASIK operation high?			
Proposed	100.0%	60.0%	36.7%	Proposed	33.3%	20.0%	20.0%
KWtf	0.0%	0.0%	0.0%	KWtf	0.0%	0.0%	0.0%
Lin$_{TSU}$	66.7%	40.0%	26.7%	Lin$_{TSU}$	33.3%	20.0%	30.0%
Lin$_{scFIN}$	0.0%	13.3%	16.7%	Lin$_{scFIN}$	0.0%	0.0%	0.0%
Are whales endangered species?				Does asbestos have toxics?			
Proposed	55.6%	33.3%	30.0%	Proposed	33.3%	40.0%	56.7%
KWtf	0.0%	0.0%	0.0%	KWtf	33.3%	20.0%	30.0%
Lin$_{TSU}$	11.1%	20.0%	13.3%	Lin$_{TSU}$	33.3%	20.0%	30.0%
Lin$_{scFIN}$	22.2%	13.3%	10.0%	Lin$_{scFIN}$	0.0%	20.0%	30.0%
Is catch and release a better way of fishing?				Does carbon dioxide cause global warming?			
Proposed	33.3%	26.7%	13.3%	Proposed	0.0%	0.0%	6.7%
KWtf	0.0%	0.0%	6.7%	KWtf	0.0%	0.0%	0.0%
Lin$_{TSU}$	66.7%	46.7%	26.7%	Lin$_{TSU}$	11.1%	6.7%	3.3%
Lin$_{scFIN}$	33.3%	26.7%	13.3%	Lin$_{scFIN}$	0.0%	0.0%	13.3%

ateness of the extracted passages. Another function of the proposed method is the reranking of passages regardless of the order of documents containing the passages during document retrieval. Therefore, a set of documents summarized by the proposed method may be different from the set of documents summarized by Lin_{TSU}. Therefore, it is observed that Lin_{scFIN} handles the same documents as the proposed method.

Because the mediatory summary is a novel concept, methods for evaluating it have not been developed yet. Although ROUGE (Lin and Hovy, 2003) is one of the most popular methods for evaluation of summaries, it may not be appropriate for the evaluation of the mediatory summary because the scoring based on N-gram in this method cannot be used to consider the fairness described in Section 2. Therefore, we evaluate the methods through the binary judgment of three human assessors, If the top summaries produced by each method are deemed to be appropriate by the three human assessors, we will be able to facilitate users' assessments of the contradictory opinions that are relevant to the given query.

Because we consider that filling a passage with all the information necessary to facilitate users' assessments is more important than shortening the passage under conditions for generation of mediatory summary, we imposed no limitation on the length of passages but the resultant penalty is obtained using Equations (6) and (7). The average length of all summaries generated by the proposed method and $KWtf$ was 288.4 characters, and none of the summaries exceeded 500 characters. Therefore, we allowed Lin_{TSU} and Lin_{scFIN} to generate summaries as long as 500 characters, which is about 200 characters longer than summaries generated by the proposed method and $KWtf$. We instructed the assessors to not judge the appropriateness of the summaries on the basis of their length.

For the experiment, we prepared the following six queries: "Are diesel engines harmful to the environment?," "Is safety of LASIK operation high?," "Are whales endangered species?," "Does asbestos have toxics?," "Is catch and release a better way of fishing?," and "Does carbon dioxide cause global warming?" We used the Japanese morphological analyzer, MeCab.[6]

[6]http://mecab.sourceforge.net/ (in Japanese)

Table 3: Average precision of assessors in terms of appropriateness of overall summaries.

	Top 3	Top 5	Top 10
Proposed	42.6%	30.0%	27.2%
KWtf	5.6%	3.3%	6.1%
Lin$_{TSU}$	37.0%	25.6%	21.7%
Lin$_{scFIN}$	9.3%	12.2%	13.9%

4.2 Result and Consideration

The kappa values between each pair of assessors' judgments on the appropriateness of the summaries were 0.79, 0.77, and 0.76, respectively; these values indicate a high level of agreement among assessors' judgments.

Table 2 shows the average precision[7] of the assessors in terms of appropriateness of summaries on the basis of responses to each query, and Table 3 shows the overall precision. The columns in Tables 2 and 3 show the precision of appropriateness for the top 3, top 5, and top 10 summaries produced by each method. It should be noted that there are only a few passages suitable for mediatory summary in all of the retrieved documents, and therefore a method for placing such suitable passages at a higher rank is more effective. We confirmed that the proposed method provided the best overall results among all the compared methods.

The difference between the proposed method and KWtf shows that classification of frequent words into positive keywords, negative ones, and others, in other words, the fairness described in Section 2 contributed to generation of appropriate mediatory summaries.

The difference between Lin$_{TSU}$ and Lin$_{scFIN}$ indicates that the order of the score $sc_{FIN}()$ described in Section 3.5 was different from that of the score of TSUBAKI. Lingua::JA::Summarize::Extract could not extract the appropriate passages from document sets on the basis of $sc_{FIN}()$, even though the document sets contained the same appropriate passages that were extracted by the proposed method. Therefore, summarization for a general purpose is insufficient for generation of mediatory summaries, and the proposed method can provide more appropriate mediatory summaries.

However, the results of the queries "Is catch and release a better way of fishing?" and "Does carbon dioxide cause global warming?" in Table 2 show that the proposed method could not extract all the appropriate passages that were extracted by Lin$_{TSU}$. We aim to improve the proposed method in future research.

[7]We use precision, which represents *#correct outputs/#total outputs*, as an evaluation measure because it is difficult to calculate the recall of the mediatory summaries that are dynamically generated from Web documents and because users tend to read just a few of the summaries generated by the system.

5 Conclusion

We proposed a method for automated generation of mediatory summaries in order to facilitate users' assessment of the credibility of information on the Web. A mediatory summary is generated by extracting a passage from Web documents on the basis of their relevance to a given query, fairness, and density of keywords, which are features of the summaries constructed to determine the credibility of information on the Web. We demonstrated the effectiveness of the generated mediatory summary in comparison with the summaries of Web documents produced by Web search engines.

Acknowledgement

This research is partially supported by the National Institute of Information and Communications Technology, Japan.

References

Juffinger, Andreas, Michael Granitzer, and Elisabeth Lex. 2009. Blog credibility ranking by exploiting verified content. In *Proceedings of the Second Workshop on Information Credibility on the Web (WICOW 2009)*, pages 51–57.

Kaneko, Koichi, Hideyuki Shibuki, Masahiro Nakano, Rintaro Miyazaki, Madoka Ishioroshi, and Tatsunori Mori. 2009. Mediatory summary

generation: Summary-passage extraction for information credibility on the web. In *Proceedings of the 23rd Pacific Asia Conference on Language, Information and Computation (PACLIC 23)*, pages 240–249.

Kawahara, Daisuke, Tetsuji Nakagawa, Takuya Kawada, Kentaro Inui, and Sadao Kurohashi. 2009. Summarizing evaluative information on the web for information credibility analysis. In *Proceedings of the 3rd International Universal Communication Symposium (IUCS 2009)*, pages 187–192.

Lin, Chin-Yew and Eduard Hovy. 2003. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the Human Language Technology Conference 2003 (HLT-NAACL-2003)*, pages 71–78.

Miyazaki, Rintaro, Ryo Momose, Hideyuki Shibuki, and Tatsunori Mori. 2009. Using web page layout for extraction of sender names. In *Proceedings of the 3rd International Universal Communication Symposium (IUCS 2009)*, pages 181–186.

Murakami, Koji, Eric Nichols, Suguru Matsuyoshi, Asuka Sumida, Shouko Masuda, Kentaro Inui, and Yuji Matsumoto. 2009. Statement map: Assisting information credibility analysis by visualizing arguments. In *Proceedings of the Second Workshop on Information Credibility on the Web (WICOW 2009)*, pages 43–50.

Nakano, Masahiro, Hideyuki Shibuki, Rintaro Miyazaki, Madoka Ishioroshi, Koichi Kaneko, and Tatsunori Mori. 2010. Construction of text summarization corpus for the credibility of information from the web. In *Proceedings of the 7th Language Resources and Evaluation Conference (LREC 2010)*, pages 3125–3131.

Ohshima, Hiroaki, Satoshi Oyama, Hiroyuki Kondo, and Katsumi Tanaka. 2009. Web information credibility analysis by geographical social support. In *Proceedings of the 3rd International Universal Communication Symposium (IUCS 2009)*, pages 193–196.

Shinzato, Keiji, Tomohide Shibata, Daisuke Kawahara, Chikara Hashimoto, and Sadao Kurohashi. 2008. Tsubaki: An open search engine inflastructure for developing new information access methodology. In *Proceedings of the Third International Joint Conference on Natural Language Processing (IJCNLP 2008)*, pages 189–196.

Towards Automatic Building of Document Keywords

Joaquim Silva
CITI/DI/FCT
Universidade Nova de Lisboa
jfs@di.fct.unl.pt

Gabriel Lopes
CITI/DI/FCT
Universidade Nova de Lisboa
gpl@di.fct.unl.pt

Abstract

Document keywords are associated to documents as summarized versions of the documents' content. Considering that the number of documents is quickly growing every day, the availability of these keywords is very important. Although, usually keywords are manually written. This motivated us to work on an approach to change this manual procedure for an automatic one.

This paper presents a language independent approach that extracts the most relevant Multiword Expressions and single words from documents and propose them to describe the core content of each document.

1 Introduction

Keywords provide efficient and sharp access to documents concerning their main topics, that is, their core content. Keywords are semantically relevant terms, usually being relevant noun-phrases rather than long full phrases. Full phrases such as "John F Kennedy's speechwriter hails Obama's address" can be extracted by summarization approaches, but it wouldn't be appropriate if used as keywords since it doesn't mean any main topic/subtopic. On the other hand, by using LocalMaxs algorithm (Silva and Lopes, 1999) it is possible to extract Multiword Expressions (MWEs) from documents and, some of the most relevant ones relatively to each document can be used as that document's descriptor, if properly selected. In this paper we will show that MWEs having 2, 3 our 4 words, that is, (2-4)-gram MWEs, are the most appropriate ones to fit the typical keywords' semantic sharpness, as would be the case of "climate change", "American Red Cross", "social and economic policy", etc., rather than (5-7)-grams and larger MWEs addressing more specific meanings, such as "skills for lifelong learning process report" or "Assessment of the use of Magnetic Resonans Tomography".

On the other hand, although MWEs extracted by LocalMaxs algorithm are usually relevant, some of them are semantically vague or simply not relevant, such as "general use" or "Annex I", not having the semantic relevance and sharpness required to form keywords. Other MWEs such as "in case of" or "as soon as possible" may be useful for lexicon enrichment to improve Natural Language Processing, but they are not relevant MWEs to be taken as keywords.

During our investigation, we discovered that the median of the words' length in each MWE has a strong influence in the MWE relevance. Thus, combining this and other factors that influence relevance, a metric, Mk, is proposed to better evaluate the relevance of each MWE under the purpose of obtaining keywords, and consequently its relevance score in each document.

Although most document keywords are multiwords, there are some single words , that is, 1-grams, whose strong and sharp meaning make them good keywords, such as "Agriculture", "salmonella", among others. Then, since we wanted to include single words in the set of the main keywords of each document, and because LocalMaxs algorithm does not extracts 1-grams, we had to select the most informative single words

Coling 2008: Poster Volume, pages 1149–1157,
Beijing, August 2010

from documents using another metric, Sk, also presented in this paper.

This paper proposes a statistical and language-independent approach to generate document descriptors based on the automatic extraction of the most informative MWEs and single words, in terms of document summarization, under the purpose of keywords, taken from each document. Next section analyzes related work. A brief explanation of the LocalMaxs algorithm is presented in section 3. In section 4 we propose the metrics Mk and Sk and consider other measures. Results are presented in section 5 and conclusion are made in the last section.

2 Related Work

In (Cigarrán et al., 2005; Liu et al., 2009; Hulth, 2004) authors propose extraction of noun phases and keywords. However, these are not language-independent approaches, since they use some language-dependent tools such as stop-words removing, lemmatization, part-of-speech tagging or syntactic pattern recognition.

In (Delort et al., 2003), authors address the issue of Web document summarization by context. They consider the context of a Web document by the textual content of all documents linking to it. According to the authors, the efficiency of this approach depends on the size of the content and context of the target document. However, its efficiency also depends on the existence of links to the target documents.

In (Aliguliyev, 2006) a generic summarization method is proposed. It extracts the most relevance sentences from the source document to form a summary. The summary can contain the main contents of different topics. This approach is based on clustering of sentences and, although results are not shown, it does not use language-dependent tools.

Other Information Extraction methods rely on predefined linguistic rules and templates to identify certain entities in text documents (Yangarber and Grishman, 2000; Jacquemin, 2001). Again, these are not language-independent approaches, despite the good results that they give rise to.

Some approaches address specific-domain problems. In (Alani et al., 2003), authors propose a method to extract artist information, such as name and date of birth from documents and then generate his or her biography. It works with meta-data triples such as (subject-relation-object), using ontology-relation declarations and lexical information. Clearly, this approach is not language-independent. In (Velardi et al., 2001), a method to extract a domain terminology from available documents such as the Web pages is proposed. The method is based on two measures: Domain Relevance and Domain Consensus that give the specificity of a terminological candidate. In (Martínez-Fernández et al., 2004) the News specific-domain is addressed. Again, this approach is not language-independent.

A supervised approach (Ercan and Cicekli, 2007) extracts keywords by using lexical chains built from the WordNet ontology (Miller, 1991), a tool which is not available for every language.

Rather than being dependent on specific languages, structured data or domain, we try to find out more general and language-independent features from free text data.

In (Silva and Lopes, 2009), a MWEs extractor and a metric, $LeastRvar$, extracts keywords from documents. However, single words are ignored as possible keywords and their global results are outperformed by our proposal.

3 Using LocalMaxs Algorithm to Extract Keyword Candidates

We used the SCP_f cohesion metric and the LocalMaxs algorithm to extract MWEs from document *corpora*. Although details about these tools are given in (Silva and Lopes, 1999; Silva et al., 1999), here follows a brief description for paper self-containment. Thus, LocalMaxs is based on the idea that each n-gram[1] has a kind of *glue* or cohesion sticking the words together within the n-gram. Different n-grams usually have different cohesion values. One can intuitively accept that there is a strong cohesion within the n-gram "Giscard d'Estaing" i.e. between the words "Giscard" and "d'Estaing". However, one cannot say that there is a strong cohesion within the 2-grams "or given" or within the "of two". Thus, in order to

[1] $w_1 \dots w_n$ or $(w_1 \dots w_n)$ are also used to denote an n-gram of length n.

measure the cohesion value not only of 2-grams, but also for every n-gram of any size in the corpus, we used the $SCP\_f(.)$ metric:

$$SCP\_f(w_1 \dots w_n) = \frac{p(w_1 \dots w_n)^2}{Avp} \quad (1)$$

$$Avp = \frac{1}{n-1} \sum_{i=1}^{n-1} p(w_1 \dots w_i) \,.\, p(w_{i+1} \dots w_n) \quad (2)$$

where $p(w_1 \dots w_n)$ is the probability of the n-gram $w_1 \dots w_n$ in the corpus. This way, any size n-gram is *transformed* in a pseudo-bigram that reflects the *average cohesion* between any two adjacent contiguous sub-n-gram of the original n-gram. Now it is possible to compare cohesions from n-grams of different sizes.

3.1 LocalMaxs Algorithm

LocalMaxs is a language independent algorithm to filter out cohesive n-grams of text elements (words, tags or characters), requiring no threshold arbitrarily assigned.

Definition 1. *Let $W = w_1 \dots w_n$ be an n-gram and $g(.)$ a cohesion generic function. And let: $\Omega_{n-1}(W)$ be the set of $g(.)$ values for all contiguous $(n-1)$-grams contained in the n-gram W; $\Omega_{n+1}(W)$ be the set of $g(.)$ values for all contiguous $(n+1)$-grams which contain the n-gram W, and let $len(W)$ be the length (number of elements) of n-gram W. So, it is stated that*

W is a Multi Element Unit (MEU) if and only if,
for $\forall x \in \Omega_{n-1}(W), \forall y \in \Omega_{n+1}(W)$
$(len(W) = 2 \wedge g(W) > y) \quad \vee$
$(len(W) > 2 \wedge g(W) > \frac{x+y}{2})$.

Then, for n-grams with $n \geq 3$, LocalMaxs algorithm elects every n-gram whose cohesion value is greater than the average of two maxima: the greatest cohesion value found in the contiguous $(n-1)$-grams contained in the n-gram, and the greatest cohesion found in the contiguous $(n+1)$-grams containing the n-gram. Thus, in the present approach we used LocalMaxs as a MWEs extractor — MWEs are MEUs where the elements are words — and used $SCP\_f(.)$ cohesion measure as the $g(.)$ function referred in the algorithm definition above.

4 Selecting Keywords from MWEs

Not every MWE extracted by LocalMaxs has equal relevance or semantic sharpness. Some MWEs are vague in terms of semantic sharpness, such as “important meeting” or “general use”; other ones are very specific in terms of the topic they point to, for example “Assessment of the use of Magnetic Resonans Tomografy”; some others are (2-4)-gram strongly informative MWEs, fitting the semantic sharpness of typical keywords such as “computer science” or ”Food and Agriculture Organization”, and will be privileged by the metric we present in subsection 4.4.

Some single words have adequate semantic sharpness to be included as keywords, such as ”Algebra” or ”Agriculture”, among others. However, most single words are not informative enough for that purpose.

As a consequence, we felt the need to work on adequate metrics to value and privilege the strongly informative MWEs and single words in order to find keywords in documents.

4.1 The *Tf-Idf* Metric

$Tf{-}Idf$ (Term frequency−Inverse document frequency) is a statistical metric often used in IR and text mining. Usually, it is used to evaluate how important a word is to a document in a *corpus*. The importance increases proportionally to the number of times a word/multiword appears in the document but it is offset by its frequency in the *corpus*. Thus, this is one of the metrics with which we will try to privilege the most informative MWEs and 1-grams in each document.

$$Tf{-}Idf(W, d_j) = p(W, d_j) \,.\, Idf(W, d_j) \quad (3)$$

$$p(W, d_j) = \frac{f(W, d_j)}{N_{d_j}} \quad (4)$$

$$Idf(W, d_j) = \log \frac{\|\mathcal{D}\|}{\|\{d_j : W \in d_j\}\|} \quad (5)$$

where $f(W, d_j)$ if the frequency of word/multiword W in document d_j and N_{d_j} stands for the number of words of d_j; $\|\mathcal{D}\|$ is the number of documents of the *corpus*. So, $Tf{-}Idf(W, d_j)$ will give a measure of the importance of W, that is a MWE or a single word, within the particular document d_j. By the structure of term Idf we can see

that it privileges MWEs and single words occurring in less documents, particularly those occurring in just one document.

4.2 The *LeastRvar* Metric

Most weakly relevant MWE and errors extracted by LocalMaxs begin or end with a so called stopword, that is a highly frequent word appearing in most documents. However, stop-words may exist in the middle of a relevant MWE, for example "United States of America" or "Life on Mars"; but usually not in the leftmost or rightmost word of the MWEs. By considering this, $LeastRvar$ was proposed in (Silva and Lopes, 2009):

$$LeastRvar(MWE_i) = least(Lrv, Rrv) \quad (6)$$

where $Lrv = Rvar(leftmostw(MWE_i))$,
$Rrv = Rvar(rightmostw(MWE_i))$

and

$$Rvar(W) = \frac{1}{\|\mathcal{D}\|} \sum_{d_i \in \mathcal{D}} \Big(\frac{p(W,d_i) - p(W,.)}{p(W,.)} \Big)^2 \ . \quad (7)$$

$p(W,.)$ means the average probability of the word W considering all documents. $Rvar(.)$ is applied to the leftmost and the rightmost word of the MWE:

$$p(W,.) = \frac{1}{\|\mathcal{D}\|} \sum_{d_i \in \mathcal{D}} p(W,d_i). \quad (8)$$

$Rvar(W)$ measures the variation of the probability of the word W along all documents. Apparently the usual formula of the variance (the second moment about the mean), would measure that variation; however, it would wrongly benefit the very frequent words such as "of", "the" or "and", among others. This happens because the absolute differences between the occurrence probabilities of any of those frequent words along all documents is high, regardless of the fact that they usually occur in every document. These differences are captured and over-valued by the variance since it measures the average value of the quantity $(distance\, from\, mean)^2$, ignoring the *order of magnitude* of the individual probabilities. Then, $Rvar(.)$ divides each *individual distance*, in the original formula of the variance, by the order of magnitude of these probabilities, that is, the mean probability, given by $p(W,.)$; see equations 7 and 8.

Then, $LeastRvar(MWE_i)$ is given by the least $Rvar(.)$ values considering the leftmost word and the rightmost word of MWE_i. This way, $LeastRvar(.)$ tends to privilege informative MWEs and penalize those multiword expressions having semantically meaningless words in the begin or in the end of it.

4.3 The *LeastCv* metric

In oder to try to obtain better results than those produced by *LeastRvar*, we changed $Rvar(.)$ to an alternative to measure the relative variation of the probability of the leftmost and rightmost words in MWEs. Then we defined:

$$LeastCv(MWE_i) = least(Lcv, Rcv) \quad (9)$$

where $Lcv = Cv(leftmostw(MWE_i))$,
$Rcv = Cv(rightmostw(MWE_i))$,

$$Cv(W) = \sigma(W)/\mu(W) \ , \quad (10)$$

$$\sigma(W) = \sqrt{\frac{1}{\|\mathcal{D}\|} \sum_{d_i \in \mathcal{D}} (p(W,d_i) - p(W,.))^2} \ , \quad (11)$$

and

$$\mu(W) = p(W,.) \ ; \quad (12)$$

$p(W,d_i)$ and $p(W,.)$ have the same meaning as in equation 7. The reader may recognize $Cv(.)$ as the *coefficient of variation*, which is given by the ratio of the standard deviation σ to the mean μ. Results in section 5 will show that $LeastCv$ also tends to privilege informative MWEs.

4.4 Two New Metrics to Find Keywords

Considering the results obtained for *LeastRvar* and *LeastCv*, as we will see in section 5, we wanted to develop a better metric to find MWE keywords and another one for single word keywords. They were built by combining some important factors that we present next.

The Median of the MWE Words' Length: Since most of the semantically meaningless words

are small and long words usually have sharp meaning, we considered the median length of the words in each MWE to help on selecting the most informative MWEs. By comparison, median length showed better results than average length. For example, MWE "Language Institute" has an average word length of 8.5 characters, but the semantically equivalent "Institute of Languages" has a different average length of 6.66. On the contrary, the median length for both MWEs presents more close values: $((8+9)/2 = 8.5)$ for "Language Institute" and 9 for "Institute of languages" (the middle number after sorting the MWE words length: 2, 9 and 9). Thus, because the median values is more robust to outliers than the average value, the length of the meaningless word "of" was, say, *ignored* in the median calculation. In fact, those equivalent meaning MWEs have similar median length values (8.5 and 9), but not so similar average length values (8.5 and 6.66). Furthermore, the robustness of the median length enables more similar values when considering MWEs in English and other equivalent MWEs in other languages where stop words are more used; for example "écoles de conduite" (driving schools), "producción de batata" (potato production), etc..

How Many Words for a Keyword? As the reader may check in documents having associated keywords, we noticed that the main document keywords are usually (2-4)-grams. So, we defined a factor, $Ckl(MWE_i)$, to measure how similar is the *pseudo number of words* of MWE_i to the *typical* number of words of keywords. We define the *pseudo number of words* of a MWE:

$$Pnw(MWE_i) = \frac{NumChars(MWE_i)}{Med(MWE_i)} \ . \tag{13}$$

$NumChars(MWE_i)$ stands for the number of characters of MWE_i and $Med(MWE_i)$ is the median length of its words. $Pnw(MWE_i)$ gives a value close to the number of meaningful words of MWE_i. For example, Pnw("Institute of Languages") $= 20/9 = 2.2$ (close to 2); Pnw("European Council") $= 15/7.5 = 2$, etc.. Now, $Ckl(.)$ is given by:

$$Ckl(MWE_i) = \frac{1}{|Pnw(MWE_i) - T| + 1} \ , \tag{14}$$

where T is the *typical* number of words of the keywords. Maximum value for $CkLen(MWE_i)$ is 1; it happens if $Pnw(MWE_i)$ equals to T. As we will see by the results in section 5, we tried two T values: 2.5 and 3.5; and compared results.

The *Mk* Metric for MWE Keywords: We built $Mk(.)$ metric by improving $LeastRvar(.)$:

$$Mk(M) = LeastRvar(M).Med(M).Ckl(M) \tag{15}$$

Thus, $Mk(.)$ privileges MWEs having not only informative leftmost and rightmost words, but also having long words and a *pseudo number of words* close to the number of words of typical keywords – for reasons of lack of space, we used M instead of MWE_i in equation 15 –.

The *Sk* Metric for *Single Word* Keywords: We built $Sk(.)$ from $Rvar(.)$ – see equation 7 – to measure how meaningful is each single word:

$$Sk(W_i) = Rvar(W_i).Len(W_i) \ . \tag{16}$$

$Len(W_i)$ means the length of words W_i. Thus, $Sk(.)$ privileges single words having, not only a high relative variation of their probabilities along all documents, but also being long words.

5 Results

We analyze the quality of the document descriptors after applying the LocalMaxs extractor followed by each of the six different metrics to three different document *corpora*, each one for a different language: English, French and Spanish. Metrics applied to MWEs were $Tf{-}Idf$, $LeastCv$, $LeastRvar$, $Mk\,[2.5]$ – that is $T = 2.5$ in equation 14; and $Mk\,[3.5]$. Metrics applied to single words were $Tf{-}Idf$ and Sk.

5.1 The Document Descriptor

We decided to represent the core content of each document by using its 15 most informative terms, in the sense of keywords: 11 MWEs and 4 single words. An independent evaluation criteria were

defined by Prof. Francisca Xavier from the Linguistics Department of *Universidade Nova de Lisboa*. It was considered that, for example, "aim of mission" and "16 December 2003" are wrong keywords, as the first one is a too vague noun phrase and the second one, just a simple date. Relevant MWEs such as "nuclear weapons" and "financial crisis" were evaluated as keywords. However, although some proposed multi-word expressions are not keywords, they are informative in the context of the descriptor and correspond to well formed morphosyntactic tags, for example, "56% of GDP" or "comfort zone": these *near-miss* cases were classified as half-correct half-wrong terms; the same classification was given to single words such as "macro-economic" – see table 7 – which, although it's not a noun, it's an informative adjective.

Thus, for each document, the extracted MWEs are sorted according to each metric and the top 11 MWEs are taken as the document's MWEs descriptor. The single words of the document are also sorted according to one of the two applied metrics ($Tf{-}Idf$ or Sk). By ignoring the rest of the MWEs and single words, there is document information which will be *lost* by these descriptors, but they must be taken as core content descriptors, not as complete/detailed reports of the documents. Although descriptors are composed by MWEs and single words, for better comparison of the metrics, tables separately show MWE descriptors or *single word* descriptors. Table 1 shows an example of a document MWE descriptor resulting from the application of one of the metrics (Mk) to the document's MWEs extracted by LocalMaxs algorithm:

Table 1: Example of an English Document MWE Descriptor – Application of the $Mk\,[2.5]$ Metric.

enterprise profits
comfort zone
medium-sized enterprises
brain drain
cold war
Balance of Payment
56% of GDP
excessive deficit
looking ahead
exports and imports
Stability and Growth Pact

5.2 The Multi-Language *Corpora* Test

We used the *EUR-Lex corpora*, http://eur-lex.europa.eu/en/, containing European Union law documents about several topics in several European languages. We took 60 documents written in each language, English, French and Spanish to form three different *sub-corpora*. These are unstructured row text documents.

To evaluate the approach's performance, we used Precision and Recall concepts. Precision was given by the number of keywords in the set of the 11 most scored MWEs proposed as descriptor, by the combination LocalMaxs$-$metric used, divided by 11. Recall was given by the number of keywords that are simultaneously in the document's descriptor proposed and in the set made of the 11 most informative keywords of the document, divided by 11.

According to the criteria mentioned above, this is the evaluation of the descriptor shown in table 1, considering Precision: 8 MWEs can be accepted as keywords (1st, 3rd, 4th, 5th, 6th, 8th, 10th and 11th); 2 near-miss MWEs (2nd and 7th); and 1 weak or wrong MWE (9th). So, precision is $(8 + 2 * 0.5)/11 = 0.818$. Concerning the document this descriptor represents, there are 3 strong keywords that should be in the descriptor, but they weren't: "financial crisis", "structural reforms" and "macroeconomic imbalances". Thus, Recall is $8/11 = 72.7$ for this case.

5.3 Results for Different Metrics and Languages

By table 2 we may see that for the same metric, Precision or Recall values are similar for English, French and Spanish. So, this approach does not seem to privilege any of these languages, and we believe that probably this happens for many other languages, as no specific morphosyntactic information was used. Even the difference between Recall values for Spanish and English produced by $LeastRvar$ (0.61 and 0.63) would probably decrease if the test *corpora* had more documents. Table 2 also shows that $Tf{-}Idf$ presents the poor-

Table 2: Precision and Recall Average Values for the Document MWE Descriptors.

Language	Metric	Precision	Recall
English	$Tf–Idf$	0.51	0.35
	$LeastCv$	0.62	0.61
	$LeastRvar$	0.65	0.63
	$Mk\,[2.5]$	**0.76**	**0.72**
	$Mk\,[3.5]$	0.74	0.68
French	$Tf–Idf$	0.50	0.35
	$LeastCv$	0.62	0.60
	$LeastRvar$	0.64	0.63
	$Mk\,[2.5]$	**0.75**	**0.71**
	$Mk\,[3.5]$	0.73	0.68
Spanish	$Tf–Idf$	0.51	0.34
	$LeastCv$	0.61	0.60
	$LeastRvar$	0.64	0.61
	$Mk\,[2.5]$	**0.75**	**0.72**
	$Mk\,[3.5]$	0.74	0.67

est results. In fact, due to its structure — see equation 3 — we can see that MWEs that occur many times in just one document are the most valued/privileged ones. This explains why the descriptors made by this measure tend to include too specific/local MWEs, regardless of some important ones. Table 3 shows a document descriptor generated by the combination LocalMaxs$–Tf–Idf$: for example MWE "new Members" occurs in just one document, 10 times; however, "new Members" is not a keyword. This is the descriptor of the same document from where other descriptors were generated by the combinations including $LeastRvar$ and $Mk\,[2.5]$, and shown in tables 4 and 1.

For reasons of space limitation we don't show descriptors produced by $LeastCv$ and $MK\,[3.5]$ metrics. However, table 2 shows that $LeastCv$ was outperformed by $LeastRvar$. This table also shows that $Mk\,[2.5]$ metric presents the highest Precision (0.76, 0.75 and 0.75 for English, French and Spanish). The highest Recall values are also obtained for the same metric: 0.72, 0.71 and 0.72 for the same languages.

Tables 5 and 6 show examples of MWE descriptors of French and Spanish documents, by the application of $Mk\,[2.5]$ as it produced the best results.

Table 3: Example of an English Document MWE Descriptor – Application of the $Tf–Idf$ Metric.

in the new Member States
in the new Member
new Members
Single Market
income convergence
some of the new Member
financial crisis
structural reforms
new and old
euro area
reap the full benefits of the Single Market

Table 4: Example of an English Document MWE Descriptor – Application of the $LeastRvar$ Metric.

five years
Cold War
old Members
enterprise profits
Central Bank
Excessive Deficit
medium-sized enterprises
comfort zone
56% of GDP
1.5% of GDP
brain drain

Tables 7 and 8 show examples of *single word* descriptors for the same document described in table 1. As we could expect, Precision and Recall values for *single word* descriptors are lower than the values for MWEs descriptors, since singles words are usually semantically less sharp than multiwords: see table 9. Sk shows better performance than $Tf–Idf$, specially for Recall.

6 Conclusions

Keywords are semantic tags associated to documents, usually declared manually by users. These tags form small document descriptors and enable applications to access to the summarized documents' core content. This paper proposes an approach to automatically generate document de-

Table 5: Example of a French Document MWE Descriptor – Application of the $Mk\,[2.5]$ Metric.

moto-fraises et motofaucheuses
agrumeraies et oliveraies
hommes Travail
Fumier liquide
familiale occupée
Mieux légiférer
d'arbres fruitiers
Superficie irriguée
Main-d'oeuvre non familiale
activités lucratives
Alignements d'arbres

Table 6: Example of a Spanish Document MWE Descriptor – Application of the $Mk\,[2.5]$ Metric.

ingredientes de cosméticos
combinaciones de ingredientes
someter a ensayo
Sustancias y Preparados
toxicidad aguda
irritación ocular
fototoxicidad aguda
explicaciones dadas
corrosión cutánea
animales utilizados
Sustancias y Preparados Químicos

Table 7: Example of an English Document *Single Word* Descriptor – Application of the Sk Metric.

vulnerabilities
growth-enhancing
post-enlargement
macro-economic

Table 8: Example of an English Document *Single Word* Descriptor – Application of the Tf–Idf Metric.

economic
new
enlargement
reforms

Table 9: Precision and Recall Average Values for the Document *Single Word* Descriptors.

Language	Metric	Precision	Recall
English	Tf–Idf	0.52	0.36
	Sk	**0.55**	**0.48**
French	Tf–Idf	0.51	0.37
	Sk	**0.54**	**0.47**
Spanish	Tf–Idf	0.52	0.37
	Sk	**0.56**	**0.48**

scriptors, as a language-independent and domain-independent alternative to related work from other authors. This approach uses LocalMaxs algorithm to extract MWEs, and two new statistical metrics, Mk and Sk, to select the 15 most relevant MWEs and single words from each document in order to form document descriptors.

Comparing the results produced by Mk with the second best metric, $LeastRvar$, we may conclude that the introduction of the median of the words' length of each MWE and the preference for (2-4)-grams, improve the quality of document descriptors by about 11% and 9% for Precision and Recall, respectively. Furthermore, by comparison of $Mk\,[2.5]$ and $Mk\,[3.5]$ results we conclude that keywords are mostly (2-3)-grams, rather than (3-4)-grams or longer n-grams.

Results also showed that Precision and Recall values are similar for the three languages tested (English, French and Spanish), which enable us to expect similar performance to other languages. Apart from the Precision and Recall values, document descriptors made by this approach does indeed capture the core content of each document. We believe this may contribute to improve document summarization. Future work will include tests in other languages and we will work to improve results, specially for single words.

References

Alani, Harith, Kim Sanghee, David E. Millard, Mark J. Weal, Paul H. Lewis, Wendy Hall and Nigel Shadbolt. 2003. Automatic Extraction of Knowledge from Web Documents. In *Proceedings of Workshop of Human Language Technology for the Semantic Web and Web Services, 2nd International Seman-*

tic Web Conference. October 20th, Sanibel Island, Florida, USA.

Aliguliyev, Ramiz M. 2006. A Novel Partitioning-Based Clustering Method and Generic Document Summarization. In *Proceedings of the 2006 IEEE/Web Intelligence/Association for Computing Machinery and the Intelligent Agent Technology International Conference (2006 Workshops)(WI-IATW'06)*. December 18-22, Hong Kong, China.

Cigarrán, Joan. M., Anselmo Peas, Julio Gonzalo and Felisa Verdejo. 2005. Automatic Selection of Noun Phrases as Document Descriptors in an FCA-Based Information Retrieval System. B. Ganter and R. Godin (Eds.). ICFCA 2005, *Lecture Notes in Computer Science* 3403, pp. 49-63. Springer-Verlag.

Ciravegna, Fabio, Alexeie Dingli, David Guthrie and Yorick Wilks. 2003. Mining Web Sites Using Unsupervised Adaptive Information Extraction. In *Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics*. April, 12-17. Budapest, Hungary.

Delort, J.-Y., B. Bouchon-Meunier and M. Rifqi. 2003. Enhanced Web Document Summarization Using Hyperlinks. In *Proceedings of the fourteenth Association for Computing Machinery conference on Hypertext and hypermedia*. August 26-30, Nottingham, UK.

Ercan, Gonenc and Ilyas Cicekli. 2007. Using lexical chains for keyword extraction. *Information Processing and Management: an International Journal archive*. Volume 43, Issue 6, November, Pages 1705-1714, Pergamon Press, Inc.. ISSN 0306-4573.

Hulth, Anette. 2004. Enhancing linguistically oriented automatic keyword extraction. *Proceedings of Human Language Technology-North American Association for Computational Linguistics 2004 conference*. Pag.17-20. May 02-07. Boston, Massachusetts. Publisher: Association for Computational Linguistics, Morristown, NJ, USA.

Jacquemin Christian. 2001. *Spotting and Discovering Terms through Natural Language Processing*. MIT Press, ISBN 0262100851.

Liu, Feifan, Deana Pennell, Fei Liu and Yang Liu. 2009. Unsupervised approaches for automatic keyword extraction using meeting transcripts. *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. May 31-June 05. Boulder, Colorado.

Martínez-Fernández, J. L., A. García-Serrano, P. Martínez, J. Villena. 2004. Automatic Keyword Extraction for News Finder. *Lectures Notes in Artificial Intelligence*, Springer-Verlag, volume 3094, pages 99–119.

Miller, George A. 1991. *The science of words.* Scientific American Library, New York.

Silva, Joaquim and Gabriel Lopes. 1999. A Local Maxima Method and a Fair Dispersion Normalization for Extracting Multi-word Units. *In Proceedings of the 6th Meeting on the Mathematics of Language*, pages 369-381. 23-25 July, University of Central Florida, Orlando.

Silva, Joaquim and Gabriel Lopes. 2009. A Document Descriptor Extractor Based on Relevant Expressions. *14 Encontro Portugus para a Inteligncia Artificial (Fourteenth Portuguese Conference on Artificial Intelligence)*. October 12-15. Univerity of Aveiro. Lectures Notes in Artificial Intelligence, Springer-Verlag, volume 5816, pages 646-657.

Silva, Joaquim, Gael Dias, Sylvie Guilloré and Gabriel Lopes. 1999. Using LocalMaxs Algorithm for the Extraction of Contiguous and Non-contiguous Multi-word Lexical Units. 9th Portuguese Conference on Artificial Intelligence. September, vora,Portugal. *Lectures Notes in Artificial Intelligence*, Pedro Barahora and Jos Alferes (Eds.). Springer-Verlag, volume 1695, pages 113-132.

Yangarber, Roman and Ralph Grishman. 2000. Machine Learning of Extraction Patterns from Unanotated Corpora: Position Statement. *Workshop on Machine Learning for Information Extraction. Held in conjunction with the 14th European Conference on Artificial Intelligence (ECAI)*. 21 August. Berlin, Humboldt University.

Velardi, Paula, Michele Missikoff, and Roberto Basili. 2001. Identification of relevant terms to support the construction of Domain Ontogies. *Association for Computational Linguistics-European Association for Computational Linguistics Workshop on Human Language Technologies*. July 6-7. Toulouse, France.

Shallow Information Extraction from Medical Forum Data

Parikshit Sondhi and **Manish Gupta** and **ChengXiang Zhai** and **Julia Hockenmaier**
Department of Computer Science
University of Illinois at Urbana Champaign
{sondhi1, gupta58, czhai, juliahmr}@illinois.edu

Abstract

We study a novel shallow information extraction problem that involves extracting sentences of a given set of topic categories from medical forum data. Given a corpus of medical forum documents, our goal is to extract two related types of sentences that describe a biomedical case (i.e., medical problem descriptions and medical treatment descriptions). Such an extraction task directly generates medical case descriptions that can be useful in many applications. We solve the problem using two popular machine learning methods Support Vector Machines (SVM) and Conditional Random Fields (CRF). We propose novel features to improve the accuracy of extraction. Experiment results show that we can obtain an accuracy of up to 75%.

1 Introduction

Conventional information extraction tasks generally aim at extracting finer granularity semantic information units such as entities and relations. While such detailed information is no doubt very useful, extraction of such information also tends to be difficult especially when the mentions of the entities to be extracted do not conform to regular syntactic patterns.

In this paper, we relax this conventional goal of extraction and study an easier extraction task where we aim at extracting sentences that belong to a set of predefined semantic categories. That is, we take a sentence as a unit for extraction. Specifically, we study this problem in the context of extracting medical case description from medical forums.

A variety of medical health forums exist online. People use them to post their problems, get advices from experienced patients, get second opinions from other doctors, or merely to vent out their frustration.

Compared with well-structured sources such as Wikipedia, forums are more valuable in the sense that they contain first hand patient experiences with richer information in terms of what treatments are better than others and why. Besides this, on forums, patients explain their symptoms much more freely than those mentioned on relatively formal sources like Wikipedia. And hence, forums are much more easier to understand for a naïve user.

However, even on targeted forums (which focus on a single disease), data is quite unstructured. There is therefore a need to structure out this information and present it in a form that can directly be used for a variety of other information extraction applications like the collecting of medical case studies pertaining to a particular disease, mining frequently discussed symptoms, identifying correlation between symptoms and treatments, etc.

A typical medical case description tends to consist of two aspects:

- **Physical Examination/Symptoms (PE)**: This covers current conditions and includes any condition that is the focus of current discussion. Note that if a drug causes an allergy, then we consider it as a PE and not a medication. Any condition that is the focus of conversation, i.e. around which

Coling 2008: Poster Volume, pages 1158–1166,
Beijing, August 2010

treatments are being proposed or questions are being asked is considered PE even if the user is recounting their past experience.

- **Medications (MED)**: Includes medications the person is currently taking, or is intending to take, or any medication on which the question is targeted. Medications do not necessarily mean drugs. Any measures (including avoiding of substances) taken to treat or avoid the symptoms are considered as medication. Sometimes, users also mention other things like constituents of the drug, how much of the drug to consume at a time, how to get access to a medication, how much it costs, side effects of medications, other qualities of medications etc.

Figure 1 shows an example of PE and MED labelings.

<MED>i was told hot peppers ie in salsa,
mexican,spicy,szechuan/polynesian type foods are great treatments.</MED>
<PE>They help against nasal/sinusitis/rhinitis conditions.</PE>
<PE>ie allergies/colds</PE>
<MED>also,i believe zyrtec and antihistimines can be and should be
taken before bedtime to eliminate daytime drowsiness.</MED>
<MED>Try vitamin c drops (also aids throat dryness) as a supplement.
Vitamin C can also be found in red peppers.
Peppers can clear passageways i heard in an article recently.</MED>

Figure 1: Example of PE and MED labelings

We thus frame the problem of extracting medical case descriptions as extracting sentences that describe any of these two aspects. Specifically, the task is to identify sentences in each of the two related categories (i.e., PE and MED) from forum posts. As an extraction task, this task is "shallower" than conventional information extraction tasks such as entity extraction in the sense that we extract a sentence as a unit, which makes the extraction task more tractable. Indeed, the task is more similar to sentence categorization. However, it also differs from a regular sentence categorization task (e.g., sentiment analysis) in that the multiple categories are usually closely related and categorization of multiple sentences may be dependent in the sense that knowing the category of one sentence may influence our decision about the category of another sentence nearby. For example, knowing that a sentence is in the category PE should increase our belief that the next sentence is of category of PE or MED.

We solve the problem using two popular machine learning methods, Support Vector Machines (SVM) and Conditional Random Fields (CRF). We define and study a large set of features, including two kinds of novel features: (1) novel features based on semantic generalization of terms, and (2) novel features specific to forums.

Since this is a novel task, there is no existing data set that we can use for evaluation. We thus create a new data set for evaluation. Experiment results show that both groups of novel features are effective and can improve extraction accuracy. With the best configurations, we can obtain an accuracy of up to 75%, demonstrating feasibility of automatic extraction of medical case descriptions from forums.

2 Related work

Medical data mining has been looked atleast since the early 2000s. Cios and Moore (2002) emphasize the uniqueness of medical data mining. They stress that data mining in medicine is distinct from that in other fields, because the data are heterogeneous, and special ethical, legal, and social constraints apply to private medical information. Treatment recommendation systems have been built that use the structured data to diagnose based on symptoms (Lazarus et al., 2001) and recommend treatments. Holt et al.(2005) provide references to medical systems that use case based reasoning methodologies for medical diagnosis. Huge amounts of medical data stored in clinical data warehouses can be used to detect patterns and relationships, which could provide new medical knowledge (Lazarus et al., 2001). In contrast, we look at the problem of converting some of the unstructured medical text data present in forum threads into structured symptoms and treatments. This data can then be used by all of the above mentioned applications.

Structuring of unstructured text has been studied by many works in the literature. Automatic information extraction (Aone and Ramos-Santacruz, 2000; Buttler et al., 2001) and wrapper induction techniques have been used for structuring web data. Sarawagi (2008) and Laen-

der et al. (2002) offer comprehensive overviews of information extraction and wrapper induction techniques respectively. The main difference between our work and main stream work on extraction is that we extract sentences as units, which is shallower but presumably more robust. Heinze et al. (2002) state that the current state-of-the-art in NLP is suitable for mining information of moderate content depth across a diverse collection of medical settings and specialties. Zhou et al. (2006), the authors perform information extraction from clinical medical records using a decision tree based classifier using resources such as WordNet [1], UMLS [2] etc. They extract past medical history and social behaviour from the records.

In other related works, sentiment classification (Pang et al., 2002; Prabowo and Thelwall, 2009; Cui et al., 2006; Dave et al., 2003) attempts to categorize text based on polarity of sentiments and is often applied at the sentence level (Kim and Zhai, 2009). Some work has also been done on extracting content from forum data. This includes finding question answer pairs (Cong et al., 2008) from online forums, auto-answering queries on a technical forum (Feng et al., 2006), ranking answers (Harabagiu and Hickl, 2006) etc. To the best of our knowledge, this is the first work on shallow extraction from medical forum data.

3 Problem formulation

Let $P = (s_1, ...s_n)$ be a sequence of sentences in a forum post. Given a set of interesting categories $C = \{c_1, ..., c_k\}$ that describe a medical case, our task is to extract sentences in each category from the post P. That is, we would like to classify each sentence s_i into one of the categories c_i or *Background*, which we treat as a special category meaning that the sentence is irrelevant to our extraction task. Depending on specific applications, a sentence may belong to more than one category.

In this paper, we focus on extracting sentences of two related categories describing a medical case: (1) Physical Examination (PE), which includes sentences describing the condition of a patient (i.e., roughly symptoms) (2) Medications (MED), which includes sentences mentioning medications (i.e., roughly treatment). These sentences provide a basic description of a medical case and can already be very useful if we can extract them.

We chose to analyze at the sentence level because a sentence provides enough context to detect the category accurately. For example, detecting the categories at word level will not help us to mark a sentence like *"I get very uncomfortable after eating cheese"* as PE or mark a sentence like *"It's best to avoid cheese in that case"* as MED. Here the problem is loosely represented by a combination of *"uncomfortable eating cheese"* and the solution is represented loosely by *"avoid cheese"*. Indeed, in preliminary analysis, we found that most of the times, the postings consist of PE and MED type sentences.

4 Methods

We use SVMs and CRFs to learn classifiers to solve our problem. SVMs represent approaches that solve the problem as a classification/categorization task while CRFs solve the problem as a sequence labeling task. In this section, we provide the basics of SVMs and CRFs.

4.1 Support Vector Machines

SVM first introduced in (Boser et al., 1992), are a binary classifier that constructs a hyperplane which separates the training instances belonging to the two classes. SVMs maximize the separation margin between this hyperplane and the nearest training datapoints of any class. The larger the margin, the lower the generalization error of the classifier. SVMs have been used to classify both linearly and non-linearly seperable data, and have been shown to outperform other popular classifiers like decision trees, Naïve Bayes classifiers, k-nearest neighbor classifiers, etc. We use SVMs as a representative classifier that does not consider dependencies between the predictions on multiple sentences.

4.2 Conditional Random Fields

Each of the sentences in the postings can itself contain features which help us to categorize it.

[1] http://wordnet.princeton.edu/

[2] http://www.nlm.nih.gov/research/umls

Besides this, statistical dependencies exist between sentences. Intuitively, a MED sentence will follow a PE sentence with high probability, but the probability of a PE sentence following an MED sentence would be low. Conditional random fields are graphical models that can capture such dependencies among input sentences. A CRF model defines a conditional distribution $p(y|x)$ where y is the predicted category (label) and x is the set of sentences (observations). CRF is an undirected graphical model in which each vertex represents a random variable whose distribution is to be inferred, and each edge represents a dependency between two random variables. The observation x can be dependent on the current hidden label y, previous n hidden labels and on any of the other observations in a n order CRF. CRFs have been shown to outperform other probabilistic graphical models like Hidden Markov Models (HMMs) and Maximum Entropy Markov Models (MeMMs). Sutton and McCallum (2006) provide an excellent tutorial on CRFs.

5 Features

To perform our categorization task, we use the following features.

- **Word based features**: This includes unigrams, bigrams and trigrams in the current sentence. Each of the n-grams is mapped to a separate boolean feature per sentence where value is 1 if it appears in sentence and 0 otherwise.
- **Semantic features**: This includes Unified Medical Language System (UMLS[3]) semantic groups of words in the current sentence. UMLS is a prominent bio-medical domain ontology. It contains approximately a million bio-medical concepts grouped under 135 semantic groups. MMTX[4] is a tool that allows mapping of free text into UMLS concepts and groups. We use these 135 semantic groups as our semantic features. In order to generate these features, we first process this sentence through MMTX API which provides all the semantic groups that were found in the sentence. Each of the semantic groups becomes a boolean feature.
- **Position based features**: We define two types of position based features: position of the current sentence in the post and position of the current post in the thread. These features are specific to the forum data. We include these features based on the observations that first post usually contains condition related sentences while subsequent posts often contain treatment measures for the corresponding condition. Each of the position number of a sentence in a post and a post in a thread is mapped to a boolean feature which gets fired for a sentence at a particular position. E.g. For a sentence at position i in a post, POSITION_IN_POST_i would be set to 1 while other features POSITION_IN_POST_j where $j \neq i$ would be set to 0.
- **User based features**: We include a boolean feature which gets fired when the sentence is a part of a post by the thread creator. This feature is important because most of the posts by a thread creator have a high probability of being a PE.
- **Tag based features(Edge features)**: We define features on tags (PE/MED/Backgnd) of previous two sentences to capture local dependencies between sentences. E.g., a set of medication related tags often follow a description of a condition. We use these features only for CRF based experiments.
- **Morphological features**: These include one boolean feature each for presence of
  - a capitalized word in the sentence
  - an abbreviation in the sentence
  - a number in the sentence
  - a question mark in the sentence
  - an exclamation mark in the sentence
- Length based features: We also consider the number of words in a sentence as a separate type of feature. Feature LENGTH_i becomes true for a sentence containing i words.

[3] http://www.nlm.nih.gov/research/umls/
[4] http://mmtx.nlm.nih.gov/

Category	Labeler 1	Labeler 2
PE	513	517
MED	286	280
Background	695	697

Table 1: Labeling results

6 Experiments

6.1 Dataset

Evaluation of this new extraction task is challenging as no test set is available. To solve this problem, we opted to created our own test set. HealthBoards[5] is a medical forum web portal that allows patients to discuss their ailments. We scraped 175 posts contained in 50 threads on allergy i.e., an average of 3.5 posts per thread and around 2 posts per user with a maximum of 9 posts by a particular user. Two humans were asked to tag this corpus as conditions (i.e., PE category) or treatments (i.e., MED category) or none on a per sentence basis. The corpus consists of 1494 sentences. Table 1 shows the labeling results. The data set is available at (http://timan.cs.uiuc.edu/downloads.html). Also the labeling results match quite well (82.86%) with a Kappa statistic value of 0.73. Occasionally (around 3%) PE and MED both occur in the same sentence and the labelers chose to mark such sentences as PE. In the case when the two labelers disagree, we manually analyzed the results and further chose one of them for our experiments.

6.2 Evaluation methodology

For evaluation, we use 5-fold cross validation. For CRFs, we used the Mallet[6] toolkit and for SVM, we used SVM-Light[7]. We experimented by varying the size of the training set, with different feature sets, using two machine learning models: SVMs and CRFs. Our aim is to accurately classify any sentence in a post as PE or MED or background. First we explore and identify the feature sets that help us in attaining higher accuracy. Next, we identify the setting (sequence labeling by CRFs or independent classification by SVMs) that works better to model our problem. We present most of our results using four metrics: precision, recall, F1 measure and average accuracy which is the ratio of correctly labeled sentences to the total sentences.

We considered the following features: all the 2647 words in the vocabulary (no stop-word removal or any other type of selection), 10858 bigrams, 135 semantic groups from UMLS, two position based features, one user based feature, two tag based features, four morphological features and one length based feature as described in the previous section. Thus our feature set is quite rich. Note that other than the usual features, semantic, position-based and user-based features are specific to the medical domain or to forum data.

6.3 Basic Results

First we considered word features, and learned a linear chain CRF model. We added other sets of features one by one, and observed variations in accuracy. Table 2 shows the accuracy in terms of precision, recall and F1. Note that these results are for an Order 1 linear-chain CRF. Accuracy is measured as ratio of the number of correct labelings of PE, MED and background to the total number of sentences in our dataset. Notice that the MED accuracy values are in general quite low compared to those of PE. As we will discuss later, accuracy is low for MED because our word-based features are not discriminative enough for the MED category.

From Table 2, we see that the accuracy keeps increasing as we add semantic UMLS based features, position based features and morphological features. However, length based features (word count), user-based faetures, and bigrams do not result in any improvements. We also tried trigrams, but did not observe any accuracy gains. Thus we find that semantic features and position-based features which are specific to the medical domain and the forum data respectively are helpful when added on top of word features, while generic features such as length-based features tend to not add value.

We also trained an order 2 CRF using the same set of features. Results obtained were similar to order 1 CRFs and so we do not report them here. This shows that local dependencies are more im-

[5]http://www.healthboards.com
[6]http://mallet.cs.umass.edu/
[7]http://svmlight.joachims.org/

Feature set	PE Prec	MED Prec	PE Recall	MED Recall	PE F1	MED F1	Accuracy %
Word	0.60	0.49	0.65	0.36	0.62	0.42	63.43
+Semantic	0.61	0.52	0.68	0.37	0.64	0.43	65.05†
+Position	0.63	0.54	0.7	0.34	0.66	0.42	65.45
+Morphological	0.64	0.52	0.69	0.36	0.66	0.42	65.70
+WordCount	0.62	0.51	0.70	0.33	0.66	0.40	65.23
+Thread Creator	0.62	0.51	0.71	0.34	0.66	0.41	65.49
+Bigrams	0.62	0.51	0.69	0.34	0.66	0.41	64.82

Table 2: Order 1 Linear Chain CRF. †Improvement over only word features significant at 0.05-level, using Wilcoxon's signed-rank test

portant in medical forum data and global dependencies do not add further signal.

Further, we perform experiments using SVMs using the same set of features. Table 3 shows accuracy results on SVM. Again PE is detected with higher accuracy compared to MED. Unlike CRFs, SVMs do not incorporate the notion of local dependencies between sentences. However, we observe that SVMs outperform CRFs, as is evident from the results in Table 3. This is interesting, since it suggests that the SVM accuracy can potentially be further enhanced by incorporating such dependency information (e.g. in the form of new features). We leave this as part of future work.

Figure 2 shows an example of a forum post (which talks about allergy to dogs) being tagged using our CRF model.

<BKG>lori-lynn , </BKG>
<PE>you said he does well with the poms , but you also said he takes shots, so i wondered if the shots were for dog allergies</PE>
<PE>a lot of his friends have dogs , though , and he ' s so very allergic that he has trouble at their homes .</PE>
<MED>we opted not to go with the shots . </MED>
<BKG>i ' m still a little leary about adopting a dog . </BKG>
<BKG>i would just hate it if we did have reactions , because i know we ' d bond with the dog very quickly . </BKG>

Figure 2: Tagging example of a forum post

6.4 Feature selection

Incremental addition of different feature types did not lead to substantial improvement in performance. This suggests that none of the feature classes contains all "good" features. We therefore perform feature selection based on information gain and choose the top 4253 features from among all the features discussed earlier, based on a threshold for the gain. This results in improvement in the accuracy values over the previous best results (Table 4).

Among the word feature set, we found that important features were *allergy, alergies, food, hives, allergic, sinus, bread.* Among bigrams, *allergic_to, ear_infections, my_throat, are_allergic, to_gluten, food_allergies* have high information gain values. Among the UMLS based semantic groups, we found that *patf (Pathologic Function), dsyn (Disease or Syndrome), orch (Organic Chemical), phsu (Pharmacologic Substance), sosy (Sign or Symptom)* have high information gain values. Also looking at the word count feature, we notice that background sentences are generally short sentences. All these features are clearly highly discriminative.

6.5 Variation in training data size

We varied the amount of training data used for learning the models to observe the variation in performance with size of training data. Table 5 shows the variation in accuracy (PE F1, MED F1 and average accuracy) for different sizes of training data using CRFs. In general, we observe that accuracy improves as we increase the training data, but the degree varies with the feature sets used. We see similar trends in SVM also. These results show that it is possible to further improve prediction accuracy by obtaining additional training data.

6.6 Probing into the low MED accuracy

As observed in Tables 2 and 3, MED accuracy is quite low compared to PE accuracy. We wish to gain a deeper insight into why the MED accuracy suffers. Therefore, we plot the frequency of words in sentences marked as PE or MED versus the rank of the word as shown in the figure 3. We removed the stop words. Observe that for PE the curve is quite steep. This indicates that there

Feature set	PE Prec	MED Prec	PE Recall	MED Recall	PE F1	MED F1	Accuracy %
Word	0.65	0.52	0.71	0.28	0.68	0.36	66.13
+Semantic	0.73	0.54	0.73	0.38	0.73	0.45	71.02†
+Position	0.71	0.52	0.71	0.35	0.71	0.42	69.61
+Morphological	0.72	0.53	0.72	0.38	0.72	0.44	70.28
+WordCount	0.74	0.54	0.72	0.37	0.73	0.44	71.55
+Thread Creator	0.74	0.56	0.72	0.39	0.73	0.46	72.02
+Bigrams	0.75	0.54	0.72	0.40	0.74	0.46	71.69

Table 3: SVM results. †Improvement over only word features significant at 0.05-level, using Wilcoxon's signed-rank test

Classifier	PE Prec	PE Recall	PE F1	MED Prec	MED Recall	MED F1	Accuracy %
SVM (all* features)	0.72	0.53	0.72	0.38	0.72	0.44	70.28
SVM (selected features)	0.75	0.75	0.75	0.61	0.33	0.44	75.08†
CRF (all* features)	0.64	0.52	0.69	0.36	0.66	0.42	65.70
CRF (selected features)	0.60	0.77	0.67	0.58	0.37	0.45	65.93†

Table 4: Accuracy using the best feature set. (*Word +Semantic +Position +Morphological features). †Improvement over all* features significant at 0.05-level, using Wilcoxon's signed-rank test

are some discriminative words which have very high frequency and so the word features observed in the training set also get fired for sentences in the test set with high probability. While for MED, we observe that most of the words have very low frequencies. This basically means that discriminative words for MED may not occur with good enough frequency. So, many of the word features that show up in the training set may not appear in the test data. Hence, MED accuracy suffers.

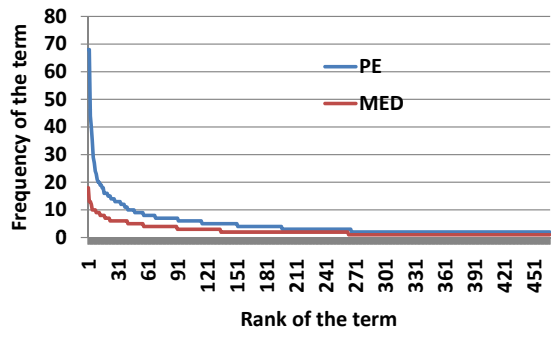


Figure 3: Freq of words vs rank for PE and MED

6.7 Multi-class vs Single class categorization

Note that our task is quite different from plain sentence categorization task. We observe that there is a dependence between the categories (PE/MED) that we are trying to predict per sentence. For example, considering 100% training data, Table 6 compares the precision, recall and F1 values when

	PE	MED	Backgnd	EOP
PE	0.54	0.13	0.28	0.05
MED	0.15	0.51	0.30	0.04
Backgnd	0.18	0.08	0.54	0.20
BOP	0.40	0.07	0.53	0.0

Table 7: Transition probability values

SVM and CRF are trained as single class classifiers using word+semantic features with the multi-class results obtained previously. Results are generally better when we do multi-class categorization versus single-class categorization. This trend was reflected for other featuresets also.

6.8 Analysis of transition probabilities

Table 7 shows the transition probabilities from one category to another as calculated based on our labelled dataset. BOP is beginning of posting and EOP is end of posting. Note that posts often start with a PE or a background sentence and often end with a background sentence. Also, consecutive sentences within a posting tend to belong to the same category.

6.9 Error analysis

We also perform some error analysis on results using the best feature set. Table 8 shows the confusion matrix for CRF/SVM. We observe many of the MED errors are because an MED sentence often gets marked as PE. This basically happens because some sentences contain both PE and MED.

Feature set	25%	50%	75%	100%
Word	0.59/0.21/0.57	0.6/0.36/0.60	0.61/0.39/0.62	0.62/0.42/0.63
+Semantic	0.61/0.17/0.59	0.63/0.32/0.61	0.64/0.38/0.63	0.64/0.43/0.65
+Position	0.59/0.18/0.56	0.64/0.29/0.60	0.65/0.33/0.62	0.66/0.42/0.65
+Morphological	0.6/0.19/0.57	0.64/0.32/0.61	0.65/0.37/0.63	0.66/0.42/0.65
Best	0.61/0.18/0.65	0.66/0.28/0.64	0.66/0.38/0.66	0.69/0.43/0.68

Table 5: Precision, recall, and F value for various sizes of training data set.

Classifier Type	PE Prec	PE Recall	PE F1	MED Prec	MED Recall	MED F1
SVM PE vs BKG	0.79	0.64	0.71	-	-	-
SVM MED vs BKG	-	-	-	0.6	0.28	0.39
SVM Multi-class	0.73	0.73	0.73	0.54	0.38	0.45
CRF PE vs BKG	0.68	0.64	0.66	-	-	-
CRF MED vs BKG	-	-	-	0.53	0.3	0.39
CRF Multi-class	0.61	0.68	0.64	0.52	0.37	0.43

Table 6: Multi-class vs Single-class categorization with word+semantic features

	PE	MED	Backgnd
PE	424/404	37/37	81/101
MED	102/70	107/95	81/125
Backgnd	164/62	55/21	618/754

Table 8: Confusion matrix showing counts of actual vs predicted labels for (Best CRF Classifier/Best SVM Classifier)

Other than that some of the PE keywords are also present in MED sentences, and since the few discriminative MED keywords are quite low in frequency, MED accuracy suffers. E.g. The sentence *"i'm still on antibiotics for the infection but they don't seem to be doing any good anymore."* was labeled as MED but marked as PE by the CRF. The sentence clearly talks about a medication. However, the keyword *"infection"* is often observed in PE sentences and so the CRF marks the sentence as PE.

7 Conclusion

In this paper, we studied a novel shallow information extraction task where the goal is to extract relevant sentences to a predefined set of categories that describe a medical case. We proposed to solve the problem using supervised learning and explored two representative approaches (i.e., CRF and SVM). We proposed and studied two different types of novel features for this task, including generalized terms and forum structure features. We also created the first test set for evaluating this problem. Our experiment results show that (1) the proposed new features are effective for improving the extraction accuracy, and (2) it is feasible to automatically extract medical cases in this way, with the best prediction accuracy above 75%.

Our work can be further extended in several ways. First, since constructing a test set is labor-intensive, we could only afford experimenting with a relatively small data set. It would be interesting to further test the proposed features on larger data set. Second, while in CRF, we have shown adding dependency features improves performance, it is unclear how to evaluate this potential benefit with SVM. Since SVM generally outperforms CRF for this task, it would be very interesting to further explore how we can extend SVM to incorporate dependency.

8 Acknowledgement

We thank the anonymous reviewers for their useful comments. This paper is based upon work supported in part by an IBM Faculty Award, an Alfred P. Sloan Research Fellowship, an AFOSR MURI Grant FA9550-08-1-0265, and by the National Science Foundation under grants IIS-0347933, IIS-0713581, IIS-0713571, and CNS-0834709.

References

Aone, Chinatsu and Mila Ramos-Santacruz. 2000. Rees: a large-scale relation and event extraction system. In *ANLP*.

Boser, Bernhard E., Isabelle Guyon, and Vladimir Vapnik. 1992. A training algorithm for optimal mar-

gin classifiers. In *Computational Learing Theory*, pages 144–152.

Buttler, David, Ling Liu, and Calton Pu. 2001. A fully automated object extraction system for the world wide web. In *ICDCS*.

Cios, Krzysztof J. and William Moore. 2002. Uniqueness of medical data mining. *Artificial Intelligence in Medicine*, 26:1–24.

Cong, Gao, Long Wang, Chin-Yew Lin, Young-In Song, and Yueheng Sun. 2008. Finding question-answer pairs from online forums. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 467–474, New York, NY, USA. ACM.

Cui, Hang, Vibhu Mittal, and Mayur Datar. 2006. Comparative Experiments on Sentiment Classification for Online Product Reviews. In *Proc. of the National Conf. on Artificial Intelligence*, pages 1265–1270.

Dave, Kushal, Steve Lawrence, and David M. Pennock. 2003. Mining the Peanut Gallery: Opinion Extraction and Semantic Classification of Product Reviews. In *Proc. of WWW*, pages 519–528.

Feng, Donghui, Erin Shaw, Jihie Kim, and Eduard Hovy. 2006. An intelligent discussion-bot for answering student queries in threaded discussions. In *IUI '06: Proceedings of the 11th international conference on Intelligent user interfaces*, pages 171–177, New York, NY, USA. ACM.

Harabagiu, Sanda and Andrew Hickl. 2006. Methods for using textual entailment in open-domain question answering. In *ACL-44: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 905–912, Morristown, NJ, USA. Association for Computational Linguistics.

Heinze, Daniel T., Mark L. Morsch, and John Holbrook. 2002. Mining free-text medical records. In *Proceedings of the AMIA Annual Symposium*.

Holt, Alec, Isabelle Bichindaritz, Rainer Schmidt, and Petra Perner. 2005. Medical applications in case-based reasoning. *Knowl. Eng. Rev.*, 20(3):289–292.

Kim, Hyun Duk and ChengXiang Zhai. 2009. Generating comparative summaries of contradictory opinions in text. In *CIKM*, pages 385–394.

Laender, Alberto H. F., Berthier A. Ribeiro-neto, Altigran S. da Silva, and Juliana S. Teixeira. 2002. A brief survey of web data extraction tools. *SIGMOD Record*.

Lazarus, R, K P Kleinman, I Dashevsky, A DeMaria, and R Platt. 2001. Using automated medical records for rapid identification of illness syndromes (syndromic surveillance): the example of lower respiratory infection. *BMC Public Health*, 1:9.

Pang, Bo, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up?: Sentiment Classification using Machine Learning techniques. In *Proc. of EMNLP*, pages 79–86.

Prabowo, Rudy and Mike Thelwall. 2009. Sentiment analysis: A combined approach. *Journal of Informetrics*, 3(2):143–157, April.

Sarawagi, Sunita. 2008. Information extraction. *Foundations and Trends in Databases*, 1.

Sutton, Charles and Andrew Mccallum, 2006. *Introduction to Conditional Random Fields for Relational Learning*. MIT Press.

Zhou, Xiaohua, Hyoil Han, Isaac Chankai, Ann Prestrud, and Ari Brooks. 2006. Approaches to text mining for clinical medical records. In *SAC '06: Proceedings of the 2006 ACM symposium on Applied computing*, pages 235–239, New York, NY, USA. ACM.

Bridging Topic Modeling and Personalized Search

Wei Song Yu Zhang Ting Liu Sheng Li
School of Computer Science
Harbin Institute of Technology
{wsong, yzhang, tliu, lisheng}@ir.hit.edu.cn

Abstract

This work presents a study to bridge topic modeling and personalized search. A probabilistic topic model is used to extract topics from user search history. These topics can be seen as a roughly summary of user preferences and further treated as feedback within the KL-Divergence retrieval model to estimate a more accurate query model. The topics more relevant to current query contribute more in updating the query model which helps to distinguish between relevant and irrelevant parts and filter out noise in user search history. We designed task oriented user study and the results show that: (1) The extracted topics can be used to cluster queries according to topics. (2) The proposed approach improves ranking quality consistently for queries matching user past interests and is robust for queries not matching past interests.

1 Introduction

The majority of queries submitted to search engines are short and ambiguous and the users of search engines often have different search intents even when they submit the same query (Janse and Saracevic, 2000)(Silverstein and Moricz, 1999). The "one size fits all" approach fails to optimize each individual's specific information need. Personalized search has be viewed as a promising direction to solve the "data overload" problem, and aims to provide different search results according to the specific preference of an individual(Pitkow and Breuel, 2002). Information retrieval (IR) communities have developed models for context sensitive search and related applications (Shen and Zhai, 2005a)(White and Chen, 2009).

The search context includes a broad range of information types such as a user's background, his personal desktop index, browser history and even the context information of a group of similar users (Teevan, 2009). In this paper, we exploit the user search history of an individual which contains the past submitted queries, results returned and the click through information. As described in (Tan and Zhai, 2006), search history is one of the most important forms of search context. When dealing with search history, distinguishing between relevant and irrelevant parts is important. The search history may contain a lot of noisy information which can harm the performance of personalization (Dou and Wen, 2007). Hence, we need to sort out relevant and irrelevant parts to optimize search personalization.

In this paper, we propose a topic model based approach to study users' preferences. The main contribution of this work is modeling user search history with topics for personalized search. Our approach mainly consists of two steps: topic extraction and relevance feedback. We assume that a user's search history is governed by the underlying hidden properties and apply probabilistic Latent Semantic Indexing (pLSI) (Hofmann, 1999) to extract topics from user search history. Each topic indexes a unigram language model. We model these extracted topics as feedback in the KL-Divergence retrieval framework. The task is to estimate a more accurate query model based on the evidence from user feedback. We distin-

Coling 2008: Poster Volume, pages 1167–1175,
Beijing, August 2010

guish relevant parts from irrelevant parts in search history by focusing on the relevance between topics and query. The closer a topic is to the current query, the more it contributes in updating the query model, which in turn is used to rerank the documents in results set.

2 Related Work

2.1 Personalized IR

Personalized search is an active ongoing research direction. Based on different representations of user profile, we classify approaches as follows:

Taxonomy based methods: this approach maps user interests to an existing taxonomy. ODP[1] is widely used for this purpose. For example, by exploiting the user search history, (Speretta and Gauch, 2005) modeled user interest as a weighted concept hierarchy created from the top 3 level of ODP. (Havelivala, 2002) proposed the "topic sensitive pagerank" algorithm by calculating a set of PageRanks for each web page on the top 16 ODP categories. (Qiu and Cho, 2006) further improved this approach by building user models from user click history. In recent studies, (Xu S. and Yu, 2008) used ODP categories for exploring folksonomy for personalized search. (Dou and Wen, 2007) proposed a method that represent user profile as a weighting vector of 67 predefined topic categories provided by KDD Cup-2005. Taxonomy based methods rely on a predefined taxonomy and may suffer from the granularity problem.

Content based methods: this category of methods use traditional text presentation model such as vector space model and language model to express user preference. Rich content information such as user search history, browser history and indexes of desktop documents are explored. The user profiles are built in the forms of term vectors or term probability distributions. For example, (Sugiyama and M., 2004) represented user profiles as vectors of distinct terms and accumulated past preferences. (Teevan and Horvitz, 2005) constructed a rich user model based on both search-related information, such as previously issued queries, and other information such as documents and emails a user had read and created. (Shen and Zhai, 2005b) used browsing histories and query sessions to construct short term individual models for personalized search.

Learning to rank methods: (Eugene and Susan, 2005) and (Eugene and Zheng, 2006) incorporated user feedback into the ranking process in a *learning to rank* framework. They leveraged millions of past user interaction with web search engine to construct implicit feedback features. However, this approach aims to satisfy majority of users rather than individuals.

2.2 Probabilistic Topic Models

Probabilistic topic models have become popular tools for unsupervised analysis of document collection. Topic models are based upon the idea that documents are mixtures of topics, where a topic is a probability distribution over words (Steyvers and Griffiths, 2007). These topics are interpretable to a certain degree. In fact, one of the most important applications of topic models is to find out semantic lexicons from a corpus. One of the most popular topic models, the probabilistic Latent Semantic Indexing Model (pLSI), was introduced by Hofmann (Hofmann, 1999) and quickly gained acceptance in a number of text modeling applications. In this study, pLSI is used to discover the underlying topics in user search history. Though pLSI is argued that it is not a complete generative model, we used it because it does not need to generate unseen documents in our case and the model is much easier to be estimated compared with sophisticated models such as LDA(David M. Blei and Jordan, 2003).

2.3 Model based Relevance Feedback

Our work is also related to language model based (pseudo) relevance feedback (Zhai and Lafferty, 2001b) and shares the similar idea with (Tan B. and Zhai, 2007). The differences are: (1) The feedback source is user search history rather than top ranked documents for a query. (2) We make use of user implicit feedback rather than explicit feedback. (3) The topics in search history could be extracted offline and updated periodically. Additionally, these topics provide an informative picture of user search history.

[1] Open Directory Project, http://dmoz.org/

Table 1: An illustration of topics extracted from a user's search history. Terms with highest probabilities are listed below each topic.

Topic 2	Topic 3	Topic 9	Topic 16
climb	movie	swim	cup
0.032	0.091	0.044	0.027
setup	download	ticket	world
0.022	0.078	0.032	0.022
equipment	dvd	notice	team
0.020	0.061	0.019	0.016
practice	watch	travel	brazil
0.009	0.060	0.016	0.011
player	cinema	hotel	storm
0.006	0.038	0.008	0.007

3 Proposed Approach

3.1 Main Idea

A user's search history usually covers multiple topics. It is crucial to distinguish between relevant and irrelevant parts for optimizing personalization. We propose a topic model based method to achieve that goal. First, we construct a document collection revealing user intents according to the user's past activities. A probabilistic topic model is applied on this collection to extract latent topics. Then the extracted topics are used as feedback. The query model is updated by highlighting the topics highly relevant to current query. Finally, the search results are reranked according to the relevance to the updated query model. Table 1 shows 4 topics extracted from a user's search history. Each topic is a unigram language model. The terms with higher probabilities belonging to each topic are listed. We can predict that the user has interests in both *movie* and *football*. However, when the user submits a query about *world cup*, the topic 16 is given higher preference for estimating a more accurate query model.

3.2 Topic Extraction from Search History

Individual's search history consists of all the past query units. Each query unit includes query text, returned search results (with title, snippets and URLs) and click through information. Here, we concatenated the title and snippet of each search result to form a document being considered as a whole. The whole search history can be seen as a collection of documents. Obviously, many documents in the collection may fail to satisfy the user's information need and are uncertain for discovering the user's preferences. Therefore, the first task is to select proper documents in search history as the preference collection for topic discovery.

3.2.1 Preference Collection

An intuitive solution is to use the documents that are clicked by the user. The assumption is that a user clicks on a result only if he is interested in the document. However, user click is sparse in real search environments and the documents not clicked by the user may also be relevant to the user's information need. We assumed that the user had only one search intent for a submitted query. To enhance this coherence within a query unit, we created only one super-document for a query unit as follows: if a query unit had clicked documents, then we concatenated these document to form a preferred document. Otherwise, we selected the top *n* documents from the search results and concatenated them as a preferred document. That is motivated by the idea of pseudo relevance feedback (Lavrenko and Croft, 2001) and used here for alleviating data sparsity. Pseudo relevance feedback is sensitive to the number of feedback documents. In this work, *n* is set to 3, because the average clicks for a query is not more than 3. By this way, we got a preference collection whose size is the same as the number of past queries.

3.2.2 Topic Extraction

Given the collection of preferred documents, we applied pLSI on this collection to extract underlying topics. We define the collection as $C=\{d_1,d_2,\ldots,d_M\}$, where d_i corresponds to the ith query unit, and M is the size of the collection. Each query unit is viewed as a mixture of different topics. It is reasonable in reality. For example, a news document about "*play basketball with obama*" might be seen as a mixture of topics "*politics*" and "*sports*".

Modeling: The basic idea of pLSI is to treat the words in each document as being generated from a mixture model where the component models are topic word distributions. Let k be the num-

ber of topics which is assumed known and fixed. θ_j is the word distribution for topic *j*. We extract topics from collection *C* using a simple probabilistic mixture model as described in (Zhai and Yu, 2004). A word *w* within document *d* can be viewed as generated from a mixture model:

$$p_d(w) = \lambda_B p(w|\theta_B) + (1-\lambda_B)\sum_{j=1}^{k}\pi_{d,j}p(w|\theta_j) \quad (1)$$

where θ_B is the background model for all the documents. The background model is used to draw common words across all the documents and lead to more discriminative and informative topic models, since θ_B gives high weights to non-topical words. λ_B is the probability that a term is generated from the background model which is set to be a constant. To draw more discriminative topic models, we set λ_B to 0.95. Parameter $\pi_{d,j}$ indicates the probability that topic *j* is assigned to the specific document *d*, where $\sum_{j=1}^{k}\pi_{d,j}$=1.

Parameter estimation: The parameters we have to estimate including the background model θ_B, $\{\theta_j\}$ and $\{\pi_{d,j}\}$. θ_B is maximum likelihood estimated (MLE) using all available text in our data set so that it is a fixed distribution. The other parameters to be estimated are $\{\theta_j\}$ and $\{\pi_{d,j}\}$. The log-likelihood of document *d* is:

$$\log p(d) = \sum_{w\in V} c(w,d)\log[\lambda_B p(w|\theta_B) + (1-\lambda_B)\sum_{j=1}^{k}\pi_{d,j}p(w|\theta_j)] \quad (2)$$

The log-likelihood of the whole collection *C* is:

$$\log(C) = \sum_{d\in C}\sum_{w\in V} c(w,d)\log[\lambda_B p(w|\theta_B) + (1-\lambda_B)\sum_{j=1}^{k}\pi_{d,j}p(w|\theta_j)] \quad (3)$$

The Expectation-Maximization (EM) algorithm (Dempster and Rubin, 1977) is used to find a group of parameters maximizing equation (3). The updating formulas are:

E-Step:

$$p(z_{d,w}=B) = \frac{\lambda_B p(w|\theta_B)}{\lambda_B p(w|\theta_B) + (1-\lambda_B)\sum_{j=1}^{k}\pi_{d,j}p(w|\theta_j)}$$

$$p(z_{d,w}=j) = \frac{\pi_{d,j}^{(m)} p^{(m)}(w|\theta_j)}{\sum_{j'=1}^{k}\pi_{d,j'}^{(m)} p^{(m)}(w|\theta_{j'})}$$

M-Step:

$$\pi_{d,j}^{(m+1)} = \frac{\sum_{w\in V} c(w,d)(1-p(z_{d,w}=B))\,p(z_{d,w}=j)}{\sum_{j'=1}^{k}\sum_{w\in V} c(w,d)(1-p(z_{d,w}=B))\,p(z_{d,w}=j')}$$

$$p^{(m+1)}(w|\theta_j) = \frac{\sum_{d\in C} c(w,d)(1-p(z_{d,w}=B))\,p(z_{d,w}=j)}{\sum_{d\in C}\sum_{w'\in V} c(w',d)(1-p(z_{d,w'}=B))\,p(z_{d,w'}=j)}$$

where $c(w,d)$ denotes the number of times *w* occurs in *d*. A hidden variable $z_{d,w}$ is introduced for the identity of each word. $p(z_{d,w}=B)$ is the probability that the word *w* in document *d* is generated by the background model. $p(z_{d,w}=j)$ denotes the probability that the word *w* in document *d* is generated using topic *j* given that *w* is not generated from the background model. Informally, the EM algorithm starts with randomly assigning values to the parameters to be estimated and then alternates between E-Step and M-Step iteratively until it yields a local maximum of the log likelihood.

Interpretation: As shown in equation (1), a word can be viewed as a mixture of topics. From the updating formulas, we can see that the dominant topic of a word depends on both itself and the context. The word tends to have the same topic with the document containing it. While the probability of assigning topic *j* to document *d* is estimated by aggregating all the fractions of words generated by topic *j* in document *d*. We can explain it in a more intuitive way with in our application. As we know, the queries are usually ambiguous. A classic example is *"apple"* which may refer to a kind of fruit, apple Inc, apple electric products, etc. Therefore, it is reasonable to assume that each word belongs to multiple latent semantic properties. If a returned result contains *"apple"* and other words like *"computer"*, *"ipod"* , etc. The word *"apple"* in this result tends to have the same topic distributions with *"computer"* and

'ipod". If the user clicks the result, we can predict that the user's real preference about query *"apple"* is related to electric products having a high probability. Further, if *"apple"* occurs frequently in many documents related to electric products, it obtains a higher probability in this topic. As a result, we not only know user's interest in electric products, but also find a preference to *"apple"* brand.

Since a document's topic depends on the words it contains, two documents with similar word distributions have similar topic distributions. In other words, each topic is like a bridge connecting queries with similar intents. In summary, the topic extraction process plays a role in our application for finding user preference, highlighting discriminative words and connecting queries with similar intents.

3.3 Topics as Feedback

The topics extracted from search history are considered as a kind of feedback. Since topic models actually are extensions of language models, we use such feedback within the KL-Divergence retrieval model (Xu and Croft, 1999)(Zhai and Lafferty, 2001b) that is a principled framework to model feedback in the language modeling approach. In this framework, feedback is treated as updating the query language model based on extra evidence obtained from the feedback sources. The information retrieval task is to rank documents according to the KL divergence $D(\theta_q||\theta_d)$ between a query language model θ_q and a document language model θ_d. The KL divergence is defined as:

$$D(\theta_q||\theta_d) = \sum_{w \in V} p(w|\theta_q) \log \frac{p(w|\theta_q)}{p(w|\theta_d)} \quad (4)$$

where *V* denotes the vocabulary. We estimate the document model θ_d using Dirichlet estimation (Zhai and Lafferty, 2001a):

$$p(w|\theta_d) = \frac{c(w,d) + \mu p(w|\theta_C)}{|d| + \mu} \quad (5)$$

where $|d|$ is document length, $p(w|\theta_C)$ is collection language model which is estimated using the whole data collection. μ is the Dirichlet prior that is set to 20 in this work. The updated query model is defined as:

$$\begin{aligned} p(w|\theta_q) &= \lambda p_{ml}(w|\theta_q) \\ &+ (1-\lambda) \sum_{j=1}^{k} p(w|\theta_j) p(z=j|q) \end{aligned} \quad (6)$$

where $p_{ml}(w|\theta_q)$ is the MLE query model. $\{\theta_j\}$ represents a set of extracted topics each of which is a unigram language model. λ is used to balance the two components. z is a hidden variable over topics. The task is to estimate the multinomial topic distribution $p(z|q)$ for query *q*. Since pLSI does not properly provide a prior, we estimate $p(z=j|q)$ as:

$$\begin{aligned} p(z=j|q) &= \frac{p(q, z=j)}{\sum_{j'=1}^{k} p(q, z=j')} \\ &\propto \frac{sim(\theta_q, \theta_j)}{\sum_{j'=1}^{k} sim(\theta_q, \theta_{j'})} \end{aligned} \quad (7)$$

Since the query text is usually very short, it is not easy to make a decision based on query text alone. Instead, we concatenate all the available documents in returned result set to form a superdocument. A language model is estimated for it. We convert both the document language model and topic models into weighted term vectors and use cosine similarity as the *sim* function. $p(z|q)$ plays an import role here as it determines the contribution of topics. The topics with higher similarity with current query contributes more in updating query model. This scheme helps to filter out noisy information in search history.

4 Evaluation and Discussion

4.1 Data Collection

To the best of our knowledge, there is no public collection with enough content information and user implicit feedback. We decided to carry out a data collection. Due to the difficulty to describe and evaluate user interests implicitly, we predefined some user interests and implemented a search system to collect user interactions.

The predefined interests belong to 5 big categories namely *Entertainment*, *Computer* & *Internet*, *Sports*, *Health* and *Social life*. Each interest is a kind of user preference such as "movies"

Table 2: An example of predefined user interests and tasks

category	Enterntainment
interest	movies
task1	search for a brief introduction of your favorite movie
task2	search for an introduction of an actor or actress you like
task3	search for movies about "artificial intelligence"

Table 3: Statistics of the data collection

user	1	2	3	4	5
#queries	218	256	177	206	311
#big category	5	5	5	5	5
#interest	25	25	25	25	25
#tasks	100	100	100	100	100
avg.#relevant results	4.17	4.22	3.89	4.12	3.24
avg.#clicked results	2.37	2.21	2.71	1.98	2.42

and "outdoor sports". For each interest, we designed several tasks each of which had a goal. Table 2 illustrates an example of a predefined user interest and related tasks. The volunteers were asked to find out the information need according to the tasks. Though we defined these interests and tasks, we did not impose any constraint on the queries. The volunteers could choose and reformulate any query they thought good for finding the desired information. But we did try to increase the possibility that a user might issue ambiguous queries by designing tasks like "search for movies about artificial intelligence" which was categorized to interest "movies", but also related to computer science.

To collect the user interaction with search engine, we implemented a Lucene based search system on Tianwang terabyte corpus(Yan and Peng, 2005). Five volunteers were asked to submit queries to this system to find information satisfying the tasks of each interest. The system recorded users' activities including submitted queries, returned search results (with title, snippet and URL) and users' click through information. When the user finished a task, he clicked a button to tell the system termination of the session containing all the queries and activities related to this task. After finishing all the tasks, the volunteers were asked to judge the top 20 results' relevance (relevant or not relevant) for each query according to the search target. Each volunteer submitted 233 queries on average. Table 3 presents some statistics of this collection.

4.2 Evaluating Topic Extraction

It is not easy to assess the quality of topics, because topic extraction is an unsupervised process and difficult to give a standard answer. Therefore, we view the topic extraction as a clustering problem that is to organize queries into clusters. To group queries into clusters through extracted topics, we use $\hat{j} = \arg\max_j \pi_{d,j}$ to assign a query to the $\hat{j}$th topic. Each topic corresponds to a cluster. All the queries are divided into *k* clusters. Based on the data collection, we setup the golden answers according to the predefined interests. We view all the queries belonging to a predefined interest(which includes multiple tasks) form a cluster which helps us to build a golden answer with 25 clusters in tatal.

One purpose of making use of topics in search history is to find more relevant parts and reduce the noise. We hope that the extracted topics are coherent. That is, a cluster should contain as many queries as possible belonging to a single interest. To evaluate coherence, we adopt *purity* (Zhao and Karypis, 2001), a commonly used metric for evaluating clustering. The higher the purity is, the better the system performs. We compare our method (denoted as PLSI) against the k-means algorithm(denoted as K-Means) on the preference collection.

Figure 1 shows the overall purity with different number of topics. Our method gained better performance than k-means algorithm consistently. It is effective to discover and organize user interests. Besides, as illustrated in Table 1, our method is able to give higher probability to discriminative words of each topic that provides a clear picture of user search history. This leads to an emergence of novel approaches for personalized browsing.

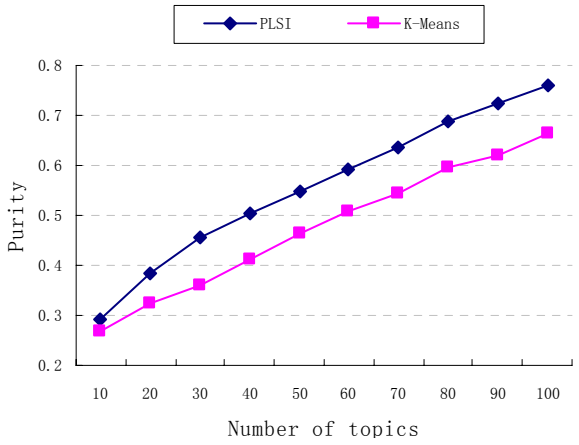


Figure 1: Average purity over 5 users gained by both PLSI and K-Means with different number of topics(clusters).

4.3 Evaluating Result Reranking

4.3.1 Metric

To quantify the ranking quality, the Discounted Cumulative Gain (DCG) (Jarvelin and Kekakainen, 2000) is used. DCG is a metric that gives higher weights to highly ranked documents and incorporates different relevance levels by giving them different gain values.

$$DCG(i) = \begin{cases} G(1), & \text{if } i = 1 \\ DCG(i-1) + \frac{G(i)}{\log(i)}, & \text{otherwise} \end{cases}$$

In our work, we use $G(i) = 1$ for the results labeled as relevant by a user and $G(i) = 0$ for the results that are not relevant. The average normalized DCG (NDCG)over all the test queries is selected to show the performance.

4.3.2 Systems

We evaluated the performance of following systems:

PLSI: The proposed method. The history model was a weighted interpolation over topics extracted from the preference collection described in session 3.2.1.

PSEUDO: From each query unit, we selected top n documents as pseudo feedback. The language history model was estimated on all these documents.

PLSI-PSEUDO: Top n documents from each query unit were concatenated to form a preferred document. The history model was constructed based on topics extracted from these preferred documents.

HISTORY: The history language model was estimated based on all the documents in search history.

TB: It was based on(Tan and Zhai, 2006)which built a unit language model for every past query and the history model was a weighted interpolation of past unit language models.

ORIGINAL: The default search system.

The first 5 systems provided schemes to smooth the query model. They estimated the query models by utilizing different types of feedback (implicit feedback or pseudo feedback) and weighting methods (topic modeling or simple language modeling). The updated query model was an interpolation between MLE query model and history language model. The interpolation parameter was set to 0.5, and n was set to 3.

4.3.3 Performance Comparison

To evaluate the performance on a test query, we focus on two conditions:

1. the test query matches some past interests. We want to check the ability of systems to find relevant information from noisy data.

2. the test query does not match any of past interests. We are interested in the robustness of the systems.

For the first case, the users were asked to select at most 2 queries they submitted for each task. These queries were used as test queries. The other queries were used to simulate the users' search history. In total we got 400 queries for testing. Figure 2 demonstrates the performance of these systems over all test queries. PLSI outperformed all other systems consistently that shows topic model based methods help to estimate a more accurate query model and the user implicit feedback is better evidence. The PLSI-PSEUDO also performed well that indicates the top documents is useful for revealing the topic of queries, even though they do not satisfy user need on occasion. TB also gained better performance than PSEUDO and HISTORY. It indicates

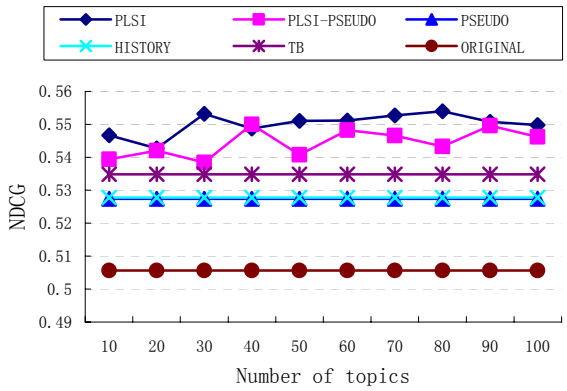


Figure 2: The overall average performance of systems, when each test query matches some user past interests

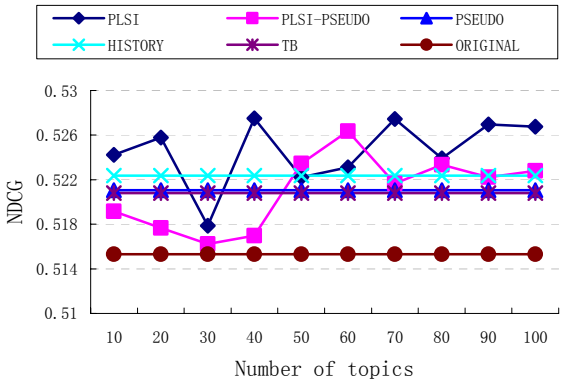


Figure 3: The overall average performance of systems, when each test query does not match any user past interest.

highlighting relevant parts in search history helps to improve the retrieval performance, when the query matches some of user past interests. Compared with default system, both HISTORY and PSEUDO improved a lot which proves that the context in search history is reliable feedback.

For the second case, each user was asked to hold out 5 interests from his collection for testing and the other interests were used as search history. The users selected queries from the held out interests as test queries. These queries did not match each user's past interests. We got 244 test queries. As figure 3 shows, though systems still performed better against ORIGINAL, the improvements were not significant. PLSI still gained the best performance. It has better ability to alleviate the effect of noise. HISTORY and PLSI are more robust than PLSI-PSEUDO which seems sensitive to the number of topics in this case.

In both cases, HISTORY gained moderate performance but quite robust. It is still a very strong baseline, though noisy information is not filtered out. PLSI performed best in both cases. PLSI-PSEUDO outperformed PSEUDO when the test queries matched user past interests and gained comparable results in second case. It shows that modeling user search history as a mixture of topics and weighting topics according to relevance between topics and query help to update a better query model. However, it is necessary to determine if a query matches past interests that helps to optimize personalized search strategies.

5 Conclusion and Future Work

In this paper, we have proposed a topic model based method for personalized search. This approach has some advantages: first, it provides a principled way to combine topic modeling and personalized search; second, it is able to find user preferences in an unsupervised way and gives an informative summary of user search history; third, it explores the underlying relationship between different query units via topics that helps to filter out the noise and improve ranking quality.

In future, we plan to do a large scale study by leveraging the already built search system or business search engines. Also, we will try to add more information to extend the existing model. Besides, it is necessary to design methods for determining whether a submitted query matches the user past interests that is crucial to apply our algorithm adaptively and selectively.

Acknowledgements

This research is supported by the National Natural Science Foundation of China under Grant No. 60736044, by the National High Technology Research and Development Program of China No. 2008AA01Z144, by Key Laboratory Opening Funding of MOE-Microsoft Key Laboratory of Natural Language Processing and Speech, Harbin Institute of Technology, HIT.KLOF.2009020. We thank the anonymous reviewers and Fikadu Gemechu for their useful comments and help.

References

David M. Blei, Andrew Y. Ng and Michael I. Jordan. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.

Dempster, A.P., Laird N.M. and D.B. Rubin. 1977. Maximum likelihood from incomplete data via the em algorithm. *Journal of Royal Statist. Soc. B*, 39:1–38.

Dou, Z., Su R. and J. Wen. 2007. A large-scale evaluation and analysis of personalized search strategies. *Proc. WWW*, pages 581–590.

Eugene, A., Eric B. and D. Susan. 2005. Improving web search ranking by incorporating user behavior information. *Proc.SIGIR*, pages 19–26.

Eugene, A. and Zijian Zheng. 2006. Identifying best bet web search results by mining past user behavior. *Proc.SIGKDD*, pages 902–908.

Havelivala, T.H. 2002. Topic-sensitive pagerank. *Proc. WWW*, pages 517–526.

Hofmann, T. 1999. Probabilistic latent semantic indexing. *Proc.SIGIR*, pages 50–57.

Janse, B.J., Spink A. Bateman J. and T. Saracevic. 2000. Real life, real users, and real needs: a study and analysis of user queries on the web. *Information Processing and Management*, 26(2):207–222.

Jarvelin, K. and J. Kekakainen. 2000. Ir evaluation methods for retrieving highly relevant documents. *Proc.SIGIR*, pages 41–48.

Lavrenko, V. and W. Croft. 2001. Relevance based language models. *Proc.SIGIR*, pages 120–127.

Pitkow, J., Schutze H. Cass T. Cooley R. Turnbull D. Edmonds A. Adar E. and T. Breuel. 2002. Personalized search. *Commun,ACM*, 45(9):50–55.

Qiu, F. and J. Cho. 2006. Automatic identification of user interest for personalized search. *Proc.WWW*, pages 727–736.

Shen, X., Tan B. and C. Zhai. 2005a. Context-sensitive information retrieval using implicit feedback. *Proc. SIGIR*, pages 43–50.

Shen, X., Tan B. and C. Zhai. 2005b. Implicit user modeling for personalized search. *Proc. CIKM*, pages 824–831.

Silverstein, C., Marais H. Henzinger M. and M. Moricz. 1999. Analysis of a very large web search engine query log. *SIGIR Forum*, 33(1):6–12.

Speretta, M. and S. Gauch. 2005. Personalized search based on user search histories. *Proc. WI'05*, pages 622–628.

Steyvers, M. and T. Griffiths. 2007. Probabilistic topic models. *Handbook of Latent Semantic Analysis. Erlbaum, Hillsdale, NJ*.

Sugiyama, K., Hatano K. and Yoshkawa. M. 2004. Personalized search based on user search histories. *Proc. WWW*, pages 675–684.

Tan, B., Shen X. and C. Zhai. 2006. Mining long-term search history to improve search accuracy. *Proc.SIGKDD*, pages 718–723.

Tan B., Atulya Velivelli, Fang H. and C. Zhai. 2007. Term feedback for information retrieval with language models. *Proc.SIGIR*, pages 263–270.

Teevan, J., Dumais S.T. and E. Horvitz. 2005. Personalizing search via automated analysis of interests and activities. *Proc.SIGKDD*, pages 449–456.

Teevan, J., Morris M.R. Bush S. 2009. Discovering and using groups to improve personalization. *Proc.WSDM*, pages 15–24.

White, R.W., Bailey P. and L. Chen. 2009. Predicting user interest from contextual information. *Proc.SIGIR*, pages 363–370.

Xu, Jinxi and W. Croft. 1999. Cluster-based language models for distributed retrieval. *Proc.SIGIR*, pages 254–261.

Xu S., Bao, S. Fei B. Su Z. and Y. Yu. 2008. Exploring folksonomy for personalized search. *Proc.SIGIR*, pages 155–162.

Yan, H., Li J. Zhu j. and B. Peng. 2005. Tianwang search engine at trec 2005: Terabyte track. *Proc.TREC*.

Zhai, C. and J. Lafferty. 2001a. A study of smoothing methods for language models applied to ad hoc information retrieval. *Proc.SIGIR*, pages 334–342.

Zhai, Chengxiang and John Lafferty. 2001b. Model-based feedback in the language modeling approach to information retrieval. *Proc.CIKM*, pages 403–410.

Zhai, C., Velivelli A. and B. Yu. 2004. A cross-collection mixture model for comparative text mining. *Proc.SIGKDD*, pages 743–748.

Zhao, Y. and G. Karypis. 2001. Criterion functions for document clustering: Experiments and analysis. *Technical Report TR #01–40, Department of Computer Science, University of Minnesota, Minneapolis, MN.*

Notes on the Evaluation of Dependency Parsers Obtained Through Cross-Lingual Projection

Kathrin Spreyer
Department of Linguistics
University of Potsdam
spreyer@uni-potsdam.de

Abstract

In this paper we address methodological issues in the evaluation of a projection-based framework for dependency parsing in which annotations for a source language are transfered to a target language using word alignments in a parallel corpus. The projected trees then constitute the training data for a data-driven parser in the target language. We discuss two problems that arise in the evaluation of such cross-lingual approaches. First, the annotation scheme underlying the source language annotations – and hence the projected target annotations and predictions of the parser derived from them – is likely to differ from previously existing gold standard test sets devised specifically for the target language. Second, the standard procedure of cross-validation cannot be performed in the absence of parallel gold standard annotations, so an alternative method has to be used to assess the generalization capabilities of the projected parsers.

1 Introduction

The manual annotation of treebanks for natural language parsing is time-consuming and expensive, but the availability of such resources is crucial for data-driven parsers, which require large amounts of training examples. A technique known as *annotation projection* (Yarowsky and Ngai, 2001) provides a means to relax this resource bottleneck to some extent: In a word-aligned parallel corpus, the text of one language (*source language*, SL), say English, is annotated with an existing parser, and the word alignments are then used to transfer (or *project*) the resulting annotations to the other language (*target language*, TL). The projected trees, albeit noisy, can then constitute the training data for data-driven TL parsers (Hwa et al., 2005; Spreyer and Kuhn, 2009). Finally, in order to assess the quality of the projected parser, its output needs to be compared to held-out TL test data.

Two problems arise in the evaluation of such approaches. First, the annotations projected from the SL usually differ stylistically from those found in the TL test data, rendering any immediate comparison between the predictions of the projected parser and the gold standard meaningless. We discuss the use of tree transformations for evaluation purposes, namely to consolidate discrepancies between the annotation schemes. We then present experiments that investigate the influence of the annotation scheme used in training on the generalization capabilities of the resulting parser. We also briefly address the interaction between annotation style and parsing algorithm (transition-based vs. graph-based).

The second problem addressed here is the assessment of variance in the training data, and hence in parser quality. The standard procedure for this purpose would be *cross-validation*. However, the popular data sets used for benchmarking parsers, such as those that emerged

Coling 2008: Poster Volume, pages 1176–1184,
Beijing, August 2010

from the CoNLL-X shared task on dependency parsing (Buchholz and Marsi, 2006), are typically based on monolingual text. This means that cross-validation is unavailable for projection-based frameworks, because no projection can be performed for the training splits in the absence of a translation in the SL. We therefore propose a validation scheme which accounts for training data variance by training a parser multiple times, on random samples drawn from the projected training data. Each of the obtained parsers can subsequently be evaluated against a fixed, held-out test set independent of the projection step, and the array of accuracy measurements thus obtained can be further subjected to significance testing to verify that observed performance differences are not merely random effects.

The paper is structured as follows. Section 2 describes the projection framework we are assuming. Section 3 summarizes and contrasts the characteristics of four different annotation schemes underlying our SL parsers (English, German) and TL test data (Dutch, Italian). Experiments with different annotation schemes and parsing algorithms are presented in Section 4. In Section 5 we discuss variance assessment in more detail. Section 6 concludes.

2 The Projection Framework

This section briefly describes how we obtain dependency parsers for new languages via annotation projection in a parallel corpus. A detailed discussion can be found in Spreyer and Kuhn (2009).

We use the Europarl corpus (Koehn, 2005) as our parallel corpus. It comprises parallel data from 11 languages; in this paper, we present experiments with English and German as SLs, and Dutch and Italian as TLs.

First, the bitexts for the language pairs under consideration (English-Dutch, English-Italian, German-Dutch, and German-Italian) are word-aligned using Giza++ (Och and Ney, 2003), and all texts are part-of-speech tagged with the TreeTagger (Schmid, 1994) according to pre-trained models.[1]

[1]Available from `http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/DecisionTreeTagger.html`.

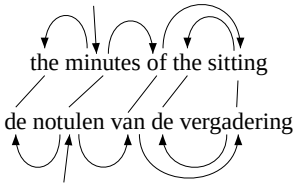


Figure 1: Dependency tree projection from English to Dutch.

Second, we annotate the SL portions, i.e., the German and English texts, with MaltParser dependency parsers (Nivre et al., 2006) trained on standard data sets for the two languages; specifically, we are using the baseline parsers of Øvrelid et al. (2010). The English training data consists of the Wall Street Journal sections 2–24 of the Penn Treebank (Marcus et al., 1993), converted to dependencies (Johansson and Nugues, 2007). The treebank data used to train the German parser is the Tiger Treebank (Brants et al., 2002), in the version released with the CoNLL-X shared task (Buchholz and Marsi, 2006).

Given the SL dependency trees, we project the dependencies to the corresponding (i.e., aligned) TL elements as shown in Figure 1. The links between the English and Dutch words indicate the word alignment. We postulate edges between TL words (e.g., *de* and *notulen*) if there is an edge between their respective SL counterparts (*the* and *minutes*).

The projected dependencies are then used as training data for TL (Dutch and Italian) dependency parsers. In order to account for the fact that many of the projected dependency structures are incomplete due to missing alignments or non-parallelism of the translation, we employ fMalt (Spreyer and Kuhn, 2009), a modified version of the MaltParser which handles fragmented training data. We restrict the admissible fragmentation to three fragments per sentence, for sentences with four or more words, based on early experiments with automatically labeled Dutch data. Sentences that receive more fragmented analyses are discarded.

Finally, we evaluate the projected TL parsers against gold standard test sets by parsing the TL test data and comparing the parser output to

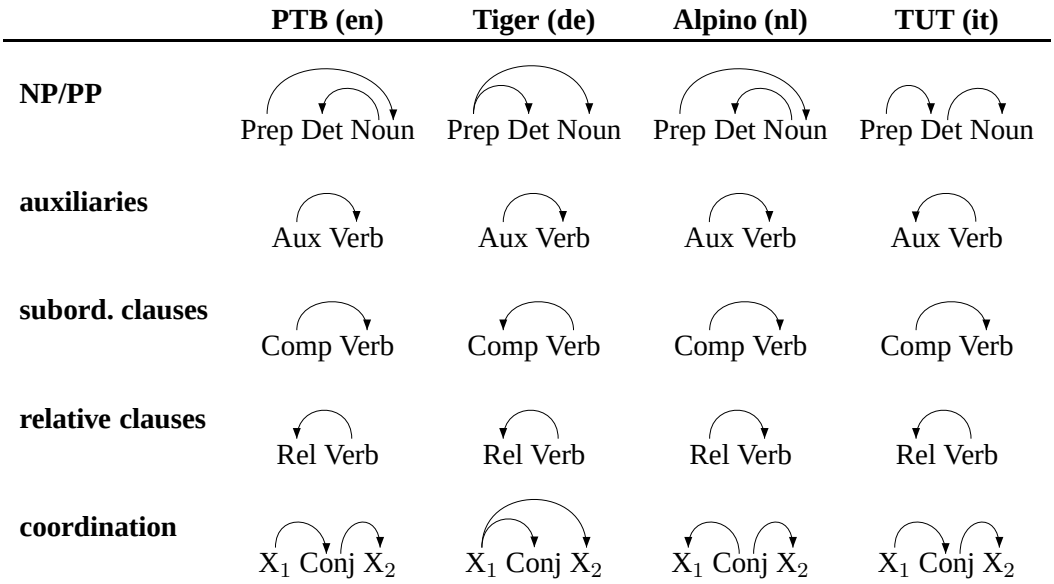


Table 1: Different annotation schemes in dependency-converted treebanks.

the reference annotations. However, we discuss below how differences in annotation style prohibit a direct comparison, and how the annotation schemes affect the learnability of the grammar and therefore the accuracy of the derived parsers.

3 Annotation Schemes

In a projection setting like the one described above, we deal with two sets of annotations: those projected from the SL, and those marked up in the TL gold standard. The four annotation schemes we compare here are those used in the Penn Treebank (PTB; WSJ sections) (Marcus et al., 1993) for English, the Tiger Treebank (Brants et al., 2002) for German, the Alpino Treebank (van der Beek et al., 2002) for Dutch, and the Turin University Treebank[2] (TUT) for Italian.

Table 1 illustrates the most obvious differences among the annotation schemes. Note that we compare annotations in the dependency-converted format. This restricts the comparison to attachment decisions and eliminates the bracket bias inherent to constituent-based comparisons (Carroll et al., 1998; Rehbein and van Genabith, 2007). Again, we use the dependency-converted data sets of the CoNLL-X shared task.

As shown in the table, both the English and the Dutch treebank annotate prepositional phrases hierarchically, with an embedded NP. The flat annotation scheme of the German treebank, on the other hand, makes every word in the PP a dependent of the preposition (with some exceptions). The Italian annotation scheme assumes a hierarchical structure like English and Dutch, but declares the determiner rather than the noun as the head of nominal phrases. Another idiosyncrasy of the Italian annotation scheme is the treatment of fused prepositions such as *della* which incorporate the determiner of the embedded NP: In the dependency-converted TUT, these fused prepositions are represented as two separate tokens, one tagged as a preposition, the other as a determiner.

Next, auxiliaries take the lexical verb as their dependent in all treebanks except the Italian TUT, which inverts the dependency, resulting in a flat structure with the lexical verb as its head. The structure of subordinate clauses is hierarchical according to the English, Dutch and Italian annotation schemes, but flat in Tiger, with the complementizer as a dependent of the embedded verb. Relative clauses, on the other hand, are assigned a flat structure in all but the Dutch scheme, where the relativizer is the head of the embedded verb. Finally, coordination is annotated in three different ways: While the treebanks for English and Italian implement a strictly right-branching strat-

[2] http://www.di.unito.it/~tutreeb

egy, the German annotation scheme attaches both the conjunction and the second conjunct to the first conjunct. The Dutch treebank annotates coordinations as flat structures, with all conjuncts depending on the conjunction.

In order to evaluate projected parsers, any differences in the source and target annotations need to be consolidated. A straightforward way of doing so is by means of tree transformations. Naturally, this begs the question of where such transformations should take place: One could transform the projected annotations to conform to the reference annotations encountered in the test set; alternatively, one can manipulate the test set to reflect the annotation decisions adopted in the source annotations. A variant of the former approach has been implemented by Hwa et al. (2005). They apply post-projection transformations to Chinese training data projected from English in order to infuse TL-specific information which has no counterpart in the source language.

We argue in favor of the alternative, since in a practical application scenario, where rapid, inexpensive development plays a prominent role, it is conceivable that the SL annotation scheme would be adopted unaltered for the TL parser. Consider, for instance, an architecture for multilingual syntax-based information retrieval which is based on parsers for various TLs, all to be derived from a single SL. Devising a tailored annotation scheme for each of the TLs would require linguistically trained personnel with extensive knowledge of the languages at hand. By contrast, adhering to the SL annotation scheme results in homogeneous parser output across the TLs and thus facilitates streamlined higher-level processing.

In Section 4 we present experiments that involve the language pairs English–Dutch, German–Dutch, English–Italian, and German–Italian. For each of the TLs Dutch and Italian, we therefore derive transformed test sets for each SL: one version according to the English PTB annotation style to evaluate the parsers projected from English, and another version according to the German Tiger-style annotations to evaluate parsers projected from German. As an example, Table 2 illustrates the transformations performed on the Italian test set for the parser projected from English.

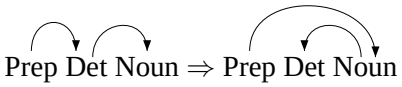
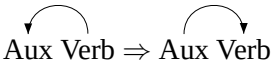
TUT (it) ⇒ PTB (en)	
NP/PP	
auxiliaries	
fused prepositions	

Table 2: Transformations performed on the Italian test set for the parser projected from English.

a.

lang	orig	PTB	Tiger
nl	–	**69.21**	67.38
it	–	**66.44**	53.09

b.

lang	orig	PTB	Tiger
nl	79.23	**80.79**	79.19
it	**88.52**	86.88	84.02

Table 3: Unlabeled attachment scores obtained by training MaltParsers on (a) projected and (b) gold standard dependencies according to different annotation schemes.

English.

4 Annotation Scheme Experiments

4.1 Learnability

If the annotation style is carried over from the source language as we suggest above, we may ask: Is one annotation scheme more appropriate than the other? When more than one source language (annotation scheme) is available, will one produce more "learnable" TL annotations than the other? We explore these questions experimentally. Table 3a shows the performance of Dutch ('nl') and Italian ('it') MaltParsers trained on annotations projected from English ('PTB') and German ('Tiger'), as evaluated against the respective transformed Dutch and Italian gold standards.

Looking at the results for Dutch, we find that there is indeed a significant difference between the parser projected from English and the one projected from German. The former, generating PTB-style dependencies, achieves 69.21% unla-

lang.	words/sent	words/frag	frags/sent
en→nl	27.83	1.95	14.25
de→nl	27.55	1.98	13.92
en→it	28.86	2.26	12.79
de→it	28.79	1.66	17.33

Table 4: Average fragmentation in the projected dependencies.

parser	orig	PTB	Tiger
MST	81.41	**83.01**	**83.87**
		Tiger ≈ *PTB* > *orig*	
Malt	79.23	**80.79**	79.19
		PTB > *orig* > *Tiger*	

Table 5: UAS of the Dutch MST parsers trained on gold standard dependencies. (MaltParser results repeated from Table 3b.)

beled attachment score (UAS). According to a t-test (cf. Section 5), this is significantly ($p<0.01$) better than the parser projected from German Tiger-style annotations, which achieves 67.38%.

Turning to Italian, the parser projected from the English PTB-style annotations again performs better. However, the huge difference of 13.35% UAS suggests a more fundamental underlying problem with the word alignment between the German and Italian sentences. And indeed, inspection of the degree of fragmentation in the Italian projected dependencies (Table 4) confirms that considerably more edges are missing in the dependencies projected from German than from English. Missing edges are an indication of missing word alignment links.

In order to control such factors and focus only on the learnability of the different annotation schemes, we report in Table 3b the results of training on gold standard monolingual treebank data (distinct from the test data), transformed – like the test sets – to conform with the English and German annotation scheme, respectively.[3] In addition, the column labeled 'orig' shows the performance obtained when the original (dependency-converted) Alpino/TUT annotation scheme is used. For Italian, the results corroborate those obtained with the projected parsers: training on the PTB-transformed treebank is significantly[4] ($p<0.01$) more effective than training on the Tiger-transformed treebank. The original TUT scheme is even more effective ($p<0.01$), which comes as no surprise given that the TUT guidelines were tailored to the traits of the Italian language.

The Dutch parser, too, responds better to the hierarchical PTB-based annotation scheme than to the flat Tiger scheme ($p<0.01$). In fact, it also outperforms the parser trained with the original Alpino annotations ($p<0.01$). This demands for further investigation, reported in the following section.

4.2 Interaction with Parsing Algorithms

The results in Table 3 affirm that the performance of a parser hinges on the annotation scheme that it is trained on. However, the learnability of a given scheme depends not only on the annotation decisions, but also on the parsing algorithm implemented by the parser. For instance, it has been noted (Joakim Nivre, p.c. 2008) that flat coordination structures like those in the Alpino Treebank generally pose a challenge to incremental, deterministic parsers like MaltParser.

In order to see to what extent our results are influenced by characteristics of the MaltParser, we repeated the experiments with the MST parser (McDonald et al., 2005), focusing on Dutch parsers from gold standard training data.[5]

The MST parser is a graph-based dependency parser which considers all possible edges to find the globally optimal tree. The results of the MST experiments are given in Table 5, together with the corresponding Malt results repeated from Table 3b. We observe that the relative learnability ranking among the three annotation schemes is indeed different with MST. While in the transition-based paradigm the original Alpino annotations still appeared more adequate for training than the

[3] We did not attempt parameter optimization, so the figures reported here do *not* represent the state-of-the-art in dependency parsing for either language.

[4] According to Dan Bikel's Randomized Parsing Evaluation Comparator: `http://www.cis.upenn.edu/~dbikel/software.html#comparator`

[5] With projected training data for Dutch, and in all experiments with Italian, MST produced the same pattern of relative performance as Malt.

trans	Malt	MST
none	79.23	81.41
$\text{coordination}_{\text{en}}$	**80.91**	**83.01**
$\text{relative}_{\text{en}}$	79.21	81.81
all_{en}	**80.79**	**83.01**
$\text{coordination}_{\text{de}}$	79.39	82.19
$\text{relative}_{\text{de}}$	79.21	81.81
$\text{subord}_{\text{de}}$	79.47	**82.67**
np/pp_{de}	**80.73**	**83.83**
all_{de}	79.19	**83.87**

Table 6: Impact of individual transformations on Dutch treebank parsers. Significant improvements ($p<0.01$) over original Alpino annotation ('none') are in bold face.

Tiger trees, it is now outperformed by both the PTB and the Tiger trees under the graph-based approach. There is no significant difference between the Tiger-based and the PTB-based parser.

To shed some light on the unexpected ranking of the Alpino annotation scheme, we look at the impact of the individual transformations separately in Table 6. The upper part of the table shows how the transformations of the Alpino data towards PTB-style annotations affects learnability. We find that both the MaltParser and the MST parser benefit from the right-branching coordination markup of the PTB scheme. The attachment of relativizers in relative clauses seems to play only a minor role and makes no significant difference.

Turning to the Tiger-style transformations, first note that the semi-flat coordination adopted in the German treebank does not seem to be superior to the flat annotations in Alpino: no significant improvement is achieved for either parser by using the former ('$\text{coordination}_{\text{de}}$'). Surprisingly, both parsers benefit from the flat annotation of prepositional phrases ('np/pp_{de}'). The MST parser, but not the MaltParser, further takes advantage of the flat subordination structure annotated in Tiger. As mentioned earlier, this is in line with the fundamentally different parsing paradigms represented by Malt and MST.

We tentatively conclude that the MST parser is in fact better at exploiting the flat aspects of the Tiger annotations, while both parsers largely benefit from the highly hierarchical coordination structure of the PTB annotation scheme. A more detailed exploration of these issues is clearly in order, and subject to future research.

4.3 Discussion

Kübler et al. (2008) present an extensive comparison of two German treebanks: the Tiger treebank with its rather flat annotation scheme, and the TüBa/DZ treebank with more hierarchical structures. They find that the flat Tiger annotation scheme is more easily learned by constituent-based (PCFG) parsers when evaluated on a dependency level. Our results suggest the opposite, but this may well be due to the differences in the experimental setup: Our training data represent dependency trees directly, and we learn incremental, deterministic dependency parsers rather than PCFGs.

5 Variance Assessment

The second question we address in this paper is the assessment of variance in the training data, and hence in parser quality. The standard procedure for this purpose would be *cross-validation*. To perform k-fold cross-validation, the data is partitioned into k splits of equal size, and one of the splits is used as test data, while the remaining k-1 splits serve as training data. The train–test cycle is repeated until each of the k subsamples has been used as test data exactly once.

However, the popular data sets used for benchmarking parsers, such as the CoNLL-X shared task data used here, are typically based on monolingual text. This means that cross-validation is unavailable for projection-based frameworks, because no projection can be performed for the training splits in the absence of a translation in the SL.

Moreover, the expected noise level in the projected dependencies requires that there be a considerable amount of training data for an evaluation to be meaningful. So even if parallel test data is available, the data partitioning performed in cross-validation may compromise the results.

We therefore propose a validation scheme which (i) does not reduce the amount of test data by partitioning (this may be a problem when only a small number of gold standard annotations is

	nl_{ptb}	nl_{tig}	it_{ptb}	it_{tig}
	68.51	67.25	66.56	54.01
	70.07	66.79	66.45	54.21
	69.21	68.13	66.07	53.37
	69.45	68.29	66.47	52.77
	68.47	67.31	66.74	52.55
	69.07	66.97	66.20	53.66
	69.99	67.87	66.56	52.70
	69.71	66.43	66.37	52.70
	68.77	67.11	66.05	52.08
	68.83	67.67	66.96	52.82
mean	69.21	67.38	66.44	53.09
sd	0.58	0.60	0.29	0.69

Table 7: Intra-system variance assessment.

available), (ii) does not require parallel test data and is independent of the projection step, and (iii) takes advantage of the fact that training data is cheap and therefore abundant in projection-based settings. Specifically, given that we have plenty of training data, we can train a particular parser multiple (say, k) times, each time sampling a fixed number of training examples from the pool of training data. The k parsers can then each parse the unseen test set, and subsequent comparison against the gold standard annotations yields k values of the performance metric at hand (here, UAS). As in conventional cross-validation, these k values are then averaged to provide an aggregated score, and they can be used to derive standard deviations etc. The arrays of measurements for different systems can further be subjected to significance tests such as the two-sample t-test to verify that observed performance differences are not merely random effects.

5.1 Experiments

We use the validation procedure just described (with k=10) to investigate the variance in the projected parsers discussed in the previous section (Table 3a). Table 7 lists the scores obtained by the individual parsers, each trained on a different random sample of 100,000 words, drawn from the pool of all projected annotations. We also show the standard deviation and repeat the mean UAS. We observe that, for a given language, standard deviation seems to correlate negatively with mean UAS; in other words, the better parsers also seem to be more robust towards variance in the training data.

5.2 Discussion

Classical cross-validation and the validation method described here do measure slightly different things. First, in cross-validation it is not only the training data that is varied, but the test data as well. Second, when two systems are compared under the cross-validation regime, the k rounds can usually be considered *paired* samples because both systems are trained and evaluated on identical partitionings of the data. In contrast, projection-based settings typically involve some form of filtering on the basis of the projected annotations; in our case, the filter restricts the degree of fragmentation in the projected dependency tree. This filtering makes it all but impossible to pair the training samples without seriously diminishing the pool from which the samples are drawn. For instance, when comparing the Italian parser projected from English (it_{ptb}) and the one projected from German (it_{tig}), a training sentence may receive a complete analysis from the English translation, and hence be included in the training pool for it_{ptb}; but the same (Italian) sentence may receive a highly fragmented analysis under projection from German (e.g., due to missing alignment links) and be discarded from the training pool for it_{tig}.

With samples that cannot be paired, it is also not obvious how evaluation strategies like the randomized comparison mentioned above (fn. 4) could be employed in a sound way (by non-statisticians).

6 Conclusions

We have discussed two issues that arise in the evaluation of frameworks that involve cross-lingual projection of annotations. We focused on the projection of dependency trees from German and English to Dutch and Italian, and presented experiments that compare parsers trained on the projected dependencies. The parsers differ in the annotation scheme they follow: When they are projected from German, they employ the flat Tiger annotation scheme of the source language; pro-

jected from English, they learn the more hierarchical PTB structures. In order to evaluate the projected parsers against target language (Dutch, Italian) gold standard annotations, we convert the test sets to the annotation scheme employed in the respective source language.

While our experiments with gold standard treebank data affirm that the annotation scheme that is being learned has some influence on the performance of the parser, one should bear in mind that in a projection scenario, the quality of the word alignment plays at least an equally important role when it comes to chosing a suitable source language and annotation scheme.

We have further proposed a validation scheme which unlike cross-validation does not require parallel test data. Instead, it exploits the fact that training data is usually available in abundance in projection scenarios, so parsers can be trained on multiple random samples and evaluated against a single, independent test set which need not be further partitioned.

Acknowledgments

The work reported in this paper was supported by the Deutsche Forschungsgemeinschaft (DFG; German Research Foundation) in the SFB 632 on Information Structure, project D4 (Methods for interactive linguistic corpus analysis).

References

Brants, Sabine, Stefanie Dipper, Silvia Hansen, Wolfgang Lezius, and George Smith. 2002. The TIGER treebank. In *Proceedings of the Workshop on Treebanks and Linguistic Theories*, pages 24–41.

Buchholz, Sabine and Erwin Marsi. 2006. CoNLL-X Shared Task on Multilingual Dependency Parsing. In *Proceedings of CoNLL-X*, pages 149–164, New York City, June.

Carroll, John, Ted Briscoe, and Antonio Sanfilippo. 1998. Parser evaluation: a survey and a new proposal. In *Proceedings of LREC 1998*, pages 447–454, Granada, Spain.

Hwa, Rebecca, Philip Resnik, Amy Weinberg, Clara Cabezas, and Okan Kolak. 2005. Bootstrapping parsers via syntactic projection across parallel texts. *Natural Language Engineering*, 11(3):311–325.

Johansson, Richard and Pierre Nugues. 2007. Extended constituent-to-dependency conversion for English. In Nivre, J., H.-J. Kaalep, and M. Koit, editors, *Proceedings of NODALIDA 2007*, pages 105–112.

Koehn, Philipp. 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. In *Proceedings of the MT Summit 2005*.

Kübler, Sandra, Wolfgang Maier, Ines Rehbein, and Yannick Versley. 2008. How to Compare Treebanks. In *Proceedings of LREC 2008*, pages 2322–2329.

Marcus, Mitchell P., Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics*, 19(2):313–330.

McDonald, Ryan, Fernando Pereira, Kiril Ribarov, and Jan Hajič. 2005. Non-projective dependency parsing using spanning tree algorithms. In *Proceedings of HLT-EMNLP 2005*.

Nivre, Joakim, Johan Hall, Jens Nilsson, Gülşen Eryiğit, and Svetoslav Marinov. 2006. Labeled pseudo-projective dependency parsing with support vector machines. In *Proceedings of CoNLL-X*, pages 221–225, New York City, June.

Och, Franz Josef and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Øvrelid, Lilja, Jonas Kuhn, and Kathrin Spreyer. 2010. Cross-framework parser stacking for data-driven dependency parsing. To appear in TAL 2010 special issue on Machine Learning for NLP 50(3), eds. Isabelle Tellier and Mark Steedman.

Rehbein, Ines and Josef van Genabith. 2007. Treebank annotation schemes and parser evaluation for German. In *Proceedings of EMNLP-CoNLL 2007*, pages 630–639, Prague, Czech Republic, June. Association for Computational Linguistics.

Schmid, Helmut. 1994. Probabilistic part-of-speech tagging using decision trees. In *International Conference on New Methods in Language Processing*, pages 44–49, Manchester, England.

Spreyer, Kathrin and Jonas Kuhn. 2009. Data-driven dependency parsing of new languages using incomplete and noisy training data. In *Proceedings of CoNLL 2009*, pages 12–20, Boulder, CO, June.

van der Beek, Leonoor, Gosse Bouma, Robert Malouf, and Gertjan van Noord. 2002. The Alpino dependency treebank. In *Computational Linguistics in the Netherlands (CLIN)*.

Yarowsky, David and Grace Ngai. 2001. Inducing Multilingual POS Taggers and NP Bracketers via Robust Projection across Aligned Corpora. In *Proceedings of NAACL 2001*, pages 200–207.

Dependency-Based Bracketing Transduction Grammar for Statistical Machine Translation

Jinsong Su, Yang Liu, Haitao Mi, Hongmei Zhao, Yajuan Lü, Qun Liu
Key Laboratory of Intelligent Information Processing
Institute of Computing Technology
Chinese Academy of Sciences
{sujinsong,yliu,htmi,zhaohongmei,lvyajuan,liuqun}@ict.ac.cn

Abstract

In this paper, we propose a novel *dependency-based bracketing transduction grammar* for statistical machine translation, which converts a source sentence into a target dependency tree. Different from conventional bracketing transduction grammar models, we encode target dependency information into our lexical rules directly, and then we employ two different maximum entropy models to determine the reordering and combination of partial dependency structures, when we merge two neighboring blocks. By incorporating dependency language model further, large-scale experiments on Chinese-English task show that our system achieves significant improvements over the baseline system on various test sets even with fewer phrases.

1 Introduction

Bracketing transduction grammar (BTG) (Wu, 1995) is an important subclass of synchronous context free grammar, which employs a special synchronous rewriting mechanism to parse parallel sentence of both languages.

Due to the prominent advantages such as the simplicity of grammar and the good coverage of syntactic diversities in different language pairs, BTG has attracted increasing attention in statistical machine translation (SMT). In flat reordering model (Wu, 1996; Zens et al., 2004), which assigns constant reordering probabilities depending on the language pairs, BTG constraint proves to be very effective for reducing the search space of phrase reordering. To pursue a better method to predict the order between two neighboring blocks[1], Xiong et al. (2006) present an enhanced BTG with a maximum entropy (ME) based reordering model. Along this line, source-side syntactic knowledge is introduced into the reordering model to improve BTG-based translation (Setiawan et al., 2007; Zhang et al., 2007; Xiong et al., 2008; Zhang and Li, 2009). However, these methods mainly focus on the utilization of source syntactic knowledge, while ignoring the modeling of the target-side syntax that directly influences the translation quality. As a result, how to obtain better translation by exploiting target syntactic knowledge is somehow neglected. Thus, we argue that it is important to model the target-side syntax in BTG-based translation.

Recently, modeling syntactic information on the target side has progressed significantly. Depending on the type of output, these models can be divided into two categories: the *constituent-output* systems (Galley et al., 2006; Zhang et al., 2008; Liu et al., 2009) and *dependency-output* systems (Eisner, 2003; Lin, 2004; Ding and Palmer, 2005; Quirk et al., 2005; Shen et al., 2008). Compared with the *constituent-output* systems, the *dependency-output* systems provide a simpler platform to capture the target-side syntactic information, while also having the best interlingual phrasal cohesion properties (Fox, 2002). Typically, Shen et al. (2008) propose a string-to-dependency model, which integrates the target-side well-formed dependency structure into translation rules. With the dependency structure, this system employs a dependency language model (LM) to exploit long distance word relations, and achieves a significant improvement over the hierarchical phrase-based system (Chiang, 2007). So

[1] A block is a bilingual phrase without maximum length limitation.

Coling 2008: Poster Volume, pages 1185–1193,
Beijing, August 2010

we think it will be a promising way to integrate the target-side dependency structure into BTG-based translation.

In this paper, we propose a novel dependency-based BTG (DepBTG) for SMT, which represents translation in the form of dependency tree. Extended from BTG, our grammars operate on two neighboring blocks with target dependency structure. We integrate target syntax into bilingual phrases and restrict target phrases to the well-formed structures inspired by (Shen et al., 2008). Then, we adopt two ME models to predict how to reorder and combine partial structures into a target dependency tree, which gives us access to capturing the target-side syntactic information. To the best of our knowledge, this is the first effort to combine the translation generation with the modeling of target syntactic structure in BTG-based translation.

The remainder of this paper is structured as follows: In Section 2, we give brief introductions to the bases of our research: BTG and dependency tree. In Section 3, we introduce DepBTG in detail. In Section 4, we further illustrate how to create two ME models to predict the reordering and dependency combination between two neighboring blocks. Section 5 describes the implementation of our decoder. Section 6 shows our experiments on Chinese-English task. Finally, we end with a summary and future research in Section 7.

2 Background

2.1 BTG

BTG is a special case of synchronous context free grammar. There are three rules utilized in BTG:

$$A \rightarrow [A^1, A^2] \quad (1)$$

$$A \rightarrow \langle A^1, A^2 \rangle \quad (2)$$

$$A \rightarrow x/y \quad (3)$$

where the reordering rules (1) and (2) are used to merge two neighboring blocks A^1 and A^2 in a straight or inverted order, respectively. The lexical rule (3) is used to translate the source phrase x into the target phrase y.

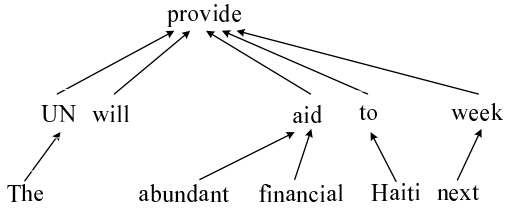


Figure 1: The dependency tree for sentence *The UN will provide abundant financial aid to Haiti next week.*

2.2 Dependency Tree

In a given sentence, each word depends on a parent word, except for the root word. The dependency tree for a given sentence reflects the long distance dependency and grammar relations between words. Figure 1 shows an example of a dependency tree, where a black arrow points from a child word to its parent word.

Compared with constituent tree, dependency tree directly models semantic structure of a sentence in a simpler form. Thus, it provides a desirable platform for us to utilize the target-side syntactic knowledge.

3 Dependency-based BTG

3.1 Grammars

In this section, we extend the original BTG into DepBTG. The rules of DepBTG, which derive from that of BTG, merge blocks with target dependency structure into a larger one. These rules take the following forms:

$$A_d \rightarrow [A_d^1, A_d^2]_{CC} \quad (4)$$

$$A_d \rightarrow [A_d^1, A_d^2]_{LA} \quad (5)$$

$$A_d \rightarrow [A_d^1, A_d^2]_{RA} \quad (6)$$

$$A_d \rightarrow \langle A_d^1, A_d^2 \rangle_{CC} \quad (7)$$

$$A_d \rightarrow \langle A_d^1, A_d^2 \rangle_{LA} \quad (8)$$

$$A_d \rightarrow \langle A_d^1, A_d^2 \rangle_{RA} \quad (9)$$

$$A_d \rightarrow x/y \quad (10)$$

where A_d^1 and A_d^2 represent two neighboring blocks with target dependency structure. Rules (4)$\sim$(9) are used to determine the reordering and combination of two dependency structures, when

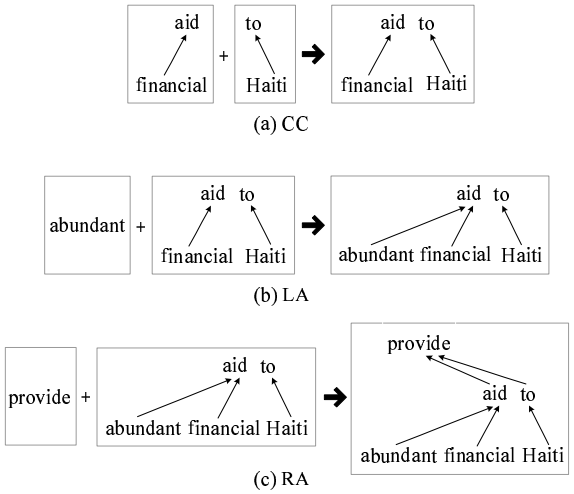


Figure 2: Dependency operations on the neighboring dependency structures. CC = coordinate concatenate, LA = left adjoining, and RA = right adjoining.

we merge two neighboring blocks. Rule (10) is applied to generate bilingual phrase (x, y) with target dependency structure learned from training corpus. To distinguish the rules with different functions, the rules (4)∼(9) and rule (10) are named as **merging rules** and **lexical rule**, respectively.

Specifically, we first merge the neighboring blocks in the straight order using rules (4)∼(6) or in the inverted order using rules (7)∼(9). Then, according to different merging rules, we conduct some operations to combine the corresponding dependency structures in the target order: coordinate concatenate (**CC**), left adjoining (**LA**) and right adjoining (**RA**).

To clearly illustrate our operations, we show the process of applying three dependency operations to build larger structures in Figure 2. Adopting rule (4), the dependency structures "*((financial) aid)*"[1] and "*(to (Haiti))*" can be combined into a larger one consisting of two sibling subtrees (see Figure 2(a)). Adopting rule (5), we can adjoin the left dependency structure "*(abundant)*" to the leftmost sub-root of the right dependency structure "*((financial) aid) (to (Haiti))*" (see Figure 2(b)). Adopting rule (6), we can include the right dependency structure "*((abundant) (financial) aid) (to (Haiti))*" as a child of the rightmost sub-root of the left dependency structure "*(provide)*" (see Figure 2(c)). In a similar way, rules (7)∼(9) are applied to deal with two partial structures in the inverted order.

[1]We use the lexicon dependency grammar (Hellwig, 2006) to express the projective dependency tree. Using this grammar, the words in the brackets are defined as the child words depending on the parent word outside the brackets.

3.2 Well-Formed Dependency Structures

As illustrated in the previous sub section, the rules of DepBTG operate on the blocks with target dependency structure. Following (Shen et al., 2008), we restrict the target phrases to the well-formed dependency structures. The main difference is that we use more relaxed constraints to extract more bilingual phrases with rational structure. Take a sentence $S = w_1 w_2 ... w_n$ for example, we denote the parent word ID of word w_i with d_i, and show the definitions of structures as follows.

Defination 1 *A dependency structure $d_{i...j}$ is* **fixed on head** *h, where $h \in [i, j]$, if and only if it meets the following conditions*

- $d_h \notin [i, j]$
- $\forall k \in [i, j]\ and\ k \neq h, d_k \in [i, j]$
- $\forall k \in [i, j],\ d_k = h\ or\ d_k \in [i, j]$

Defination 2 *A dependency structure $d_i ... d_j$ is* **floating with children** C, *for a non-empty set $C \subseteq \{i...j\}$, if and only if it meets the following conditions*

- $\exists h \notin [i, j], s.t. \forall k \in C, d_k = h$
- $\forall k \in [i, j]\ and\ k \notin C, d_k \in [i, j]$
- $\forall k \notin [i, j],\ d_k \notin [i, j]$ $\underline{or\ d_k = c_l}$ $\underline{or\ d_k = c_r}$

where c_l and c_r represent the IDs of the leftmost and rightmost words in the set *C*, respectively. Note that the underline indicates the difference between our definition and that of (Shen et al., 2008). In our model, we regard the floating structure, which is not complete on its boundary sub-roots, as an useful structure, since it will become a complete constituent by combining it with other partial structures. For example, the dependency

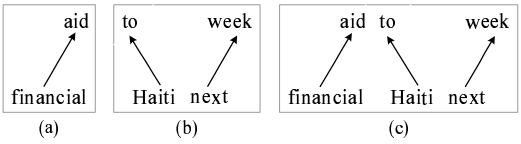


Figure 3: (a) A fixed structure and (b) (c) two floating structures. Note that (c) is ill-formed in (Shen et al., 2008).

structures shown in Figure 3 are all well-formed structures. However, according to the definitions of (Shen et al., 2008), 3(c) is ill-formed because *aid* does not include its leftmost child word *abundant* in the structure.

4 ME Models for Merging Rules

4.1 The Models

A simple way to estimate the probabilities of the merging rules is to adopt maximum likelihood estimation to obtain the conditional probabilities. However, this method is not applicable to merging rules because the dependency structures become larger and larger during decoding, which are very sparse in the corpus.

Inspired by MEBTG translation (Xiong et al., 2006), which considers phrase reordering as a classification problem, we model the reordering and combination of two neighboring dependency structures based on the ME principle. Owing to data sparseness and the complexity of multi-class classification, we establish two ME models rather than an unified ME model: one for the reordering between blocks, called **reordering model**; the other for the dependency operations on the corresponding dependency structures, called **operation model**.

Thus, according to the ME scheme, we decompose the probability Ω of each merging rule into

$$\begin{aligned}\Omega &= p_{\theta_1}(o|A_d^1, A_d^2) \cdot p_{\theta_2}(d|A_d^1, A_d^2) \\ &= \frac{\exp(\Sigma_i \theta_{1i} h_{1i}(o, A_d^1, A_d^2))}{\Sigma_o \exp(\Sigma_i \theta_{1i} h_{1i}(o, A_d^1, A_d^2))} \cdot \\ &\quad \frac{\exp(\Sigma_j \theta_{2j} h_{2j}(d, A_d^1, A_d^2))}{\Sigma_d \exp(\Sigma_j \theta_{2j} h_{2j}(d, A_d^1, A_d^2))}\end{aligned}$$

where the functions $h_{1i} \in \{0, 1\}$ are the features of the ME-based reordering model, θ_{1i} are the corresponding weights, and $o \in \{straight, inverted\}$. Similarly, the functions $h_{2j} \in \{0, 1\}$ and the weights θ_{2j} are trained for the ME-based operation model, and $d \in \{CC, LA, RA\}$.

4.2 Example Extraction

To train the ME models, we extract examples from a string-to-dependency word-aligned corpus during the process of bilingual phrases extraction (Koehn et al., 2005), and then collect various features for the models.

For the reordering model, we adopt the method of (Xiong et al., 2006) to extract reordering examples. Due to the limit of space, we skip the details of this method.

For the operation model, given an operation training example consisting of two neighboring dependency structures: the left structure d_l and the right structure d_r, we firstly classify it into different categories by the dependency relation between d_l and d_r:

- if d_l and d_r have the same parent, the category of the example is CC;
- if d_l depends on the leftmost sub-root of d_r, the category of the example is LA;
- if d_r depends on the rightmost sub-root of d_l, the category of the example is RA.

For instance, Figure 4 shows an operation example with RA operation, where the sub-root word *week* of d_r depends on the rightmost sub-root word *provide* of d_l.

Then, we collect various features from the following nodes: the rightmost sub-root of d_l, and its rightmost child node; the leftmost sub-root of d_r, and its leftmost child node. Here, we speculate that these nodes may carry useful information for the dependency combination of the two structures, since they locate nicely at the boundary subtrees of d_l and d_r. For simplicity, we refer to these nodes as the *feature nodes* of the example. Let's revisit Figure 4, the feature nodes of the example are marked with dashed ellipses. The rightmost sub-root word of d_l is *provide*, and its rightmost child word is *to*; The leftmost sub-root word of d_r is *week*, and its leftmost child word is *next*.

Type	Name	Description
Lexical Features	$W_{lh}(d_r)$	The leftmost sub-root word of d_r
	$W_{rh}(d_l)$	The rightmost sub-root word of d_l
	$W_{llc}(d_r)$	The leftmost child word of $W_{lh}(d_r)$
	$W_{rrc}(d_l)$	The rightmost child word of $W_{rh}(d_l)$
POS Features	$P_{lh}(d_r)$	The POS of $W_{lh}(d_r)$
	$P_{rh}(d_l)$	The POS of $W_{rh}(d_l)$
	$P_{llc}(d_r)$	The POS of $W_{llc}(d_r)$
	$P_{rrc}(d_l)$	The POS of $W_{rrc}(d_l)$

Table 1: Feature categories in the ME-based operation model.

Type	Features and Instances
Unigram Features	$W_{rh}(d_l)$ = provide $W_{rrc}(d_l)$ = to $W_{lh}(d_r)$ = week $W_{llc}(d_r)$ = next $P_{rh}(d_l)$=VV $P_{rrc}(d_l)$ = TO $P_{lh}(d_r)$ = NN $P_{llc}(d_r)$ = ADJ
Bigram Features	$W_{rh}(d_l)\_W_{lh}(d_r)$ = provide_week $W_{rh}(d_l)\_P_{lh}(d_r)$ = provide_NN $P_{rh}(d_l)\_W_{lh}(d_r)$ = VV_week $P_{rh}(d_l)\_P_{lh}(d_r)$ = VV_NN
	$W_{rh}(d_l)\_W_{llc}(d_r)$ = provide_next $W_{rh}(d_l)\_P_{llc}(d_r)$ = provide_ADJ $P_{rh}(d_l)\_W_{llc}(d_r)$ = VV_next $P_{rh}(d_l)\_P_{llc}(d_r)$ = VV_ADJ
	$W_{rrc}(d_l)\_W_{lh}(d_r)$ = to_week $W_{rrc}(d_l)\_P_{lh}(d_r)$ = to_NN $P_{rrc}(d_l)\_W_{lh}(d_r)$ = TO_week $P_{rrc}(d_l)\_P_{lh}(d_r)$ = TO_NN

Table 2: ME operation features and instances of the example shown in Figure 4.

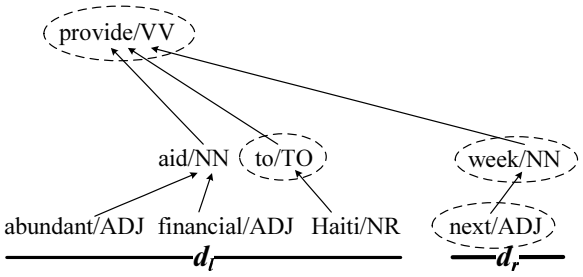


Figure 4: An example with RA category consisting of the neighboring dependency structures d_l and d_r. The dashed ellipses denote the *feature nodes* of the example, and each node consists of one word and its corresponding POS tag.

In addition, to keep the number of operation examples acceptable, we follow (Xiong et al., 2006) to only extract the smallest one from the examples with the same *feature nodes* in each sentence.

4.3 Features

To capture reordering information, we use the boundary words of bilingual blocks as features, which are proved to be very effective in (Xiong et al., 2006).

To capture dependency operation information, we design two kinds of features on the *feature nodes*: the Lexical features and Parts-of-speech (POS) features. With the POS features, the operation ME model will do exact predicating to the best of its ability, and then can back off to approximately predicating if exact predicating fails. Table 1 shows these feature categories in detail.

Furthermore, we also use some bigram features, since it is generally admitted that the combination of different features can lead to better performance than unigram features. To better understand our operation features, we continue with the example shown in Figure 4, listing features and instances in Table 2.

5 Implementation Details

5.1 Decoder

We develop a CKY-style decoder which uses the following features: (1) Phrase translation probabilities in two directions, (2) Lexical translation probabilities in two directions, (3) N-gram LM

score, (4) ME-based reordering model score, (5) Number of phrases, (6) Number of target words, (7) ME-based operation model score, (8) Dependency LM scores at word level and POS level separately, and (9) Discount on ill-formed dependency structures. Here, the former six features are also used in MEBTG translation.

5.2 Dependency Language Model

Following (Shen et al., 2008), we apply different tri-gram dependency LMs at word level and POS level separately to DepBTG translation.

Given a dependency structure, where w_h is the parent word, $w_L = w_{l_1}...w_{l_n}$ and $w_R = w_{r_1}...w_{r_m}$ are child word sequences on the left side and right side respectively, the probability of a tri-gram is computed as follows:

$$\begin{aligned} & P(w_L, w_R|w_h\text{-as-head}) \\ = \; & P(w_L|w_h\text{-as-head}) \cdot P(w_R|w_h\text{-as-head}) \end{aligned}$$

Here $P(w_L|w_h\text{-as-head})$ can be decomposed into:

$$\begin{aligned} & P(w_L|w_h\text{-as-head}) \\ = \; & P(w_{l_1}|w_h\text{-as-head}) \cdot P(w_{l_2}|w_{l_1}, w_h\text{-as-head}) \\ & ... \cdot P(w_{l_n}|w_{l_{n-1}}, w_{l_{n-2}}) \end{aligned}$$

where '-as-head' is used to distinguish the head word from child word in the language model. In like manner, $P(w_R|w_h\text{-as-head})$ has a similar calculation method.

5.3 Ill-Formed Dependency Structure

To preserve the good coverage of bilingual phrases, we keep some bilingual phrases with the special ill-formed dependency structure. Different from the well-formed structures, where all the children of the sub-roots are complete, these ill-formed structures are not complete on the children of the boundary sub-roots, lacking a well-formed sub structure on the boundary. We consider them as useful structures with gaps, each of which can be combined with some well-formed structures into a larger well-formed one. To reduce the search space, we constrain the number of gap to one on each boundary. During decoding, we directly substitute the gap in a structure with another well-formed structure which has the same direction.

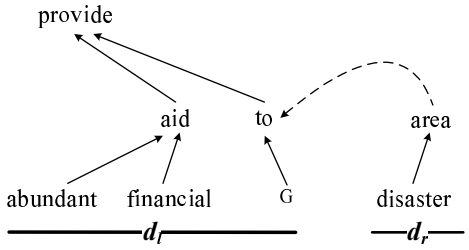


Figure 5: Dependency combination of the ill-formed dependency structure d_l with the right well-formed dependency structure d_r. G denotes gap and the dotted line denotes the substitution of the gap G with d_r.

For example, there are two dependency structures in Figure 5: d_l is an ill-formed structure with a right gap, and d_r is a well-formed one. Instead of investigating three operations to combine these structures, we fill the gap of d_l with d_r, and then compute the corresponding score of the RA operation on the sub structures "*(to)*" and "*((disaster) area)*" in the ME-based operation model.

6 Experiment

6.1 Setup

The training corpus[1] comes from LDC with 1.54M bilingual sentences (41M Chinese words and 48M English words). We run GIZA++ (Och and Ney, 2000) to obtain word alignments with the heuristic method "grow-diag-final-and". Then we parse the English sentences to generate a string-to-dependency word-aligned corpus using the parser (Huang et al., 2009). From this corpus, we extract bilingual phrases with dependency structure. Here, the maximum length of the source phrase is set to 7. For the n-gram LM, we use SRILM Toolkits (Stolcke, 2002) to train a 4-gram LM on the Xinhua portion of the Gigaword corpus. For the dependency LM, we train different 3-gram dependency LMs at word level and POS level separately on the English side of the training corpus.

During the process of bilingual phrase extraction, we collect the neighboring blocks without

[1]The training corpus consists of six LDC corpora: LDC2002E18, LDC2003E07, LDC2003E14, Hansards part of LDC2004T07, LDC2004T08 , LDC2005T06.

any length limitation to obtain examples for two ME models. For the reordering model, we obtain about $22.6M$ examples with monotone order and $4.8M$ examples with inverted order. For the operation model, we obtain about $5.9M$ examples with CC operation, $14.8M$ examples with LA operation, and $9.7M$ examples with RA operation. After collecting various features from the examples, we use the ME training toolkit developed by Zhang (2004) to train ME models with the following parameters: iteration number i=200 and Gaussian prior g=1.0.

The 2002 NIST MT Evaluation test set is used as the development set. The 2003 and 2005 NIST MT Evaluation test sets are our test sets. We perform the MERT training (Och, 2003) to tune the optimal feature weights on the development set. To run the decoder, we prune the phrase table with b = 100, prune the chart with n = 50, α = 0.1. See (Xiong et al., 2006) for the meanings of these parameters. The translation quality is evaluated by case-insensitive BLEU-4 metric (Papineni et al., 2002), as calculated by mteval-v11b.pl.

6.2 Results

Since (Xiong et al., 2006) has made a deep investigation on the ME-based reordering model, we mainly focus on the study of the ME-based operation model. To explore the utility of the various features in the operation model, we randomly select about $10K$ examples from all the operation examples as held-out data, and use the rest examples as training data. Then, we train the operation models on different feature sets and investigate the performance of models on the held-out data.

Table 3 shows the accuracy rates of the ME operation models using different feature sets. We find that the bigram feature set provides the most persuasive evidences and achieves best performance than other feature sets.

To investigate the influences of various factors on the system performance, we carried out experiments on the NIST Chinese-English task with the following systems:

- MEBTG + all: an MEBTG translation system, which uses all bilingual phrases. It is our baseline system;
- MEBTG + filter1: a baseline system, which uses the bilingual phrases consistent to the well-formed dependency structures by (Shen et al., 2008);
- MEBTG + filter2: a baseline system, which uses the bilingual phrases consistent to our well-formed dependency structures;
- MEBTG + filter3: a baseline system, which uses the bilingual phrases consistent to our well-formed dependency structures and the special ill-formed dependency structures;
- DepBTG + unigram features: a DepBTG system which only uses the unigram features in the ME-based operation model;
- DepBTG + bigram features: a DepBTG system which only uses the bigram features in the ME-based operation model;
- DepBTG + all features: a DepBTG system which uses all features in the ME-based operation model;
- DepBTG + unigram features + dep LMs: a DepBTG system with dependency LMs, where only the unigram features are adopted in the ME-based operation model;
- DepBTG + bigram features + dep LMs: a DepBTG system with dependency LMs, where only the bigram features are adopted in the ME-based operation model;
- DepBTG + all features + dep LMs: a DepBTG system with dependency LMs, where all features are adopted in the ME-based operation model.

Model	Accuracy Rate
lexical features	87.614%
POS features	88.232%
unigram features	90.024%
bigram features	$\mathbf{93.907}\%$
all features	93.290%

Table 3: The accuracy rates of the ME-based operation models on the held-out data set using different feature sets. Unigram features include lexical features and POS features, and bigram features are the combinations of different unigram features.

System	Type	#Bp	MT03	MT05
MEBTG	all(baseline)	$81.4M$	33.41	32.65
	filter1	$27.8M$	32.17($\downarrow$ 1.24)	31.26($\downarrow$ 1.39)
	filter2	$33.7M$	32.77($\downarrow$ 0.64)	31.93($\downarrow$ 0.72)
	filter3	$58.5M$	33.29($\downarrow$ 0.12)	32.71($\uparrow$ 0.06)
DepBTG	unigram features	$59.9M$	33.46($\uparrow$ 0.05)	32.67($\uparrow$ 0.02)
	bigram features	$59.9M$	33.57($\uparrow$ 0.16)	32.89($\uparrow$ 0.24)
	all features	$59.9M$	33.59($\uparrow$ 0.18)	32.86($\uparrow$ 0.21)
	unigram features + dep LMs	$59.9M$	33.90($\uparrow$ 0.49)	33.29($\uparrow$ 0.64)
	bigram features + dep LMs	$59.9M$	**34.18**($\uparrow$ 0.77)	**33.58**($\uparrow$ 0.93)
	all features + dep LMs	$59.9M$	34.10($\uparrow$ 0.69)	33.55($\uparrow$ 0.90)

Table 4: Experimental results on Chinese-English NIST Task.

Experiment results are summarized in Table 4. Our baseline system extracts $81.4M$ bilingual phrases and achieves the BLEU scores of 33.41 and 32.65 on two test sets separately. Adopting the constraint of the well-formed structures by (Shen et al., 2008), we extract $27.8M$ bilingual phrases, which lead to great drops in BLEU score: 1.24 points and 1.39 points on two test sets separately(see Row 3). Using the constraint of our well-formed structures, the number of extracted bilingual phrases is $33.7M$. We observe the similar results that the performance drops 0.64 points and 0.72 points over the baseline system on two test sets, respectively (see Row 4). Furthermore, we add some bilingual phrases with the special ill-formed structure into our phrase table, and the number of the bilingual phrases in use is 58.5M accounting up 71.9% of the full phrases. For two test sets, our system achieves the BLEU scores of 33.29 and 32.71 (see Row 5), which are very close to the scores of baseline system. Those experimental results demonstrate that phrase coverage has a great effect on the system performance and our definitions of the allowed dependency structures are useful to retain rational bilingual phrases.

Then, by employing the ME-based operation model and two 3-gram dependency LMs, the DepBTG system outperforms the MEBTG system in almost all cases. The experimental results indicate that the dependency LMs are more effective than the ME-based operation model for DepBTG system. Especially, using bigram features and dependency LMs, the DepBTG system obtains absolute improvements on two test sets: $\mathbf{0.77}$ BLEU points on NIST03 test set and $\mathbf{0.93}$ BLEU points on NIST05 test set (see Row 10), which are both statistically significant at $p < 0.05$ using the significance tester developed by Zhang et al. (2004).

7 Conclusion and Future Work

In this paper, we propose a novel dependency-based BTG to directly model the syntactic structure of the translation. Using the bilingual phrases with target dependency structure, our system employs two ME models to generate the translation in line with dependency structure. Based on the target dependency structure, our system filters 26.4% bilingual phrases (from $81.4M$ to $59.9M$), captures the target-side syntactic knowledge by dependency language models, and achieves significant improvements over the baseline system.

There is some work to be done in the future. To better utilize the syntactic information, we will put more effort on the study of the dependency LM with deeper syntactic knowledge. Moreover, we believe that modeling the syntax of both sides is a promising method to further improve BTG-based translation and this will become a study emphasis in our future research. Finally, inspired by (Tu et al., 2010), we will replace 1-best dependency trees with dependency forests to further increase the phrase coverage.

Acknowledgement

The authors were supported by National Natural Science Foundation of China, Contracts

60736014 and 60873167, and 863 State Key Project No.2006AA010108. We thank the anonymous reviewers for their insightful comments. We are also grateful to Zhaopeng Tu, Shu Cai and Xianhua Li for their helpful feedback.

References

Chiang, David. 2007. Hierarchical phrase-based translation. *Computational Linguistics*.

Ding, Yuan and Martha Palmer. 2005. Machine translation using probabilistic synchronous dependency insertion grammars. In *Proc. of ACL*.

Eisner, Jason. 2003. Learning non-isomorphic tree mappings for machine translation. In *Proc. of ACL*.

Fox, Heidi J. 2002. Phrasal cohesion and statistical machine translation. In *Proc. of EMNLP*.

Galley, Michel, Jonathan Graehl, Kevin Knight, Daniel Marcu, Steve DeNeefe, Wei Wang, and Ignacio Thayer. 2006. Scalable inference and training of context-rich syntactic translation models. In *Proc. of ACL*.

Hellwig, Peter. 2006. Parsing with dependency grammars, volume ii. *An International Handbook of Contemporary Research*.

Huang, Liang, Wenbin Jiang, and Qun Liu. 2009. Bilingually-constrained (monolingual) shift-reduce parsing. In *Proc. of EMNLP*.

Koehn, Philipp, Amittai Axelrod, Alexandra Birch Mayne, Chris Callison-Burch, Miles Osborne, and David Talbot. 2005. Statistical phrase-based translation. In *Proceedings International Workshop on Spoken Language Translation*.

Lin, Dekang. 2004. A path-based transfer model for machine translation. In *Proc. of Coling*.

Liu, Yang, Yajuan Lü, and Qun Liu. 2009. Improving tree-to-tree translation with packed forests. In *Proc. of ACL*.

Och, Franz Josef and Hermann Ney. 2000. Improved statistical alignment models. In *Proc. of ACL*.

Och, Franz Josef. 2003. Minimum error rate training in statistical machine translation. In *Proc. of ACL*.

Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proc. of ACL*.

Quirk, Christopher, Arul Menezes, and Colin Cherry. 2005. Dependency treelet translation: Syntactically informed phrasal smt. In *Proc. of ACL*.

Setiawan, Hendra, Min-Yen Kan, and Haizhou Li. 2007. Ordering phrases with function words. In *Proc. of ACL*.

Shen, Libin, Jinxi Xu, and Ralph Weischedel. 2008. A new string-to-dependency machine translation algorithm with a target dependency language model. In *Proc. of ACL*.

Stolcke, Andreas. 2002. Srilm - an extensible language modeling toolkit. In *Proc. of ICSLP*.

Tu, Zhaopeng, Yang Liu, Young-Sook Hwang, Qun Liu, and Shouxun Lin. 2010. Dependency forest for statistical machine translation. In *Proc. of COLING*.

Wu, Dekai. 1995. Stochastic inversion transduction grammars, with appliction to segmentation, bracketing, and alignment of parallel corpora. In *Proc. of IJCAI*.

Wu, Dekai. 1996. A polynomial-time algorithm for statistical machine translation. In *Proc. of ACL*.

Xiong, Deyi, Qun Liu, and Shouxun Lin. 2006. Maximum entropy based phrase reordering model for statistical machine translation. In *Proc. of ACL*.

Xiong, Deyi, Min Zhang, Aiti Aw, and Haizhou Li. 2008. Linguistically annotated BTG for statistical machine translation. In *Proc. of Coling*.

Zens, Richard, Hermann Ney, Taro Watanabe, and Eiichiro Sumita. 2004. A polynomial-time algorithm for statistical machine translation. In *Proc. of Coling*.

Zhang, Min and Haizhou Li. 2009. Tree kernel-based svm with structured syntactic knowledgefor btg-based phrase reordering. In *Proc. of EMNLP*.

Zhang, Ying, Stephan Vogel, and Alex Waibel. 2004. Interpreting bleu/nist scores how much improvement do we need to have a better system? In *Proc. of LREC*.

Zhang, Dongdong, Mu Li, Chi-Ho Li, and Ming Zhou. 2007. Phrase reordering model integrating syntactic knowledge for smt. In *Proc. of EMNLP*.

Zhang, Min, Hongfei Jiang, Aiti Aw, Haizhou Li, Chew Lim Tan, and Sheng Li. 2008. A tree sequence alignment-based tree-to-tree translation model. In *Proc. of ACL*.

Zhang, Le. 2004. Maximum entropy modeling tooklkit for python and c++.

Semi-supervised Semantic Pattern Discovery with Guidance from Unsupervised Pattern Clusters

Ang Sun
Computer Science Department
New York University
asun@cs.nyu.edu

Ralph Grishman
Computer Science Department
New York University
grishman@cs.nyu.edu

Abstract

We present a simple algorithm for clustering semantic patterns based on distributional similarity and use cluster memberships to guide semi-supervised pattern discovery. We apply this approach to the task of relation extraction. The evaluation results demonstrate that our novel bootstrapping procedure significantly outperforms a standard bootstrapping. Most importantly, our algorithm can effectively prevent semantic drift and provide semi-supervised learning with a natural stopping criterion.

1 Introduction

The Natural Language Processing (NLP) community faces new tasks and new domains all the time. Without enough labeled data of a new task or a new domain to conduct supervised learning, semi-supervised learning (SSL) is particularly attractive to NLP researchers since it only requires a handful of labeled examples, known as seeds. SSL starts with these seeds to train an initial model; it then applies this model to a large volume of unlabeled data to get more labeled examples and adds the most confident ones as new seeds to re-train the model. This iterative procedure has been successfully applied to a variety of NLP tasks, such as hypernym/hyponym extraction (Hearst, 1992), word sense disambiguation (Yarowsky, 1995), question answering (Ravichandran and Hovy, 2002), and information extraction (Brin, 1998; Collins and Singer, 1999; Riloff and Jones, 1999; Agichtein and Gravano, 2000; Yangarber et al., 2000; Chen and Ji, 2009).

While SSL can give good performance for many tasks, it is a procedure born with two defects. One is semantic drift. When SSL is under-constrained, the semantics of newly promoted examples might stray away from the original meaning of seed examples as discussed in (Brin, 1998; Curran et al., 2007; Carlson et al., 2010). For example, a SSL procedure to learn semantic patterns for the *LocatedIn* relation (PERSON in LOCATION/GPE[1]) might accept patterns for the *Employment* relation (employee of GPE / ORGANIZATION) because many unlabeled pairs of names are connected by patterns belonging to multiple relations. Patterns connecting <*Bill Clinton, Arkansas*> include *LocatedIn* patterns such as "*visit*", "*arrive in*" and "*fly to*", but also patterns indicating other relations such as "*governor of*", "*born in*", and "*campaign in*". Similar analyses can be applied to many other examples such as <*Bush, Texas*> and <*Schwarzenegger, California*>. Without careful design, SSL procedures usually accept bogus examples during certain iterations and hence the learning quality degrades.

The other shortcoming of SSL is its lack of natural stopping criteria. Most SSL algorithms either run a fixed number of iterations (Agichtein and Gravano, 2000) or run against a separate labeled test set to find the best stopping criterion (Abney, 2008). The former solution needs a human to keep eyeballing the learning quality of different iterations and set ad-hoc thresholds accordingly. The latter requires a

[1] These are the types of relations and names used in the NIST-sponsored ACE evaluation. http://www.itl.nist.gov/iad/mig//tests/ace/. GPE represents a Geo-Political Entity — an entity with land and a government.

Coling 2008: Poster Volume, pages 1194–1202,
Beijing, August 2010

separate labeled test set for each new task or domain. They make SSL less appealing than it could be since the intention of using SSL is to minimize supervision.

In this paper, we propose a novel learning framework which can automatically monitor the semantic drift and find a natural stopping criterion for SSL. Central to our idea is that instead of using unlabeled data directly in SSL, we first cluster the seeds and unlabeled data in an unsupervised way before conducting SSL. The semantics of unsupervised clusters are usually unknown. However, the cluster to which the seeds belong can serve as the target cluster. Then we guide the SSL procedure using the target cluster. Under such learning settings, semantic drift can be automatically detected and a stopping criterion can be found: stopping the SSL procedure when it tends to accept examples belonging to clusters other than the target cluster.

We demonstrate in this paper the above general idea by considering a bootstrapping procedure to discover semantic patterns for extracting relations between named entities (NE). Standard bootstrapping usually starts with some high-precision and high frequency seed patterns for a specific relation to match named entities, then it uses newly promoted entities to search for additional confident patterns connecting them. It is a procedure driven by the duality between patterns and entities: a good pattern can connect more than one pair of named entities and a pair of named entities is usually connected by more than one good pattern.

We present a new bootstrapping procedure in which we first cluster the seed and other patterns in a large corpus based on distributional similarity. We then guide the bootstrapping using the target cluster.

The next section describes our unsupervised pattern clusters. Section 3 presents the details of our novel bootstrapping procedure with guidance from pattern clusters. We evaluate our algorithms in Section 4 and present related work in Section 5. We draw conclusions and point to future work in Section 6.

2 Pattern Clusters

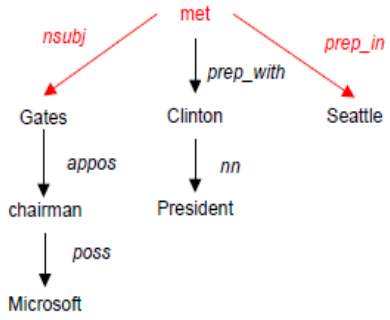
2.1 Distributional Hypothesis

The Distributional Hypothesis (Harris, 1954) states that words that tend to occur in similar contexts tend to have similar meanings. Lin and Pantel (2001) extended this hypothesis to cover patterns (dependency paths in their case). The idea of the extension is that if two patterns tend to occur in similar contexts then the meanings of the patterns tend to be similar. For example, in "*X solves Y*" and "*X finds a solution to Y*", "*solves*" and "*finds a solution to*" share many common *Xs* and *Ys* and hence are similar to each other. This extended distributional hypothesis serves as the basis on which we compute similarities for each pair of patterns.

2.2 Pattern Representation — Shortest Dependency Path

We adopt a shortest dependency path (SDP) representation of relation patterns. SDP has demonstrated its power in kernel methods for relation extraction (Bunescu and Mooney, 2005). Its capability in capturing most of the information of interest is also evidenced by a systematic comparison of effectiveness of different information extraction (IE) patterns in (Stevenson and Greenwood, 2006) [2] . For example, "*nsubj* ← met → *prep_in*" is able to represent *LocatedIn* between "*Gates*" and "*Seattle*" while a token-based pattern would be much less general because it would have to specify all the intervening tokens.

Figure 1. Stanford dependency tree for sentence *"Gates, Microsoft's chairman, met with President Clinton in Seattle".*



[2] SDP is equivalent to the linked chains described in Stevenson and Greenwood (2006) when the dependency of a sentence is represented as a tree not a graph.

2.3 Pre-processing

We tag and parse each sentence in our corpus with the NYU named entity tagger[3] and the Stanford dependency parser. Then for each pair of names in the dependency tree, we extract the SDP connecting them. Names in the path are replaced by their types. We require SDP to contain at least one verb or noun. We use the base form of words in SDP. We also require the length of the path (defined as the number of dependency relations and words in it) to be between 3 and 7. Short paths are more likely to be generic patterns such as "*of*" and can be handled separately as in (Pantel and Pennacchiotti, 2006). Very long paths are more likely to be non-relation patterns and too sparse to be useful even if they are relation patterns.

2.4 Clustering Algorithm

The basic idea of our clustering algorithm is to group all the paths (including the seed paths used later for SSL) in our corpus into different clusters based on distributional similarities. We first extract a variety of features from the named entities *X* and *Y* connected by a path *P* as shown in Table 1. We then compute an analogue of tf-idf for each feature *f* of *P* as follows: tf as the number of corpus instances of *P* having feature *f* divided by the number of instances of *P*; idf as the total number of paths in the corpus divided by the number of paths with at least one instance with feature *f*. Then we adopt a vector space model, i.e., we construct a tf-idf feature vector for each *P*. Now we compute the similarity between two vectors/paths using Cosine similarity and cluster all the paths using Complete Linkage.

Some technical details deserve more attention here.

Feature extraction: We extract more types of features than the DIRT paraphrase discovery procedure used in (Lin and Pantel, 2001). Lin and Pantel (2001) considered *X* and *Y* separately while we also use the conjunction of *X* and *Y*. We also extract named entity types as features since we are interested in discovering relations among different types of names. Some names are ambiguous such as *Jordan*. We hope coupling the type with the string of the name may alleviate the ambiguity.

Table 1. Sample features for "*X visited Y*" as in "*Jordan visited China*"

Feature Type	Example
Name Type of *X*	*LEFT_PERSON*
Name Type of *Y*	*RIGHT_GPE*
Combination of Types of X and *Y*	*PERSON_GPE*
Conjunction of String and Type of *X*	*LEFT_Jordan_PERSON*
Conjunction of String and Type of *Y*	*RIGHT_China_GPE*
Conjunction of Strings and Types of *X* and *Y*	*Jordan_PERSON_China_GPE*

Similarity measure and clustering method: There are many ways to compute the similarity/distance between two feature vectors, such as *Cosine*, *Euclidean*, *Hamming*, and *Jaccard coefficient*. There are also many standard clustering algorithms. A systematic comparison of the performance of different distance measures and clustering algorithms is beyond the scope of this paper.

3 Semi-supervised Relation Pattern Discovery

We first present a standard bootstrapping algorithm coupled with analyses of some of its shortcomings. Then we describe our new bootstrapping procedure which is guided by pattern clusters.

3.1 Bootstrapping without Guidance

The procedure associates a precision between 0 and 1 with each pattern, and a confidence between 0 and 1 with each name pair. Initially the seed patterns for a specific relation *R* have precision 1 and all other patterns 0. It consists of the following steps:

Step1: Use seed patterns to match new NE pairs and evaluate NE pairs.

Intuitively, for a newly matched NE pair N_i, if many of the *k* patterns connecting the two names are high-precision patterns then the name pair has a high confidence. The confidence is computed by the following formula.

$$Conf(N_i) = 1 - \prod_{j=1}^{k} (1 - \Pr ec(p_j)) \quad (1)$$

[3] Please refer to Grishman et al. (2005) and http://cs.nyu.edu/grishman/jet/license.html

Problem: While the intuition is correct, in practice this will over-rank NE pairs which are not only matched by patterns belonging to the target relation *R* but are also connected by patterns of many other relations. This is because of the initial settings used in many SSL systems: seeds are assigned high confidence. Thus all NE pairs matched by initial seed patterns will have very high confidence.

Suppose the target relation is *LocatedIn*, and "*visited*" is a seed pattern; then the <*Clinton*, *Arkansas*> example will be over-rated because we cannot take into account that it would also match patterns of other relations such as *PersonGovernorOfLocation* and *PersonBornInLocation* in a real corpus. This will cause a vicious circle, i.e., bogus NE pairs extract more bogus patterns which further extract more bogus NE pairs. We believe this flaw of the initial settings partially results in the semantic drift problem.

One can imagine that this is not a problem that can be solved by using a different formula to replace the one presented here. A possible solution is to study the structure of unlabeled data (NE pairs in our case) and integrate this structure information into the initial settings. Indeed, this is where pattern clusters come into play. We will demonstrate this in Section 3.2.

Step 2: Use NE pairs to search for new patterns and rank patterns.

Similar to the intuition in Step 1, for a pattern *p*, if many of the NE pairs it matches are very confident then *p* has many supporters and should have a high ranking. We can use formula (2) to estimate the confidence of patterns and rank them.

$$Conf(p) = \frac{Sup(p)}{|H|} \bullet \log Sup(p) \qquad (2)$$

Here $|H|$ is the number of unique NE pairs matched by *p* and *Sup(p)* is the sum of the support it can get from the $|H|$ pairs:

$$Sup(p) = \sum_{j=1}^{|H|} Conf(N_j) \qquad (3)$$

The precision of *p* is given by the average confidence of the NE pairs matched by *p*.

$$\Pr ec(p) = \frac{Sup(p)}{|H|} \qquad (4)$$

Formula (4) normalizes the precision to range from 0 to 1. As a result the confidence of each NE pair is also normalized to between 0 and 1.

Step 3: Accept patterns

Most systems accept the *K* top ranked patterns in Step 2 as new seeds, subject to some restrictions such as requiring the differences of confidence of the *K* patterns to be within a small range.

Step 4: Loop or stop

The procedure now decides whether to repeat from Step 1 or to terminate.

Most systems simply do not know when to stop. They either run a fixed number of iterations or use some held-out data to find one criterion that works the best for the held-out data.

3.2 Bootstrapping Guided by Clusters

Recall that our clustering algorithm in Section 2 provides us with *K* clusters, each of which contains *n* (*n* differs in different clusters) patterns. Every pattern in our corpus now has a cluster membership (the seed patterns have the same membership).

The most important benefit from our pattern clusters is that now we can measure how strongly a NE pair N_i is associated with our target cluster C_t (the one to which the seed patterns belong).

$$\Pr ob(N_i \in C_t) = \frac{\sum_{p \in C_t} freq(N_i, p)}{m} \qquad (5)$$

Here $freq(N_i, p)$ is the number of times *p* matches N_i and *m* is the total number of pattern instances matching N_i.

We integrate this prior cluster distribution of each NE pair into the initial settings of our new bootstrapping procedure.

Step1: Use seed patterns to match new NE pairs and evaluate NE pairs.

Assumption: A good NE pair must be strongly associated with the target cluster and can be matched by multiple high-precision patterns.

So we evaluate a NE pair by the harmonic mean of two confidence scores, namely the confidence as its association with the target cluster and the confidence given by the patterns matching it.

$$Conf(N_i) = 2 \bullet \frac{Semi\_Conf(N_i) \bullet Cluster\_Conf(N_i)}{Semi\_Conf(N_i) + Cluster\_Conf(N_i)} \quad (6)$$

$$Semi\_Conf(N_i) = 1 - \prod_{j=1}^{k} (1 - Prec(p_j)) \quad (7)$$

$$Cluster\_Conf(N_i) = \Pr ob(N_i \in C_t) \quad (8)$$

Under such settings, <*Clinton, Arkansas*> will be assigned a lower confidence score for the *LocatedIn* relation than it is in the standard bootstrapping. Even if we assign high precision to our seed patterns such as "*visited*" and consequently the *Semi_Conf* is very high, it can still be discounted by the *Cluster_Conf*[4].

Step 2: Use NE pairs to search for new patterns and rank patterns.

All the measurement functions are the same as those used in the standard bootstrapping. However, with better ranking of NE pairs in Step 1, the patterns are also ranked better than they are in the standard bootstrapping.

Step 3: Accept patterns

We also accept the *K* top ranked patterns.

Step 4: Loop or stop

Since each pattern in our corpus has a cluster membership, we can monitor the semantic drift easily and naturally stop: it drifts when the procedure tries to accept patterns which do not belong to the target cluster; we can stop when the procedure tends to accept more patterns outside of the target cluster.

If our clustering algorithm can give us perfect pattern clusters, we can stop bootstrapping immediately after it accepts the first pattern not belonging to the target cluster. Then the bootstrapping becomes redundant since all it does is to consume the patterns of the target cluster.

Facing the reality of the behavior of many clustering algorithms, we allow the procedure to occasionally accept patterns outside of the target cluster but we are not tolerant when it tries to accept more patterns outside of the target cluster than patterns in it. Note that when such patterns are accepted they will be moved to the target cluster and invoke the recomputation of *Cluster_Conf* of NE pairs connected by these patterns. The ranking functions in step 1 and 2 insure that the procedure will only accept patterns which can gain strong support from NE pairs that are strongly associated with the target cluster and are connected by many confident patterns.

4 Experiments

4.1 Corpus

Our corpora contain 37 years of news articles: TDT5, NYT(94-00), APW(98-00), XINHUA(96-00), WSJ(94-96), LATWP(94-97), REUFF(94-96), REUTE(94-96), and WSJSF(87-94). It contains roughly 65 million sentences and 1.3 billion tokens.

4.2 Seeds

Seeds of the 3 relations we are going to test are given in table 2. *LocatedIn* detects relation between PERSON and LOCATION/GPE; Social (*SOC*) detects social relations (either business or family) between PERSON and PERSON; Employment (*EMP*) detects employment relations between PERSON and ORGANIZATION.

Table 2. Seed Patterns

Relation	Seeds
Located-in	*nsubj'* visit *dobj* *nsubj'* travel *prep_to* *poss'* trip *prep_to*
SOC	*appos* friend/lawyer *poss* *appos* son/spokesman *prep_of/prep_for* *nsubj'* fire *dobj* *nsubjpass'* fire *agent*
EMP[5]	*appos* chairman/executive/founder *prep_of* *appos* editor *prep_of* *appos* director/head/officer/analyst *prep_at* *appos* manager *prep_with*

(*nsubj*, *dobj*, *prep*, *appos*, *poss*, *nsubjpass*, *agent* stand for subject, direct object, preposition, apposition, possessive, passive nominal subject and complement of passive verb. The quote marks in Table 2 and Table 3 denote inverse dependencies in the dependency path.)

We work on these three relations mainly because of the availability of benchmark evaluation data. These are the most frequent relations in our evaluation data.

[4] The *Cluster_Conf* of <*Clinton, Arkansas*> related to the *LocatedIn* relation is indeed very low (less than 0.1) in our experiments.

[5] We provide more seeds (executives and staff) for *EMP* because it has been pointed out in (Sun, 2009) that *EMP* contains a lot of job titles.

4.3 Unsupervised Experiments

We run the clustering algorithm described in Section 2 using all the 37 years' data. We require that a pattern match at least 7 distinct NE pairs and that an NE pair must be connected by at least 7 unique patterns. As a result, there are 635,128 patterns (22,225 unique ones) used in experiments. We use 0.005 as the cutoff threshold of complete linkage. The threshold is decided by trying a series of thresholds and searching for the maximal[6] one that is capable of placing the seed patterns for each relation into a single cluster. Table 3 shows the top 15 patterns (ranked by their corpus frequency) of the cluster into which our *LocatedIn* seeds fall.

Table 3. Top 15 patterns in the *LocatedIn* Cluster

Index	Pattern	Frequency
1	*nsubj'* said *prep_in*	2203
2	*nsubj'* visit *dobj*	1831
3	*poss'* visit *prep_to*	1522
4	*nsubj'* return *prep_to*	1394
5	*nsubj'* tell *prep_in*	1363
6	*nsubj'* be *prep_in*	1283
7	*nsubj'* arrive *prep_in*	1113
8	*nsubj'* leave *dobj*	1106
9	*nsubj'* go *prep_to*	926
10	*nsubj'* fly *prep_to*	700
11	*nsubj'* come *prep_to*	658
12	*appos* leader *poss*	454
13	*poss'* trip *prep_to*	442
14	*rcmod* be *prep_in*	419
15	*nsubj'* make *prep_in*	418

4.4 Semi-supervised Experiments

To provide strong statistical evidence, we divide our data into 10 folds (combinations of news articles from different years and different news resources). We then run both the standard and our new bootstrapping on the 10 folds. For both procedures, we accept n patterns in a single iteration (n is initialized to 2 and set to $n + 1$ after each iteration). We run 50 iterations in the standard bootstrapping and 1,325 patterns are accepted for each fold and each relation. Our new bootstrapping procedure stops when there are two consecutive iterations in which more than half of the newly accepted patterns do not belong to the target cluster. Thus the number of patterns accepted for each fold and each relation differs as the last iteration differs.

[6] We choose the maximal value because many clusters will be merged to a single one when the threshold is close to 0, making the clusters too general to be useful.

4.5 Evaluation

The output of our bootstrapping procedures is 60 sets of patterns (3 relations × 2 methods × 10 folds). We need a data set and evaluation method which can compare their effectiveness equally and consistently.

Evaluation data: ACE 2004 training data. ACE does not provide relation annotation between each pair of names. For example, in "*US President Clinton said that the United States …*" ACE annotates an *EMP* relation between the name "*US*" and nominal "*President*". There is no annotation between "*US*" and "*Clinton*". However, it provides entity co-reference information which connects "*President*" to "*Clinton*". So we take advantage of this entity co-reference information to automatically re-annotate the relations where possible to link a pair of names within a single sentence. The re-annotation yields an *EMP* relation between "*US*" and "*Clinton*". The re-annotation is reviewed by hand to avoid adding a relation linking "*Clinton*" and the more distant co-referent "*United States*", even though "*US*" and "*the United States*" refer to the same entity. This data set provides us with 412/3492 positive/negative relation instances between names. Among the 412 positive instances, there are 188/117/35 instances for *EMP/LocatedIn/SOC* relations.

Evaluation method: We adopt a direct evaluation method, i.e., use our sets of patterns to extract relations between names on ACE data. Applying patterns to a benchmark data set can provide us with better precision/recall analyses. We use a strict pattern match strategy. We can certainly take advantage of loose match or add patterns as additional features to feature-based relation extraction systems to boost our performance but we do not want these to complicate the comparison of the standard and our new bootstrapping procedures.

4.6 Results and Analyses

We average our results on the 10 folds. We plot precision against recall and semantic drift rate against iterations (Drift). We compute the semantic drift rate as the percentage of false

Figure 2. Performance for *EMP/LocatedIn/SOC*

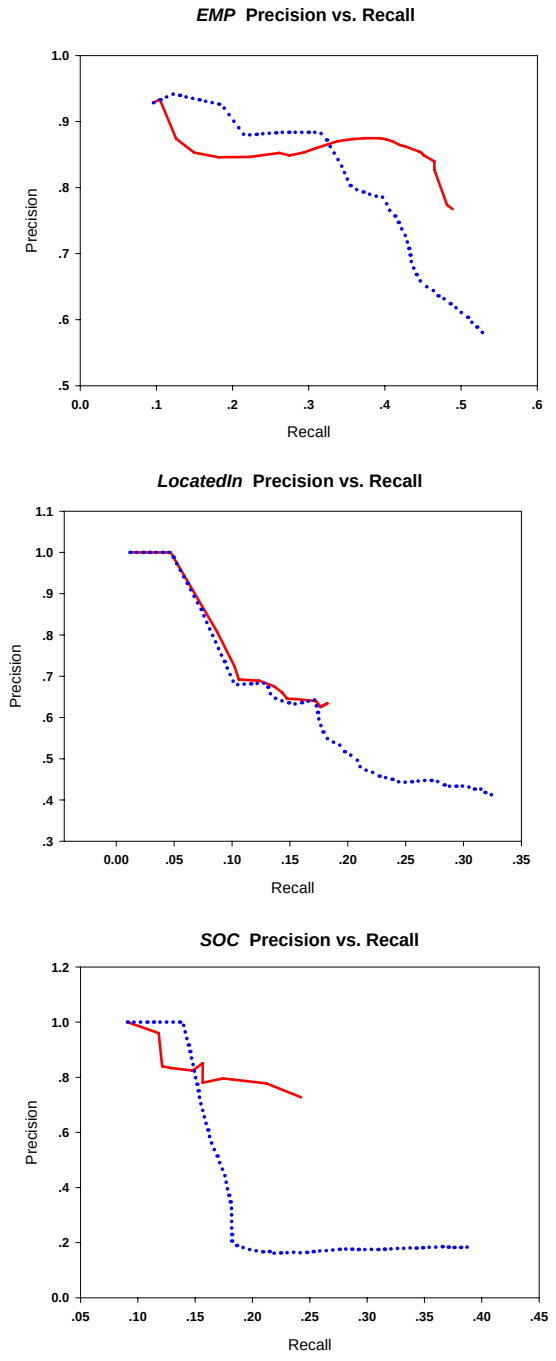
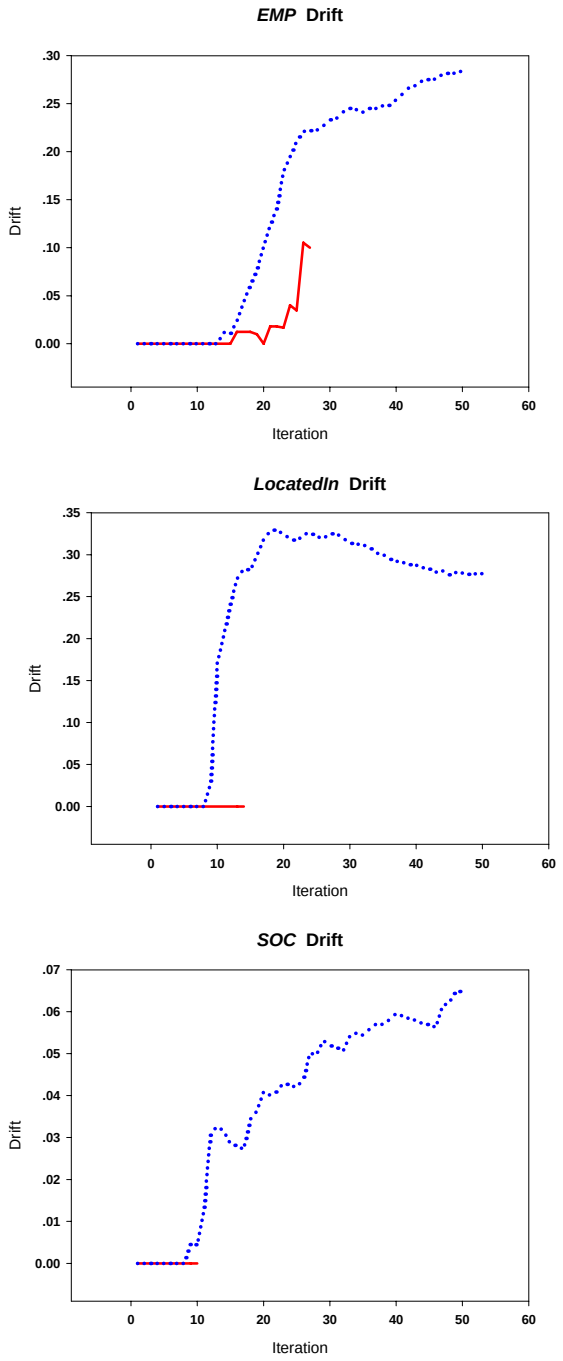


Figure 3. Drift for *EMP/LocatedIn/SOC*



positive instances belonging to ACE relations other than the target relation. Take *EMP* for example, we compute how many of the false positive instances belonging to other relations such as *LocatedIn*, *SOC* and other ACE relations. In all plots, red solid lines represent bootstrapping with guidance from clusters and blue dotted lines standard bootstrapping.

There are a number of conclusions that can be drawn from these results. We are particularly interested in the following two questions: To what extent did we prevent semantic drift by the guidance of pattern clusters? Did we stop at the right point, i.e., can we keep high precision while maintaining near maximal recall?

1) It is obvious from the drift curves that our bootstrapping effectively prevents semantic drift. Indeed, there is no drift at all when *LocatedIn*

and *SOC* learners terminate. Although drift indeed occurs in the *EMP* relation, its curve is much lower than that of the standard bootstrapping.

2) Our new procedure terminates when the precision is still high while maintaining a reasonable recall. Our bootstrapping for *EMP/SOC/LocatedIn* terminates at F-measures of 60/37/28 (in percentage). We conducted the Wilcoxon Matched-Pairs Signed-Ranks Test on the 10 folds, comparing the F-measures of the last iteration of our bootstrapping guided by clusters and the iteration which provides the best average F-measure over the 3 relations of the standard bootstrapping. The results show that the improvement of using clusters to guide bootstrapping is significant at a 97% confidence level.

We hypothesize that when working on dozens or hundreds of relations the gain of our procedure will be even bigger since we can effectively prevent inter-class errors.

5 Related Work

Recent research starts exploring unlabeled data for discriminative learning. Miller et al., (2004) augmented name tagging training data with hierarchical word clusters and encoded cluster membership in features for improving name tagging. Lin and Wu (2009) further explored a two-stage cluster-based approach: first clustering phrases and then relying on a supervised learner to identify useful clusters and assign proper weights to cluster features. Other similar work includes (Wong and Ng, 2007) for name tagging, and (Koo et. al., 2008) for dependency parsing.

While similar in spirit, our supervision is minimal, i.e., we only use a few seeds while the above approaches rely on a large amount of labeled data. To the best of our knowledge, the theme explored in this paper is the first study of using pattern clusters for preventing semantic drift in semi-supervised pattern discovery.

Recent research also explored the idea of driving SSL with explicit constraints constructed by hand such as identifying mutual exclusion of different categories (i.e., people and sport are mutually exclusive). This is termed constraint-driven learning in (Chang et al., 2007), coupled learning in (Carlson et al., 2010) and counter-training in (Yangarber, 2003). The learning quality largely depends on the completeness of explicit constraints. While we share the same goal, i.e., to prevent semantic drift, we rely on unsupervised clusters to discover implicit constraints for us instead of generating constraints by hand.

Our research is also close to semi-supervised IE pattern learners including (Riloff and Jones, 1999), (Agichtein and Gravano, 2000), (Yangarber et al., 2000), and many others. While they conduct bootstrapping on unlabeled data directly, we first cluster unlabeled data and then bootstrap with help from clusters.

There are also clear connections to work on unsupervised relation discovery (Hasegawa et al., 2004; Zhang et al., 2005; Rosenfeld and Feldman, 2007). They group pairs of names into relation clusters based on the contexts between names while we group the contexts/patterns into clusters based on features extracted from names.

6 Conclusions and Future Work

We presented a simple algorithm for clustering patterns and used pattern clusters to guide semi-supervised semantic pattern discovery. The novel bootstrapping procedure can achieve the best F-1 score while maintaining a good trade-off between precision and recall. We also demonstrated that it can effectively prevent semantic drift and naturally terminate.

We plan to extend this idea to improve relation extraction performance with a richer model as used in (Zhang et al., 2004; Zhou et al., 2008) than a simple pattern learner. The feature space will be much larger than the one adopted in this paper. We will investigate how to overcome the memory bottleneck when we apply rich models to millions of instances.

7 Acknowledgements

We would like to thank Prof. Satoshi Sekine for his useful suggestions.

References

Steven Abney. 2008. *Semisupervised Learning for Computational Linguistics*, Chapman and Hall.

Eugene Agichtein and Luis Gravano. 2000. Snowball: Extracting relations from large plain-text

collections. In *Proc. of the Fifth ACM International Conference on Digital Libraries*.

Sergey Brin. Extracting patterns and relations from the World-Wide Web. 1998. In *Proc. of the 1998 Intl. Workshop on the Web and Databases*.

Razvan C. Bunescu and Raymond J. Mooney. 2005. A Shortest Path Dependency Kernel for Relation Extraction. In *Proc. of HLT/EMNLP*.

Andrew Carlson, Justin Betteridge, Richard C. Wang, Estevam Rafael Hruschka Junior and Tom M. Mitchell. 2010. Coupled Semi-Supervised Learning for Information Extraction. In *WSDM*.

Ming-Wei Chang, Lev Ratinov, and Dan Roth. 2007. Guiding semisupervision with constraint-driven learning. In *Proc. of ACL-2007*, Prague.

Zheng Chen and Heng Ji. 2009. Can One Language Bootstrap the Other: A Case Study on Event Extraction. In *NAACL HLT Workshop on Semi-supervised Learning for NLP*.

Michael Collins and Yoram Singer. 1999. Unsupervised models for named entity classication. In *Proc. of EMNLP-99*.

James R. Curran, Tara Murphy, and Bernhard Scholz. 2007. Minimising semantic drift with Mutual Exclusion Bootstrapping. In *Proc. of PACLING*.

Ralph Grishman, David Westbrook and Adam Meyers. 2005. NYU's English ACE 2005 System Description. ACE 2005 Evaluation Workshop.

Zellig S. Harris. 1954. Distributional Structure. *Word. Vol 10,1954, 146-162*.

Takaaki Hasegawa, Satoshi Sekine, Ralph Grishman. 2004. Discovering Relations among Named Entities from Large Corpora. In *Proc. of ACL-04*.

Marti Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proc. of the 14th Intl. Conf. on Computational Linguistics*.

Terry Koo, Xavier Carreras, and Michael Collins. 2008. Simple Semi-supervised Dependency Parsing. In *Proceedings of ACL-08: HLT*.

Dekang Lin and Patrick Pantel. 2001. Discovery of inference rules for question-answering. *Natural Language Engineering, 7(4):343–360*.

Dekang Lin and Xiaoyun Wu. 2009. Phrase Clustering for Discriminative Learning. In *Proceedings of the ACL and IJCNLP 2009*.

Marie-Catherine de Marneffe and Christopher D. Manning. 2008. The Stanford typed dependencies representation. In *COLING Workshop on Cross-framework and Cross-domain Parser Evaluation*.

Scott Miller, Jethran Guinness and Alex Zamanian. 2004. Name Tagging with Word Clusters and Discriminative Training. In *Proc. of HLT-NAACL*.

Patrick Pantel and Marco Pennacchiotti. 2006. Espresso: Leveraging Generic Patterns for Automatically Harvesting Semantic Relations. In *Proc. of COLING-06 and ACL-06*.

Deepak Ravichandran and Eduard Hovy. 2002. Learning Surface Text Patterns for a Question Answering System. In *Proc. of ACL-2002.*

Ellen Riloff and Rosie Jones. 1999. Learning dictionaries for information extraction by multi-level bootstrapping. In *Proc. of AAAI-99*.

Benjamin Rosenfeld, Ronen Feldman. 2007. Clustering for Unsupervised Relation Identification. In *Proc. of CIKM '07*.

Mark Stevenson and Mark A. Greenwood. 2006. Comparing Information Extraction Pattern Models. In *Proceedings of the Workshop on Information Extraction Beyond The Document*.

Mark Stevenson and Mark A. Greenwood. 2005. A Semantic Approach to IE Pattern Induction. In *Proc. of the 43rd Annual Meeting of the ACL*.

Ang Sun. 2009. A Two-stage Bootstrapping Algorithm for Relation Extraction. In *RANLP-09*.

Yingchuan Wong and Hwee Tou Ng. 2007. One Class per Named Entity: Exploiting Unlabeled Text for Named Entity Recognition. In *Proc. of IJCAI-07*.

Roman Yangarber. 2003. Counter-training in the discovery of semantic patterns. In *Proc. of ACL*.

Roman Yangarber, Ralph Grishman, Pasi Tapanainen and Silja Huttunen. 2000. Automatic acquisition of domain knowledge for information extraction. In *Proc. of COLING-2000*.

David Yarowsky. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *Proc. of ACL-95*.

Min Zhang, Jian Su, Danmei Wang, Guodong Zhou, and Chew Lim Tan. 2005. Discovering Relations Between Named Entities from a Large Raw Corpus Using Tree Similarity-Based Clustering. In *IJCNLP 2005, LNAI 3651, pp. 378 – 389*.

Zhu Zhang. (2004). Weakly supervised relation classification for information extraction. In *Proc. of CIKM'2004*.

GuoDong Zhou, JunHui Li, LongHua Qian and QiaoMing Zhu. 2008. Semi-supervised learning for relation extraction. *IJCNLP'2008:32-39*.

Utilizing Variability of Time and Term Content, within and across Users in Session Detection

Shuqi Sun[1], **Sheng Li**[1], **Muyun Yang**[1], **Haoliang Qi**[2], **Tiejun Zhao**[1]
[1]Harbin Institute of Technology, [2]Heilongjiang Institute of Technology
{sqsun, ymy, tjzhao}@mtlab.hit.edu.cn, lisheng@hit.edu.cn
haoliang.qi@gmail.com

Abstract

In this paper, we describe a SVM classification framework of session detection task on both Chinese and English query logs. With eight features on the aspects of temporal and content information extracted from pairs of successive queries, the classification models achieve significantly superior performance than the stat-of-the-art method. Additionally, we find through ROC analysis that there exists great discrimination power variability among different features and within the same feature across different users. To fully utilize this variability, we build local models for individual users and combine their predictions with those from the global model. Experiments show that the local models do make significant improvements to the global model, although the amount is small.

1 Introduction

To provide users better experiences of search engines, inspecting users' activities and inferring users' interests are indispensible. Query logs recorded by search engines serves well for these purposes. Query log conveys the user interest information in the form of slices of the query stream. Thus the task of session detection consists in distinguishing slice that corresponds to a user interest from other ones, and thus this paper, we adopt the definition of a session following (Jansen et al., 2007):

(A session is) a series of interactions by the user toward addressing a single information need.

This definition is equivalent to that of the "search goal" proposed by Jones and Klinkner (2008), which corresponds to an *atomic* information need, resulting in one or more queries.

This paper adopts a classification point of view to the task of session detection (Jones and Klinkner, 2008). Given a pair of successive queries in a query log, we examine it in various viewpoints (i.e. features) such as time proximity and similarity of the content of the two queries to determine whether these two queries cross a border of a search session. In other words, we classify the gap between the two queries into two classes: session shift and session continuation. In practice, search goals in a search mission and different search missions could be intermingled, and increase the difficulty of correctly identifying them. In this paper, we do not take this issue into account and simply treat all boundaries between intermingled search goals as session shifts. The chief advantage in this choice is that we will have the opportunity to make classification model working online without caching user's queries that are pending to be assigned to a session.

Various studies built accurate models in predicting session boundaries and in distinguishing intermingled sessions, and they are summarized in Section 2. However, none of these works analyzed the contribution of individual features from a user-oriented viewpoint, or evaluated a feature's discrimination power in a general scenario independent of its usage, as this paper does by conducting ROC analyses. During these analyses, we found that the discrimination power of features varies dramatically, and for different users, the discrimination power of a particular feature also does not remain constant.

Thus, it is appealing to build local models for users with have sufficient size of training examples, and combine the local models' predictions with those made by the global model trained by the whole training data. However, few of previ-

Coling 2008: Poster Volume, pages 1203–1210,
Beijing, August 2010

ous works build user-specific models for the sake of characterizing the variability in user's search activities, except that of Murray et al. (2006). To fully make use of these two aspects of variability, inspired by Murray et al., we build users' local models based on a much broader range of evidences, and show that different local models vary to a great extent, and experiments show that the local models do make significant improvements to the global model, although the amount is small.

The remainder of this paper is organized as follows: Section 2 summarizes the related work of the session detection task. In Section 3, we first describe our classification framework as well as the features utilized. Then we conduct various evaluations on both English and Chinese query logs. Section 4 introduces the approaches to building local models based on an analysis of the variability of the discrimination power of features, and combine predictions of local models with those of the global model. Section 5 discusses the experimental results and concludes this paper.

2 Related Work

The simplest method in session detection is defining a timeout threshold and marking any time gaps of successive queries that exceed the threshold as session shifts. The thresholds adopted in different studies were significantly different, ranging from 5 minutes to 30 minutes (Silverstein et al., 1999; He and Göker, 2000; Radlinski and Joachims, 2005; Downey et al., 2007). Other study suggested adopting a dynamic timeout threshold. Murray et al. (2006) proposed a user-centered hierarchical agglomerative clustering algorithm to determine timeout threshold for each user dynamically, other than setting a fixed threshold. However, Jones and Klinkner (2008) pointed out that single timeout criterion is always of limited utility, whatever its length is, and incorporating timeout features with other various features achieved satisfactory classification accuracy.

An effective approach to combining the time out features with various evidences for session detection is machine learning. He et al. (2002) collected statistical information from human annotated query logs to predict the probability a "New" pattern indicates a session shift according to the time gap between successive queries. Özmutlu and colleagues re-examined He et al.'s work, and explored other machine learning techniques such as neural networks, multiple linear regression, Monte Carlo simulation, conditional probabilities (Gayo-Avello, 2009), and HMMs (Özmutlu, 2009).

In recent studies, Jones and Klinkner (2008) built logistic regression models to identify search goals and missions, and tackled the intermingled search goal/mission issue by examining arbitrary pairs of queries in the query log. Another contribution of Jones and Klinkner is that they made a thorough analysis of contributions of individual features. However, they explored the features' contributions from a feature selection point of view rather than from a user-oriented one, and thus failed to characterize the variability of the discrimination power of the features when applied to different users.

3 Learning to Detect Session Shifts

3.1 Feature Extraction

We adopt eight features covering both the temporal and the content aspect of pairs of successive queries. Most these features are commonly used by previous studies (He and Göker, 2000; Özmutlu, 2006; Jones and Klinkner, 2008). However, in this paper, we will analyze their contributions to the resulted model in a quite different way from that in previous works.

Let $Q = (q_1, q_2, \ldots, q_n)$ denote a query log. The features are extracted from every successive pair of queries (q_i, q_{i+1}). Table 1 summarizes the features we adopt. The normalization described in Table1 is done according to the type of the feature. Features describing characters are normalized by the average length of the two queries, while those describing character-n-grams are normalized by the average size of the n-gram sets of the two queries. Character-n-grams (e.g. bigrams "*ca*" and "*at*" in "*cat*") are robust to different representations of the same topic (e.g. "*IR*" as *Information Retrieval*) and typos (e.g. "*speling" as "spelling"*), and serve as a simple stemming method. In practice, character-*n*-grams are accumulative, which means they consist of all *m*-grams with $m \leq n$.

The feature "avg_ngram_distance", a variant of the "lexical distance" in (Gayo-Avello, 2009), is more complicated than to be described briefly.

Here we first define n-gram distance (*ND*) from q_i to q_j, which is formalized as follows:

$$ND(q_i \rightarrow q_j) = 1 - \frac{\#\text{ of char.-}n\text{-gram in } q_i \text{ occur in } q_j}{\#\text{ of char.-}n\text{-gram in } q_j}$$

Note that character-n-grams are accumulative and there could be multiple occurrences of a character-n-gram in a query, so the number of a character-*n*-gram is the sum of that of all *m*-grams with $m \leq n$, and multiple occurrences are all considered. At last, the average of character-n-gram distance (*ACD*) of the pair (q_i, q_{i+1}) is:

$$ACD(q_i, q_{i+1}) = \frac{ND(q_i \rightarrow q_{i+1}) + ND(q_{i+1} \rightarrow q_i)}{2}$$

There are seven features describing the content aspect of a query pair, and they are more or less overlapped (e.g. edit_distance vs. common_char). However, we show in the next subsection that all these features are beneficial to the final performance.

Feature	Description
time_interval	time interval between successive queries
avg_ngram_ distance	avg. of character-n-gram distances
edit_disance	normalized Levenshtein edit distance
common_prefix	normalized length of prefix shared
common_suffix	normalized length of suffix shared
common_char	normalized number of characters shared
common_ngram	normalized number of character-n-grams shared
Jaccard_ngram	Jaccard distance between character-n-gram sets

Table 1. Features used in classification models

3.2 Data Preparation

The query logs we explored include an English search log tracked by AOL from Mar 1, 2006 to May, 31 2006 (Pass et al., 2006), and a Chinese search log tracked by Sogou.com, which is one of the major Chinese Search Engines, from Mar 1, 2007 to Mar 31, 2007[1]. We applied systematic sampling over the user space on the two logs, which yielded 223 users and 2809 users, corresponding to 6407 and 6917 query instances respectively[2]. Sampling over the user space instead of over the query space avoids the bias to the most active users who submit much more queries than average users.

For each sampled dataset, we invited annotators who are familiar with IR and search process to determine each pair of successive queries of interest is across the border of a session. We made trivial pre-split process under two rules:

> Queries from different users are not in the same session.
>
> Queries from different days are not in the same session.

Table 2 shows some basic statistics of the annotated data set. During the annotation process, the annotators were guided to identify the user's information need at the finest granularity ever possible, because we focus on the atomic information needs as described in Section 1. Consequently, the average numbers of queries in a session in both query logs are lower than previous studies.

	AOL log	Sogou log
Queries	6407	6917
Sessions	4571	5726
Queries per session	1.40	1.21
Longest session	21	12

Table 2. Summary of the annotation results in both query logs

3.3 Learning Framework

In this section we seek to build accurate global classification model based on the whole training data obtained in the previous sub-subsection for both the query logs. We built the models within SVM framework. The implementation of SVM we used is libSVM (Chang and Lin, 2001). For the sake of evaluations and of model integration in the next section, we set the prediction of SVM to be *probability estimation* of the test example being positive. All features were pre-scaled into [0, 1] interval. We adopted the polynomial kernel, and for both datasets, we exhaustively tried each of the subset of the eight features using 5-fold cross validation. We found that using all the eight features yielded the best classification accuracy. Thus in the experiments in rest of this

[1] http://www.sogou.com/labs/resources.html

[2] The sampling schema and sample size was determined following (Gayo-Avello, 2009).

section and the next section, we adopt the entire feature set to build global classification models.

There is one parameter to be determined for feature extraction: the length of character-n-grams. The proper lengths on AOL log and Sogou log are different. We tried the length from 1 to 9, and according to cross validation accuracy, we found the best lengths for the two logs as 6 and 3 respectively.

3.4 Experimental Results

3.4.1 Baseline Methods

We provide two base line methods for comparisons. The first method is the commonly used timeout methods. We tried different timeout thresholds from 5 minutes to 30 minutes with a step of 5 minutes, and found that for both query logs the 5 minutes' threshold yield the best overall performance.

The second method achieved the best performance on the AOL log (Gayo-Avello, 2009), which addresses the session detection problem using a geometric interpolation method, in comparison to previous studies on this query log. We re-implemented this method and evaluated it on both the datasets. Similarly, the best parameters for the two query logs are different, such as the length of a character-n-gram. We only report the performance with the best parameter settings.

3.4.2 Analyzing the Performance

We analyze the performance of the SVM models according to precision, recall, F_1-mean and $F_{1.5}$-mean of predictions on session shift and continuation against human annotation data.

The F_β-mean is defined as:

$$F_\beta\text{-mean} = \frac{(1+\beta^2)PR}{\beta^2 P + R}$$

where P denotes precision and R denotes recall. He et al. (2002) regards recall more important than precision, and set the value of β in F_β-mean to 1.5. We also report performance under this measure.

In addition to traditional precision / recall based measures, we also perform ROC (Receiver Operating Characteristic) analysis to determine the discrimination power of different methods. The best merit of ROC analysis is that given a reference set, which is usually the human annotation results, it evaluates a set of indicator's discrimination power for arbitrary binary classification problem *independent* of the critical value with which the class predictions are made.

Specifically, in the context session detection, regardless of the critical value that splits the classifier outputs into positive ones and negative ones (e.g. the 5-minutes' timeout threshold and 50% probability in SVM's output), the ROC analysis provides the overall discrimination power evaluation of the output set of a certain method (by trying to set each output value as the critical value). For the baseline method by Gayo-Avello, the core of the decision heuristics also had a critical value to be determined. For details, readers could refer to (Gayo-Avello, 2009).

3.4.3 Precision, Recall, and F-means

Before we examine the discrimination power of each session detection method's output independent of the threshold value selected. In this sub-subsection, we begin with a more traditional evaluation schema: setting a proper threshold to produce binary predictions. It is straightforward to set the threshold for SVM method to 50%, and as described in sub-subsection 3.1.1, the threshold for timeout method is 5 minutes. The threshold of Gayo-Avello's method is implied in its heuristics.

Table 3 and Table 4 show the experimental results on AOL log and Sogou log respectively. For each dataset, we performed 1000-times bootstrap resampling, generating 1000 bootstrapped datasets with the same size as the original dataset. To test the statistical significance of performance differences, we adopted Wilcoxon signed-rank test on the performance measures computed from the 1000 bootstrapped dataset, and found comparisons between each pair of methods were all significant at 95% level.

The results show that SVM method clearly outperforms the baseline methods, and timeout method performs poorly. It may be argued that the poor performance of timeout method is due to the improper threshold value chosen. In this case, the ROC analysis, which assesses the discrimination power of a method's output set independent of the threshold value chosen, is more suitable for performance evaluation.

Gayo-Avello method significantly outperforms the timeout method. But due to its heuristic nature, it is less likely to do better than the supervised-learning methods, although it avoids the over fitting issue. The Gayo-Avello method's unstable performance in predicting session con-

tinuations implies that its heuristics did not generalize well to Chinese query logs.

		Timeout	Gayo-Avello	SVM
P	shift	75.92	89.35	**90.96**
	cont.	63.05	85.32	**92.06**
R	shift	64.49	87.85	**93.82**
	cont.	74.77	87.08	**88.50**
F_1	shift	69.74	88.60	**92.37**
	cont.	68.41	86.19	**90.25**
$F_{1.5}$	shift	67.62	88.31	**92.92**
	cont.	70.72	86.53	**89.57**

Table 3. Precision (P), recall (R), F_1-mean (F_1), and $F_{1.5}$-mean ($F_{1.5}$) of SVM method and the two baseline methods on AOL dataset.

		Timeout	Gayo-Avello	SVM
P	shift	67.75	75.10	**87.53**
	cont.	52.82	**83.51**	81.62
R	shift	59.52	**91.44**	86.17
	cont.	61.53	58.84	**83.33**
F_1	shift	63.37	82.47	**86.85**
	cont.	56.84	69.04	**82.47**
$F_{1.5}$	shift	61.83	85.71	**86.59**
	cont.	58.56	64.72	**82.80**

Table 4. Precision (P), recall (R), F_1-mean (F_1), and $F_{1.5}$-mean ($F_{1.5}$) of SVM method and the two baseline methods on Sogou dataset.

3.4.4 ROC Analysis

By setting certain threshold value, we analyzed the three method's performance using precision / recall based measures. In this sub-subsection, we try to set each value in an output set as the threshold value, and evaluate the discrimination power of methods by the area under the ROC curve.

Figure 1 shows the ROC curves of the SVM method and the two baseline methods: timeout and Gayo-Avello, for predicting session shifts. ROC curves for predicting session continuations are symmetric with respect to the reference line, so we omit them in the rest of this paper for the sake of space limit.

The results show that SVM method clearly outperforms the baseline methods in the prospective of discrimination power, with ROC area 0.9562 on AOL dataset and 0.9154 on Sogou dataset. The curves of the two baseline methods are clearly under that of SVM method. This means baseline methods can never achieve accuracy as high as SVM method w.r.t. a fixed false alarm (classification error) rate, nor false alarm rate as low as SVM method w.r.t. a fixed accuracy rate. Again, Gayo-Avello method significantly outperforms timeout method, while underperforms the SVM method. For the question in the previous sub-subsection, coinciding with previous studies (Murray et al., 2006; Jones and Klinkner, 2008), applying single timeout threshold always yields limited discrimination power, wherever the operating point on ROC curve (i.e. threshold value) is set.

4 Making Use of the Variability of Discrimination Power

In this section, we first analyze the amount of contribution that each feature makes and show that the contribution, i.e. the discrimination power of each feature varies dramatically across different users. Then, we propose an approach to making use of this variability. Finally through experimental results, we show that the proposed approach makes small, yet significant improvements to the SVM method in Section 3.

4.1 Variability of Discrimination Power

The ROC analysis of individual feature provides adequate characterizations of the discrimination power of the feature. Another advantage of adopting ROC analysis is that the results are independent not only of the critical value, but also of the scale of the feature values.

Figure 2 shows the ROC curves of all the eight features in both datasets. Note that some features are with a higher value indicating session continuation rather than session shift, so their ROC curves are below the reference line. The feature "time_interval" behaves exactly the same as the timeout method in Figure 1. For the rest of the features, "avg_ngram_distance", "common_ngram" and "Jaccard_ngram" achieve the best discrimination powers, showing the character-n-gram representation is effective. The feature "common_char" performs significantly better in Sogou dataset than in AOL dataset, because Chinese characters convey much more information than English characters do. "common_suffix" performing worse than "common_prefix" reflects the custom of users. Users tend to add terms at the end of the query in a searching iteration, thus predicting session continuations by examining the common suffixes is problematic.

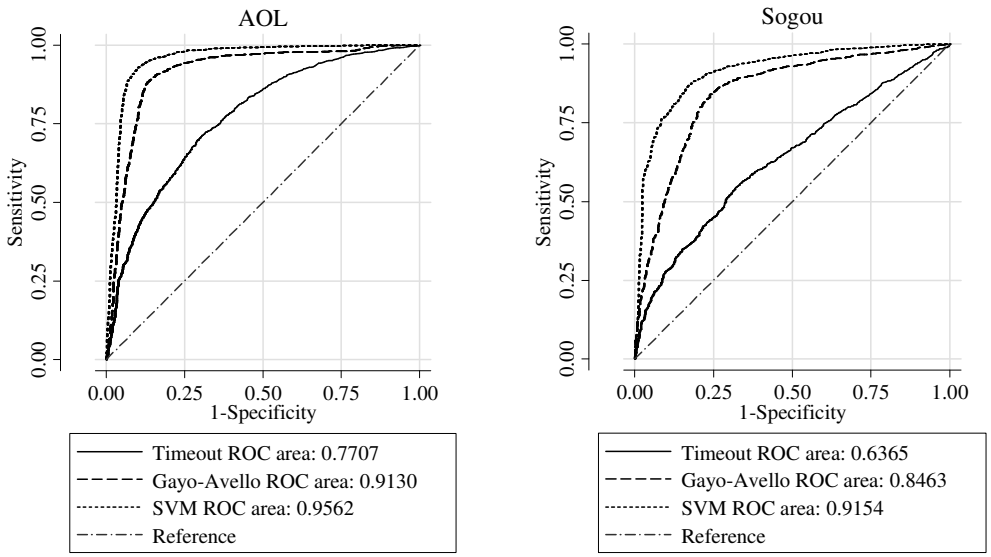


Figure 1. ROC analysis of SVM method and two baseline methods for predicting session shifts on both AOL and Sogou dataset. All comparisons between ROC areas within the same dataset are at least 95% statistically significant, because the corresponding confidence intervals do not overlap.

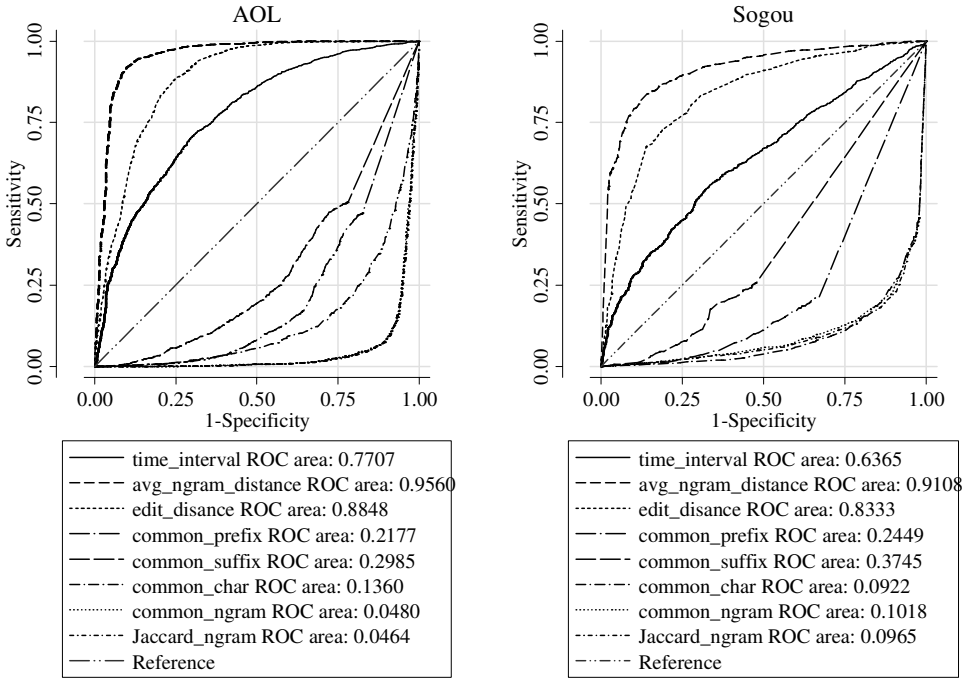


Figure 2. ROC analysis of individual features for predicting session shifts on both AOL and Sogou dataset. Note that some curves with similar ROC area values overlap each other.

In spite of the discrimination power a feature has, its behavior on different users is worthwhile to be examined. For selecting users that have sufficient data to draw stable conclusions, we consider only users who issued more than 50 queries in the datasets. Unfortunately, there are too few users (6 users) qualified in Sogou dataset, so we show only the statistics of ROC area values of each of the features in Table 5 based on 37 users in AOL dataset.

The statistics in Table 5 show that for different users. Recall that in sub-subsection 3.3.2, a 0.04 difference of ROC area make the performance of the SVM method significantly better

than that of the Gayo-Avello's method. Thus, the discrimination power of a feature is likely to vary significantly, because all the standard deviations are at 0.03 or even higher level. Especially, the minimum and maximum values show that for these users, some of the findings above from the whole dataset do not hold. This implies that it is likely more feasible to build specific local models for these users to make full use of the variability within the same feature.

Feature	avg.	sdev.	min.	max.
time_interval	0.780	0.088	0.476	0.912
avg_ngram_ distance	0.954	0.034	0.861	1.000
edit_disance	0.883	0.056	0.733	0.990
common_prefix	0.224	0.069	0.099	0.327
common_suffix	0.299	0.113	0.064	0.578
common_char	0.143	0.082	0.037	0.493
common_ngram	0.051	0.037	0.000	0.187
Jaccard_ngram	0.049	0.036	0.000	0.173

Table 5. Average, standard deviation, minimum, and maximum ROC areas of individual features

4.2 Building Local Models

We built individual local models for each user that issued more than 50 queries in AOL dataset. We also performed 5-fold cross validations and set the prediction to be the probability estimation of a test example being positive. The feature selection process showed again that all the eight features are beneficial, and none of them should be excluded.

In each fold of cross validation, we performed 90%-bagging on the training set 10 times to get the variance estimations of the local model. For each example in the test set, we set the final output on it to be the average of the 10 outputs, and recorded the standard deviation of the outputs on this example which is used during the model combination. We also conducted the same process for the global model for the sake of combination process described below.

4.3 Combing with the Global Model

Since the predictions of both the local and the global models are probability estimations, it is reasonable to combine them using linear combination. For each example, there are two outputs O_l and O_g coming from local and global models accordingly. For each example e of a user's sub dataset U, we have the outputs $O_l(e)$ and $O_g(e)$ as well as the normalized deviations $D_l(e)$ and $D_g(e)$ (by the largest deviation in U of the corresponding models). The final output $O(e)$ is defined as:

$$O(e) = \frac{D_l(e) \cdot O_g(e) + D_l(e) \cdot O_g(e)}{D_l(e) + D_g(e)}$$

		Global	Local	Combine
P	shift	**90.48**	88.53	90.43
	cont.	91.75	92.12	**92.52**
R	shift	93.94	94.44	**94.56**
	cont.	**87.20**	84.16	87.04
F_1	shift	92.18	91.39	**92.45**
	cont.	89.41	87.96	**89.69**
$F_{1.5}$	shift	92.85	92.54	**93.25**
	cont.	88.55	86.46	**88.65**

Table 6. Precision (P), recall (R), F_1-mean (F_1), and $F_{1.5}$-mean ($F_{1.5}$) of global model (bagging), local model (bagging) and combined model

This combination process is similar to (Osl et al., 2008). Note that the more the deviation of a model is, the less feasible the corresponding model is. We compared the performance of three models: global model, local model, and combined model. The results are summarized in Table 6. All comparisons between different models are statistically significant at 95% level, based on the same bootstrapping settings in subsubsection 3.4.3. The combined model shows slight (may due to the inferior performance of the local model), yet significant improvement to the global model. In spite of the amount of the improvement, the local model did correct some errors of the global model. It may be not acceptable to build such an expensive combined model for a limited improvement. Nevertheless, the results do show that the variability across different users is exploitable.

5 Discussion and Conclusion

In this paper, we built a learning framework of detecting sessions which corresponds to user's interest in a query log. We considered two aspect of a pair of successive queries: temporal aspect and content aspect, and designed eight features based on these two aspects, and the SVM models built with these features achieved satisfactory performance (92.37% F_1-mean on session shift, 90.25% F_1-mean on session continuation), significantly better than the best-ever approach on AOL query log.

The analysis of the features' discrimination power was conducted not only among different features, but also within the same feature when applied to different users in the query log. By analyzing the statistics of ROC area values of each of the features based on 37 users in AOL dataset, experimental results showed that there is considerable variability in both these aspects. To make full use of this variability, we built local models for individual user and combine the yielded predictions with those yielded by the global model. Experiments showed that the local model did make significant improvements to the global model, although the amount was small (92.45% vs. 92.18% F_1-mean on session shift, 89.69% vs. 89.41% F_1-mean on session continuation).

In future studies, we will explore other learning frameworks which better integrate the local model and the global model, and will try to acquire more data to build local models. We will also analyze more deeply the characteristics of ROC analysis in the feature selection process.

Acknowledgement

This work is supported by the Key Project of Natural Science Foundation of China (Grant No.60736044), and National 863 Project (Grant No.2006AA010108). The authors are grateful for the anonymous reviewers for their valuable comments.

References

Chang Chih-Chung and Chih-Jen Lin. 2001. LIBSVM : a library for support vector machines. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm

Downey Doug, Susan Dumais, and Eric Horvitz. 2007. Models of searching and browsing: languages, studies, and applications. In *Proceedings of the 20th international joint conference on Artificial intelligence*, pages 2740-2747, Hyderabad, India.

Gayo-Avello Daniel. 2009. A survey on session detection methods in query logs and a proposal for future evaluation, *Information Science* 179(12):1822-1843.

He Daqing and Ayse Göker. 2000. Detecting Session Boundaries from Web User Logs. In *BCS/IRSG 22nd Annual Colloqui-um on Information Retrieval Research*, pages 57-66.

He Daqing, Ayse Göke, and David J. Harper. 2002. Combining evidence for automatic web session identification. *Information Processing and Management: an International Journal*, 38(5):727-742.

Jansen Bernard J., Amanda Spink, Chris Blakely, and Sherry Koshman. 2007. Defining a session on Web search engines: Research Articles. *Journal of the American Society for Information Science and Technology*, 58(6):862-871

Jones Rosie and Kristina Lisa Klinkner. 2008. Beyond the session timeout: automatic hierarchical segmentation of search topics in query logs. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 699-708, Napa Valley, California, USA.

Murray G. Craig, Jimmy Lin, and Abdur Chowdhury. 2007. Identification of user sessions with hierarchical agglomerative clustering. *American Society for Information Science and Technology*, 43(1):1-9.

Osl Melanie, Christian Baumgartner, Bernhard Tilg, and Stephan Dreiseitl. 2008. On the combination of logistic regression and local probability estimates. In *Proceedings of Third International Conference on Broadband Communications, Information Technology & Biomedical Applications*, pages 124-128.

Özmutlu Seda. 2006. Automatic new topic identification using multiple linear regression. *Information Processing and Management*: *an International Journal*, 42(4):934-950.

Özmutlu Huseyin C. 2009. Markovian analysis for automatic new topic identification in search engine transaction logs. *Applied Stochastic Models in Business and Industry*, 25(6):737-768.

Pass Greg, Abdur Chowdhury, and Cayley Torgeson. 2006. A picture of search. In *Proceedings of the 1st international conference on Scalable information systems*, Hong Kong.

Radlinski Filip and Thorsten Joachims. 2005. Query chains: learning to rank from implicit feedback. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 239-248, Chicago, Illinois, USA.

Silverstein Craig, Hannes Marais, Monika Henzinger, and Michael Moricz. 1999. Analysis of a very large web search engine query log. *ACM SIGIR Forum*, 33(1):6-12.

Word-Based and Character-Based Word Segmentation Models: Comparison and Combination

Weiwei Sun
Department of Computational Linguistics, Saarland University
German Research Center for Artificial Intelligence (DFKI)
wsun@coli.uni-saarland.de

Abstract

We present a theoretical and empirical comparative analysis of the two dominant categories of approaches in Chinese word segmentation: word-based models and character-based models. We show that, in spite of similar performance overall, the two models produce different distribution of segmentation errors, in a way that can be explained by theoretical properties of the two models. The analysis is further exploited to improve segmentation accuracy by integrating a word-based segmenter and a character-based segmenter. A *Bootstrap Aggregating* model is proposed. By letting multiple segmenters vote, our model improves segmentation consistently on the four different data sets from the second SIGHAN bakeoff.

1 Introduction

To find the basic language units, i.e. words, segmentation is a necessary initial step for Chinese language processing. There are two dominant models for Chinese word segmentation. The first one is what we call "word-based" approach, where the basic predicting units are words themselves. This kind of segmenters sequentially decides whether the local sequence of characters make up a word. This word-by-word approach ranges from naive maximum matching (Chen and Liu, 1992) to complex solution based on semi-Markov conditional random fields (CRF) (Andrew, 2006). The second is "character-based" approach, where basic processing units are characters which compose words. Segmentation is formulated as a classification problem to predict whether a character locates at the beginning of, inside or at the end of a word. This character-by-character method was first proposed in (Xue, 2003), and a number of sequence labeling algorithms have been exploited.

This paper is concerned with the behavior of different segmentation models in general. We present a theoretical and empirical comparative analysis of the two dominant approaches. Theoretically, these approaches are different. The word-based models do prediction on a dynamic sequence of possible words, while character-based models on a static character sequence. The former models have a stronger ability to represent word token features for disambiguation, while the latter models can better induce a word from its internal structure. For empirical analysis, we implement two segmenters, both using the *Passive-Aggressive* algorithm (Crammer et al., 2006) to estimate parameters. Our experiments indicate that despite similar performance in terms of overall F-score, the two models produce different types of errors, in a way that can be explained by theoretical properties. We will present a detailed analysis that reveals important differences of the two methods in Sec. 4.

The two types of approaches exhibit different behaviors, and each segmentation model has strengths and weaknesses. We further consider integrating word-based and character-based models in order to exploit their complementary strengths and thereby improve segmentation accuracy beyond what is possible by either model in isolation. We present a *Bootstrap Aggregating* model to combine multiple segmentation systems. By

Coling 2008: Poster Volume, pages 1211–1219,
Beijing, August 2010

letting multiple segmenters vote, our combination model improves accuracy consistently on all the four different segmentation data sets from the second SIGHAN bakeoff. We also compare our integrating system to the state-of-the-art segmentation systems. Our system obtains the highest reported F-scores on three data sets.

2 Two Methods for Word Segmentation

First of all, we distinguish two kinds of "words": (1) Words in dictionary are word types; (2) Words in sentences are word tokens. The goal of word segmentation is to identify word tokens in a running text, where a large dictionary (i.e. list of word types) and annotated corpora may be available. From the view of *token*, we divide segmentation models into two main categories: word-based models and character-based models. There are two key points of a segmentation model: (1) How to decide whether a local sequence of characters is a word? (2) How to do disambiguation if ambiguous segmentation occurs? For each model, we separately discuss the strategies for word prediction and segmentation disambiguation.

2.1 Word-Based Approach

It may be the most natural idea for segmentation to find word tokens one by one. This kind of segmenters read the input sentences from left to right, predict whether current piece of continuous characters is a word token. After one word is found, segmenters move on and search for next possible word. There are different strategies for the word prediction and disambiguation problems. Take for example maximum matching, which was a popular algorithm at the early stage of research (Chen and Liu, 1992). For word prediction, if a sequence of characters appears in a dictionary, it is taken as a word candidate. For segmentation disambiguation, if more than one word types are matched, the algorithm chooses the longest one.

In the last several years, machine learning techniques are employed to improve word-based segmentation, where the above two problems are solved in a uniform model. Given a sequence of characters $\mathbf{c} \in \mathcal{C}^n$ (n is the number of characters), denote a segmented sequence of words $\mathbf{w} \in \mathcal{W}^m$ (m is the number of words, i.e. m varies with $\mathbf{w}$), and a function GEN that enumerates a set of segmentation candidates $\mathrm{GEN}(\mathbf{c})$ for $\mathbf{c}$. In general, a segmenter solves the following "argmax" problem:

$$\hat{\mathbf{w}} = \arg \max_{\mathbf{w} \in \mathrm{GEN}(\mathbf{c})} \theta^\top \Phi(\mathbf{c}, \mathbf{w}) \quad (1)$$

$$= \arg \max_{\mathbf{w} \in \mathrm{GEN}(\mathbf{c})} \theta^\top \sum_{i=1}^{|\mathbf{w}|} \phi(\mathbf{c}, w_{[1:i]}) \quad (2)$$

where Φ and ϕ are global and local feature maps and θ is the parameter vector to learn. The inner product $\theta^\top \phi(\mathbf{c}, w_{[1:i]})$ can been seen as the confidence score of whether w_i is a word. The disambiguation takes into account confidence score of each word, by using the sum of local scores as its criteria. Markov assumption is necessary for computation, so ϕ is usually defined on a limited history. Perceptron and semi-Markov CRFs were used to estimate θ in previous work (Zhang and Clark, 2007; Andrew, 2006).

2.2 Character-Based Approach

Most previous data-driven segmentation solutions took an alternative, character-based view. This approach observes that by classifying characters as different positions in words, segmentation can be treated as a sequence labeling problem, assigning labels to the characters in a sentence indicating whether a character c_i is a single character word (*S*) or the begin (*B*), middle (*I*) or end (*E*) of a multi-character word. For word prediction, word tokens are inferred based on the character classes. The main difficulty of this model is character ambiguity that most Chinese characters can occur in different positions within different words. Linear models are also popular for character disambiguation (i.e. segmentation disambiguation). Denote a sequence of character labels $\mathbf{y} \in \mathcal{Y}^n$, a linear model is defined as:

$$\hat{\mathbf{y}} = \arg \max_{\mathbf{y} \in \mathcal{Y}^{|\mathbf{c}|}} \theta^\top \Psi(\mathbf{c}, \mathbf{y}) \quad (3)$$

$$= \arg \max_{\mathbf{y} \in \mathcal{Y}^{|\mathbf{c}|}} \theta^\top \sum_{i=1}^{|\mathbf{c}|} \psi(\mathbf{c}, y_{[1:i]}) \quad (4)$$

Note that local feature map ψ is defined only on the sequence of characters and their labels.

Several discriminative models have been exploited for parameter estimation, including perceptron, CRFs, and discriminative latent variable CRFs (Jiang et al., 2009; Tseng, 2005; Sun et al., 2009b).

2.3 Theoretical Comparison

Theoretically, the two types of models are different. We compare them from four aspects.

2.3.1 Internal Structure of Words

Chinese words have internal structures. In most cases, Chinese character is a morpheme which is the smallest meaningful unit of the language. Though we cannot exactly infer the meaning of a word from its character components, the character structure is still meaningful. Partially characterizing the internal structures of words, one advantage of character-based models is the ability to induce new words. E.g., character "者/person" is usually used as a suffix meaning "one kind of people". If a segmenter never sees "工作者/worker" in training data, it may still rightly recognize this word by analyzing the prefix "工作/work" with label *BI* and the suffix "者" with label *E*. In contrast, current word-based models only utilize the weighted features as word prediction criteria, and thus word formation information is not well explored. For more details about Chinese word fomation, see (Sun et al., 2009a).

2.3.2 Linearity and Nonlinearity

A majority of structured prediction models are linear models in the sense that the score functions are linear combination of parameters. Both previous solutions for word-based and character-based systems utilize linear models. However, both "linear" models incur nonlinearity to some extent. In general, a sequence classification itself involves nonlinearity in a way that the features of current token usually encode previous state information which is linear combination of features of previous tokens. The interested readers may consult (Liang et al., 2008) for preliminary discussion about the nonlinearity in structured models. This kind of nonlinearity exists in both word-based and character-based models. In addition, in most character-based models, a word should take a *S* label or start with a *B* label, end with *E* label, and only have *I* label inside. This inductive way for word prediction actually behaves nonlinearly.

2.3.3 Dynamic Tokens or Static Tokens

Since word-based models take the sum of part score of each individual word token, it increases the upper bound of the whole score to segment more words. As a result, word-based segmenter tends to segment words into smaller pieces. A difficult case occurs when a word token w consists of some word types which could be separated as words on their own. In such cases a word-based segmenter more easily splits the word into individual words. For example, in the phrase "四千三百/4300 米/meter (4300 meters)", the numeral "四千三百" consists of two individual numeral types "四千 (4000)" and "三百(300)". A word-based segmenter more easily made a mistake to segment two word tokens. This phenomenon is very common in named entities.

2.3.4 Word Token or Word Type Features

In character-based models, features are usually defined by the character information in the neighboring n-character window. Despite a large set of valuable features that could be expressed, it is slightly less natural to encode predicted word token information. On the contrary, taking words as dynamic tokens, it is very easy to define word token features in a word-based model. Word-based segmenters hence have greater representational power. Despite of the lack of word token representation ability, character-based segmenters can use word type features by looking up a dictionary. For example, if a local sequence of characters following current token matches a word in a dictionary; these word types can be used as features. If a string matches a word type, it has a very high probability (ca. 90%) to be a word token. So word type features are good approximation of word token features.

3 Baseline Systems

For empirical analysis, we implement segmenters in word-based and character-based architectures respectively. We introduce them from three aspects: basic models, parameter estimation and feature selection.

Algorithm 1: The *PA* learning procedure.

input : Data $\{(\mathbf{x}_t, \mathbf{y}_t), t = 1, 2, ..., n\}$

1 Initialize: $\mathbf{w} \leftarrow (0, ..., 0)$
2 **for** $I = 1, 2, ...$ **do**
3 **for** $t = 1, ..., n$ **do**
4 Predict: $\mathbf{y}_t^* = \arg\max_{\mathbf{y} \in \text{GEN}(\mathbf{x}_t)} \mathbf{w}^\top \Phi(\mathbf{x}_t, \mathbf{y})$
5 Suffer loss: $l_t = \rho(\mathbf{y}_t, \mathbf{y}_t^*) + \mathbf{w}^\top \Phi(\mathbf{x}_t, \mathbf{y}_t^*) - \mathbf{w}^\top \Phi(\mathbf{x}_t, \mathbf{y}_t)$
6 Set: $\tau_t = \frac{l_t}{||\Phi(\mathbf{x}_t, \mathbf{y}_t^*) - \Phi(\mathbf{x}_t, \mathbf{y}_t)||^2 + 0.5C}$
7 Update: $\mathbf{w} \leftarrow \mathbf{w} + \tau_t(\Phi(\mathbf{x}_t, \mathbf{y}_t) - \Phi(\mathbf{x}_t, \mathbf{y}_t^*))$
8 **end**
9 **end**

3.1 Models

For both word-based and character-based segmenters, we use linear models introduced in the section above. We use a first order Markov models for training and testing. In particular, for word-based segmenter, the local feature map $\phi(\mathbf{c}, w_{[1:i]})$ is defined only on $\mathbf{c}, w_{i-1}$ and w_i, and thereby Eq. 2 is defined as $\hat{\mathbf{w}} = \arg\max_{\mathbf{w} \in \text{GEN}(\mathbf{c})} \theta^\top \sum_{i=1}^{|\mathbf{w}|} \phi(\mathbf{c}, w_{i-1}, w_i)$. This model has a first-order Semi-Markov structure. For decoding, Zhang and Clark (2007) used a beam search algorithm to get approximate solutions, and Sarawagi and Cohen (2004) introduced a Viterbi style algorithm for exact inference. Since the exact inference algorithm is efficient enough, we use this algorithm in our segmenter at both training and testing time.

For our character-based segmenter, the local feature map $\psi(\mathbf{c}, y_{[1:i]})$ is defined on $\mathbf{c}, y_{i-1}$ and y_i, and Eq. 4 is defined as $\hat{\mathbf{y}} = \arg\max_{\mathbf{y} \in \mathcal{Y}^{|\mathbf{c}|}} \theta^\top \sum_{i=1}^{|\mathbf{c}|} \psi(\theta, y_{i-1}, y_i)$. In our character-based segmenter, we also use a Viterbi algorithm for decoding.

3.2 Learning

We adopt *Passive-Aggressive* (PA) framework (Crammer et al., 2006), a family of margin based online learning algorithms, for the parameter estimation. It is fast and easy to implement. Alg. 1 illustrates the learning procedure. The parameter vector $\mathbf{w}$ is initialized to $(0, ..., 0)$. A *PA* learner processes all the instances (t is from 1 to n) in each iteration (I). If current hypothesis ($\mathbf{w}$) fails to predict $\mathbf{x}_t$, the learner update $\mathbf{w}$ through calculating the loss l_t and the difference between $\Phi(\mathbf{x}_t, \mathbf{y}_t^*)$ and $\Phi(\mathbf{x}_t, \mathbf{y}_t)$ (line 5-7). There are three variants in the update step. We here only present the PA-II rule[1], which performs best in our experiments.

The PA algorithm utilizes a paradigm of cost-sensitive learning to resolve structured prediction. A cost function ρ is necessary to calculate the loss l_t (line 5). For every pair of labels $(\mathbf{y}^*, \mathbf{y})$, users should define a cost $\rho(\mathbf{y}^*, \mathbf{y})$ associated with predicting $\mathbf{y}^*$ when the correct label is $\mathbf{y}$. ρ should be defined differently for different purposes. There are two natural costs for segmentation: (1) sum of the number of wrong and missed word predictions and (2) sum of the number of wrongly classified characters. We tried both cost functions for both models. We find that the first one is suitable for word-based segmenter and the second one is suitable for character-based segmenter. We do not report segmentation performance with "weaker" cost in later sections. C (in line 6) is the slack variable. In our experiments, the segmentation performance is not sensitive to C. In the following experiments, we set $C = 1$.

3.3 Features

3.3.1 Word-based Segmenter

For the convenience of illustration, we denote a candidate word token w_i with a context $c_{j-1}[w_{i-1}c_j...c_k][w_i c_{k+1}...c_l]c_{l+1}$.

The character features includes,

Boundary character unigram: c_j, c_k, c_{k+1}, c_l and c_{l+1}; *Boundary character bigram*: $c_k c_{k+1}$ and $c_l c_{l+1}$.

Inside character unigram: c_s $(k+1 < s < l)$; *Inside character bigram*: $c_s c_{s+1}$ $(k+1 < s < l)$.

Length of current word.

Whether c_{k+1} and c_{k+1} are *identical*.

Combination Features: c_{k+1} and c_l,

The word token features includes,

Word Unigram: previous word w_{i-1} and current word w_i; *Word Bigram*: $w_{i-1}w_i$.

[1] See the original paper for more details.

The *identity* of w_i, if it is a *Single character word*.

Combination Features: w_{i-1} and length of w_i, w_i and length of w_{i-1}. c_{k+1} and length of w_i, c_l and length of w_i.

3.3.2 Character-based Segmenter

We use the exact same feature templates discribed in (Sun et al., 2009b). The features are divided into two types: character features and word type features. Note that the word type features are indicator functions that fire when the local character sequence matches a word unigram or bigram. Dictionaries containing word unigrams and bigrams was collected from the training data. Limited to the document length, we do not give the discription for the features. We suggest readers to refer to the original paper for details.

4 Empirical Analysis

We present a series of experiments that relate segmentation performance to a set of properties of input words. We argue that the results can be correlated to specific theoretical aspects of each model.

4.1 Experimental Setting

We used the data provided by the second SIGHAN Bakeoff (Emerson, 2005) to test the two segmentation models. The data contains four corpora from different sources: Academia Sinica Corpus (AS), City University of Hong Kong (CU), Microsoft Research Asia (MSR), and Peking University (PKU). There is no fixed standard for Chinese word segmentation. The four data sets above are annotated with different standards. To catch general properties, we do experiments on all the four data sets. Three metrics were used for evaluation: precision (P), recall (R) and balanced F-score (F) defined by 2PR/(P+R).

4.2 Baseline Performance

Tab. 1 shows the performance of our two segmenters. Numbers of iterations are respectively set to 15 and 20 for our word-based segmenter and character-based segmenter. The word-based segmenter performs slightly worse than the character-based segmenter. This is different from the experiments reported in (Zhang and Clark, 2007). We think the main reason is that we use a different learning architecture.

	Model	P(%)	R(%)	F
AS	Character	94.8	94.7	94.7
	Word	93.5	94.8	94.2
CU	Character	95.5	94.6	95.0
	Word	94.4	94.7	94.6
MSR	Character	96.1	96.5	96.3
	Word	96.0	96.3	96.1
PKU	Character	94.6	94.9	94.8
	Word	94.7	94.3	94.5

Table 1: Baseline performance.

4.3 Word Frequency Factors

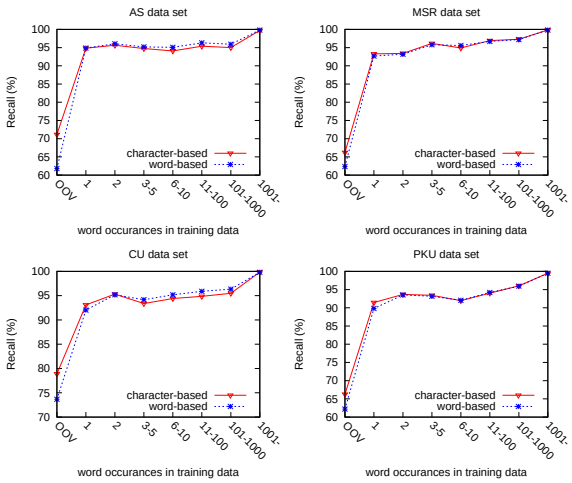


Figure 1: Segmentation recall relative to gold word frequency.

Our theoretical analysis also suggests that character-based has stronger word induction ability because it focuses more on word internal structures and thereby expresses more nonlinearity. To test the word induction ability, we present the recall relative to word frequency. If a word appears in a training data many times, the learner usually works in a "memorizing" way. On the contrary, infrequent words should be correctly recognized in a somehow "inductive" way. Fig. 1 shows the recall change relative to word frequency in each training data. Note that, the words with frequency 0 are out-of-vocabulary (OOV) words. We can clearly see that character-based model outperforms word-based model for infrequent word, especially OOV words, recognition. The "memoriz-

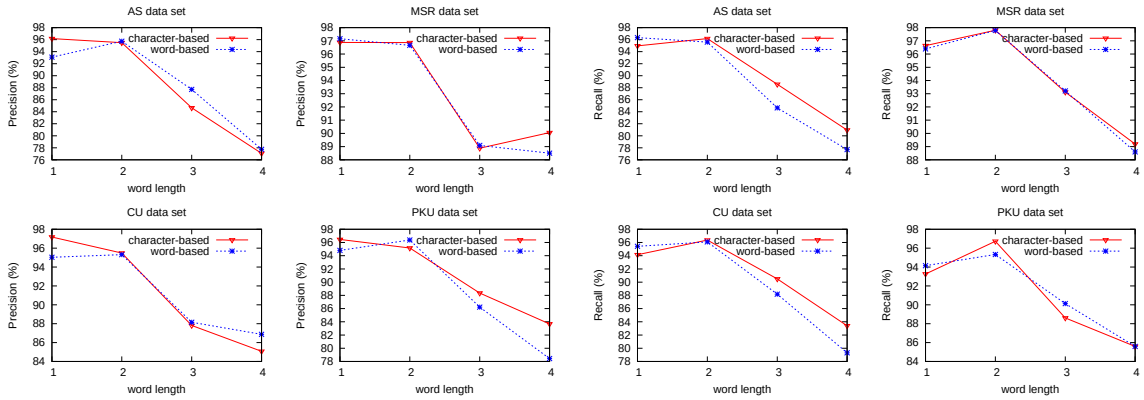


Figure 2: Segmentation precision/recall relative to gold word length in training data.

ing" ability of the two models is similar; on the AS and CU data sets, the word-based model performs slightly better. Neither model is robust enough to reliably segment unfamiliar words. The recall of OOV words is much lower than in-vocabulary words.

4.4 Length Factors

Length	AS	CU	MSR	PKU
1	61254	19116	48092	45911
2	52268	18186	49472	49861
3	6990	2682	4652	5132
4	1417	759	2711	2059
5(+)	690	193	1946	656

Table 2: Word length statistics on test sets.

Tab. 2 shows the statistics of word counts relative to word length on each test data sets. There are much less words with length more than 4. Analysis on long words may not be statistical significant, so we only present length factors on small words (length is less than 5). Fig. 2 shows the precision/recall of both segmentation models relative sentence length. We can see that word-based model tends to predict more single character words, but making more mistakes. Since about 50% word tokens are single-character words, this is one main source of error for word-segmenter. This can be explained by theoretical properties of dynamic token prediction discussed in Sec. 2.3.3. The score of a word boundary assignment in a word-based segmenter is defined like $\theta^\top \sum_{i=1}^{|\mathbf{w}|} \phi(\mathbf{c}, w_{[1:i]})$. The upper bound of this score varies with the length $|\mathbf{w}|$. If a segmentation result is with more fragments, i.e. $|\mathbf{w}|$ is larger, the upper bound of its score is higher. As a result, in many cases, a word-based segmenter prefers shorter words, which may cause errors.

4.5 Feature Factors

We would like to measure the effect of features empirically. In particular, we do not use dynamic *word token features* in our word-based segmenter, and *word type features* in our character-based segmenter as comparison with "standard" segmenters. The difference in performance can be seen as the contribution of *word features*. There are obvious drops in both cases. Though it is not a fair comparison, word token features seem more important, since the numerical decrease in the word-based experiment is larger.

	word-based		character-based	
	$-$	$+$	$-$	$+$
AS	93.1	94.2	94.1	94.7
CU	92.6	94.6	94.2	95.0
MSR	95.7	96.1	95.8	96.3
PKU	93.3	94.5	94.4	94.8

Table 3: F-score of two segmenters, with $(-)$ and without $(+)$ *word token/type features*.

4.6 Discussion

The experiments highlight the fundamental difference between word-based and character-based models, which enlighten us to design new models. The above analysis indicates that the theoretical differences cause different error distribution.

The two approaches are either based on a particular view of segmentation. Our analysis points out several drawbacks of each one. It may be helpful for both models to overcome their shortcomings. For example, one weakness of word-based model is its word induction ability which is partially caused by its neglect of internal structure of words. A word-based model may be improved by solving this problem.

5 System Combination

The error analysis also suggests that there is still space for improvement, just by combining the two existing models. Here, we introduce a classifier ensemble method for system combination.

5.1 Upper Bound of System Combination

To get an upper bound of the improvement that can be obtained by combining the strengths of each model, we have performed an oracle experiment. We think the optimal combination system should choose the right prediction when the two segmenters do not agree with each other. There is a *gold segmenter* that generates gold-standard segmentation results. In the oracle experiment, we let the three segmenters, i.e. baseline segmenters and the gold segmenter, vote. The three segmenters output three segmentation results, which are further transformed into IOB2 representation (Ramshaw and Marcus, 1995). Namely, each character has three *B* or *I* labels. We assign each character an oracle label which is chosn by at least two segmenters. When the baseline segmenters are agree with each other, the gold segmenter cannot change the segmentation whether it is right or wrong. In the situation that the two baseline segmenters disagree, the vote given by the gold segmenter will decide the right prediction. This kind of optimal performance is presented in Tab. 4. Compared these results with Tab. 1, we see a significant increase in accuracy for the four data sets. The upper bound of error reduction with system combination is over 30%.

5.2 Our Model

Bootstrap aggregating (Bagging) is a machine learning ensemble meta-algorithm to improve classification and regression models in terms of stability and classification accuracy (Breiman, 1996). It also reduces variance and helps to avoid overfitting. Given a training set D of size n, Bagging generates m new training sets D_i of size $n' \leq n$, by sampling examples from D uniformly. The m models are fitted using the above m bootstrap samples and combined by voting (for classification) or averaging the output (for regression).

We propose a Bagging model to combine multiple segmentation systems. In the training phase, given a training set D of size n, our model generates m new training sets D_i of size $63.2\% \times n$ by sampling examples from D without replacement. Namely no example will be repeated in each D_i. Each D_i is separately used to train a word-based segmenter and a character-based segmenter. Using this strategy, we can get $2m$ *weak* segmenters. Note that the sampling strategy is different from the standard one. Our experiment shows that there is no significant difference between the two sampling strategies in terms of accuracy. However, the *non-placement* strategy is more efficient. In the segmentation phase, the $2m$ models outputs $2m$ segmentation results, which are further transformed into IOB2 representation. In other words, each character has $2m$ *B* or *I* labels. The final segmentation is the voting result of these $2m$ labels. Note that since $2m$ is an even number, there may be equal number of *B* and *I* labels. In this case, our system prefer *B* to reduce error propagation.

	P(%)	R(%)	F	ER (%)
AS	96.6	96.9	96.7	37.7
CU	97.4	97.1	97.3	46.0
MSR	97.5	97.7	97.6	35.1
PKU	96.8	96.2	96.5	32.7

Table 4: Upper bound for combination. The error reduction (ER) rate is a comparison between the F-score produced by the oracle combination system and the character-based system (see Tab. 1).

5.3 Results

Fig. 4 shows the influence of m in the bagging algorithm. Because each new data set D_i in bagging algorithm is generated by a random procedure, the performance of all bagging experiments are not the same. To give a more stable evaluation, we repeat 5 experiments for each m and show the

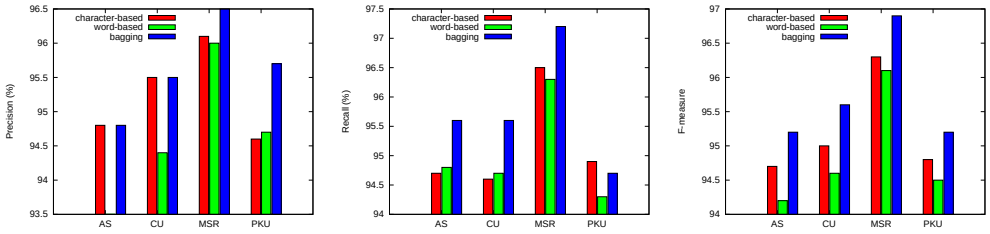


Figure 3: Precision/Recall/F-score of different models.

averaged F-score. We can see that the bagging model taking two segmentation models as basic systems consistently outperform the baseline systems and the bagging model taking either model in isolation as basic systems. An interesting phenomenon is that the bagging method can also improve word-based models. In contrast, there is no significant change in character-based models.

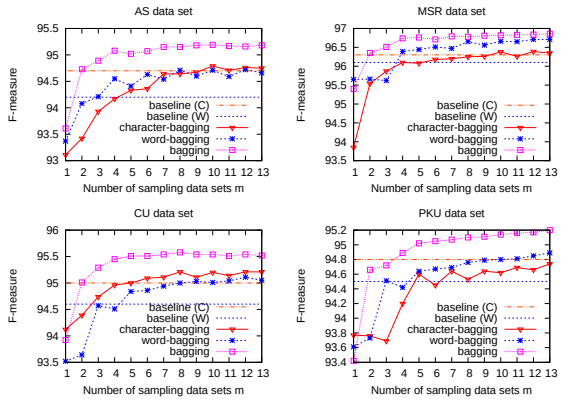


Figure 4: F-score of bagging models with different numbers of sampling data sets. *Character-bagging* means that the bagging system built on the single character-based segmenter. *Word-bagging* is named in the same way.

Fig. 3 shows the precision, recall, F-score of the two baseline systems and our final system for which we generate $m = 15$ new data sets for bagging. We can see significant improvements on the four datasets in terms of the balanced F-score. The improvement of precision and recall are not consistent. The improvement of AS and CU datasets is from the recall improvement; the improvement of PKU datasets is from the precision improvement. We think the different performance is mainly because the four datasets are annotated by using different standards.

	AS	CU	MSR	PKU
(Zhang et al., 2006)	95.1	95.1	97.1	95.1
(Zhang and Clark, 2007)	94.6	95.1	97.2	94.5
(Sun et al., 2009b)	N/A	94.6	**97.3**	**95.2**
This paper	**95.2**	**95.6**	96.9	**95.2**

Table 5: Segmentation performance presented in previous work and of our combination model.

Tab. 5 summarizes the performance of our final system and other systems reported in a majority of previous work. The left most column indicates the reference of previous systems that represent state-of-the-art results. The comparison of the accuracy between our integrating system and the state-of-the-art segmentation systems in the literature indicates that our combination system is competitive with the best systems, obtaining the highest reported F-scores on three data sets.

6 Conclusion

We have presented a thorough study of the difference between word-based and character-based segmentation approaches for Chinese. The theoretical and empirical analysis provides insights leading to better models. The strengths and weaknesses of the two methods are not exactly the same. To exploit their complementary strengths, we propose a Bagging model for system combination. Experiments show that the combination strategy is helpful.

Acknowledgments

The work is supported by the project TAKE (Technologies for Advanced Knowledge Extraction), funded under contract 01IW08003 by the German Federal Ministry of Education and Research. The author is also funded by German Academic Exchange Service (DAAD).

References

Galen Andrew. 2006. A hybrid markov/semi-markov conditional random field for sequence segmentation. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 465–472. Association for Computational Linguistics, Sydney, Australia.

Leo Breiman. 1996. Bagging predictors. *Machine Learning*, 24(2):123–140.

Keh-Jiann Chen and Shing-Huan Liu. 1992. Word identification for mandarin Chinese sentences. In *Proceedings of the 14th conference on Computational linguistics*, pages 101–107. Association for Computational Linguistics, Morristown, NJ, USA.

Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. 2006. Online passive-aggressive algorithms. *JOURNAL OF MACHINE LEARNING RESEARCH*, 7:551–585.

Thomas Emerson. 2005. The second international Chinese word segmentation bakeoff. In *In Proceedings of the Second SIGHAN Workshop on Chinese Language Processing*, pages 123–133.

Wenbin Jiang, Liang Huang, and Qun Liu. 2009. Automatic adaptation of annotation standards: Chinese word segmentation and pos tagging – a case study. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 522–530. Association for Computational Linguistics, Suntec, Singapore.

Percy Liang, Hal Daumé, III, and Dan Klein. 2008. Structure compilation: trading structure for features. In *ICML '08: Proceedings of the 25th international conference on Machine learning*, pages 592–599. ACM, New York, NY, USA.

L. A. Ramshaw and M. P. Marcus. 1995. Text chunking using transformation-based learning. In *Proceedings of the 3rd ACL/SIGDAT Workshop on Very Large Corpora, Cambridge, Massachusetts, USA*, pages 82–94.

Sunita Sarawagi and William W. Cohen. 2004. Semi-markov conditional random fields for information extraction. In *In Advances in Neural Information Processing Systems 17*, pages 1185–1192.

Weiwei Sun, Zhifang Sui, Meng Wang, and Xin Wang. 2009a. Chinese semantic role labeling with shallow parsing. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1475–1483. Association for Computational Linguistics, Singapore.

Xu Sun, Yaozhong Zhang, Takuya Matsuzaki, Yoshimasa Tsuruoka, and Jun'ichi Tsujii. 2009b. A discriminative latent variable Chinese segmenter with hybrid word/character information. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 56–64. Association for Computational Linguistics, Boulder, Colorado.

Huihsin Tseng. 2005. A conditional random field word segmenter. In *In Fourth SIGHAN Workshop on Chinese Language Processing*.

Nianwen Xue. 2003. Chinese word segmentation as character tagging. In *International Journal of Computational Linguistics and Chinese Language Processing*.

Ruiqiang Zhang, Genichiro Kikui, and Eiichiro Sumita. 2006. Subword-based tagging by conditional random fields for Chinese word segmentation. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, pages 193–196. Association for Computational Linguistics, New York City, USA.

Yue Zhang and Stephen Clark. 2007. Chinese segmentation with a word-based perceptron algorithm. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 840–847. Association for Computational Linguistics, Prague, Czech Republic.

Confidence Measures for Error Discrimination in an Interactive Predictive Parsing Framework[1]

Ricardo Sánchez-Sáez, Joan Andreu Sánchez and José Miguel Bened
Instituto Tecnológico de Informática
Universidad Politécnica de Valencia
{rsanchez,jandreu,jbenedi}@dsic.upv.es

Abstract

We study the use of Confidence Measures (CM) for erroneous constituent discrimination in an Interactive Predictive Parsing (IPP) framework. The IPP framework allows to build interactive tree annotation systems that can help human correctors in constructing error-free parse trees with little effort (compared to manually post-editing the trees obtained from an automatic parser). We show that CMs can help in detecting erroneous constituents more quickly through all the IPP process. We present two methods for precalculating the confidence threshold (globally and per-interaction), and observe that CMs remain highly discriminant as the IPP process advances.

1 Introduction

Within the Natural Language Processing (NLP) field, we can tell apart two different usage scenarios for automatic systems that output or work with natural language. On one hand, we have the cases in which the output of such systems is expected to be used in a vanilla fashion, that is, without validating or correcting the results produced by the system. Within this usage scheme, the most important factor of a given automatic system is the quality of the results. Although memory and computational requirements of such systems are usually taken into account, the ultimate aim of most research that relates to this scenario is to minimize the amount of error (measured with metrics like Word Error Rate, BLEU, F-Measure, etc.) present within the results that are being produced.

The second usage scenario arises when there exists the need for perfect and completely error-free results, for example, flawlessly translated sentences or correctly annotated syntactic trees. In such cases, the intervention of a human validator/corrector is unavoidable. The corrector will review and validate the results, making the suitable modifications before the system output can be employed. In these kind of tasks, the most important factor to be minimized is the human effort that has to be applied to transform the system's potentially incorrect output into validated and error-free output. Measuring user effort has an intrinsic subjectivity that makes it hard to be quantitatized. Given that the user effort is usually inversely proportional to the quality of the system output, most research about problems associated to this scenario t to minimize just the system's error rate as well.

Interactive Predictive NLP Systems

Only recently, more comparable and reproducible evaluation methods for Interactive Natural Language Systems have started to be developed, within the context of Interactive Predictive Systems (IPS). These systems formally integrate the correcting user into the loop, making him part of the system right at its theoretical framework. IPSs allow for human correctors to spare effort because the system updates its output after each individual user correction, potentially fixing several errors at each step. Interactive Predictive methods have been studied and successfully used in fields

[1]Work partially supported by the Spanish MICINN under the MIPRCV "Consolider Ingenio 2010" (CSD2007-00018), MITTRAL (TIN2009-14633-C03-01), Prometeo (PROMETEO/2009/014) research projects, and the FPU fellowship AP2006-01363.

Coling 2008: Poster Volume, pages 1220–1228,
Beijing, August 2010

like Handwritten Text Recognition (HTR) (Toselli et al., 2008) and Statistical Machine Translation (SMT) (Vidal et al., 2006; Barrachina et al., 2009) to ease the work of transcriptors and translators.

In IPS related research the importance of the system base error rate *per se* is diminished. Instead, the intention is to measure how well the user and the system work together. For this, formal user simulation protocols together with new objective effort evaluation metrics such as the Word Stroke Ratio (WSR) (Toselli et al., 2008) or the Key-Stroke and Mouse-Ratio (KSMR) (Barrachina et al., 2009) started to be used as a benchmark. These ratios reflect the amount of user effort (whole-word corrections in the case of WSR; keystrokes plus mouse actions in the case of KSMR) given a certain output. To get the amount of user effort into context they should be measured against the corresponding error ratios of comparable non-interactive systems: Word Error Rate for WSR and Character Error Rate for KSMR.

This dichotomy in evaluating either system performance or user effort applies to Syntactic Parsing as well. The objective of parsing is to precisely determine the syntactic structure of sentences written in one of the several languages that humans use. Very bright research has been carried out in this field, resulting in several top performing completely automatic parsers (Collins, 2003; Klein and Manning, 2003; McClosky et al., 2006; Huang, 2008; Petrov, 2010). However, these produce results that are erroneous to some extent, and as such unsuitable for some applications without a previous manual correction. There are many problems where error-free results consisting in perfectly annotated trees are needed, such as handwritten mathematical expression recognition (Yamamoto et al., 2006) or construction of large new gold treebanks (de la Clergerie et al., 2008).

When using automatic parsers as a baseline for building perfect syntactic trees, the role of the human annotator is usually to post-edit the trees and correct the errors. This manner of operating results in the typical two-step process for error correcting, in which the system first generates the whole output and then the user verifies or amends it. This paradigm is rather inefficient and uncomfortable for the human annotator. For example, a basic two-stage setup was employed in the creation of the Penn Treebank annotated corpus: a rudimentary parsing system provided a skeletal syntactic representation, which then was manually corrected by human annotators (Marcus et al., 1994). Additional works within this field have presented systems that act as a computerized aid to the user in obtaining the perfect annotation (Carter, 1997; Oepen et al., 2004; Hiroshi et al., 2005). Subjective measuring of the effort needed to obtain perfect annotations was reported in some of these works, but we feel that a more comparable metric is needed.

With the objective of reducing the user effort and making the laborious task of tree annotation easier, the authors of (Sánchez-Sáez et al., 2009a) devised an Interactive Predictive Parsing (IPP) framework. That work embeds the human corrector into the automatic parser, and allows him to interact in real time within the system. In this manner, the system can use the readily available user feedback to make predictions about the parts of the trees that have not been validated by the corrector. The authors simulated user interaction and calculated effort evaluation metrics, establishing that an IPP system results in amounts slightly above 40% of effort reduction for a manual annotator compared to a two-step system.

Confidence Measures in NLP

Annotating trees syntactically, even with the aid of automatic systems, generally requires human intervention with a high degree of specialization. This fact partially justifies the shortage in large manually annotated treebanks. Endeavors directed at easing the burden for the experts performing this task could be of great help.

One approach that can be followed in reducing user effort within an IPS is adding information that helps the user to locate the individual errors in a sentence, so he can correct them in a hastier fashion. The use of the Confidence Measure (CM) formalism goes in this direction, allowing us to assign a *probability of correctness* for individual erroneous constituents of a more complex output block of a NLP system.

In fields such as HTR, SMT or Automatic Speech Recognition (ASR), the output sentences

have a global probability (or score) that reflects the likeness of the output sentence being correct. CMs allow precision beyond the sentence level in predicting errors: they can be used to label the individual words as either correct or incorrect. Automatic systems can use CMs to help the user in identifying the erroneous parts of the output in a faster way or to aid with the amendments by suggesting replacement words that are likely to be correct.

Previous research shows that CMs have been successfully applied within the ASR (Wessel et al., 2001), HTR (Tarazón et al., 2009; Serrano et al., 2010) and SMT (Ueffing and Ney, 2007) fields. In these works, the ability of CMs in detecting erroneous constituents is assessed by the classical confidence metrics: the Confidence Error Rate (CER) and the Receiver Operating Characteristic (ROC) (Ueffing and Ney, 2007).

However, until recent advances, the use of CMs remained largely unexplored in Parsing. Assessing the correctness of the different parts of a parsing tree can be useful in improving the efficiency and usability of an IPP system, not only by *tagging* parts with low confidence for the user to review, but also by automating part of the correction process itself by presenting constituents that yield a higher confidence when an error is confirmed by the user.

CMs for parsing in the form of combinations of features calculated from n-best lists were proposed in (Benedí et al., 2007). Later on, the authors of (Sánchez-Sáez et al., 2009b) introduced a statistical method for calculating a CM for each of the constituents in a parse tree. In that work, CMs are calculated using the posterior probability of each tree constituent, approach which is similar to the word-graph based methods in the ASR and SMT fields.

In this paper, we apply Confidence Measures to the Interactive Predictive Parsing framework to asses how CMs are increasingly more accurate as the user validates subtrees within the interactive process. We prove that after each correction performed by the user, the CMs of the remaining unvalidated constituents are more helpful to detect errors.

2 Interactive Predictive Parsing

In this section we review the IPP framework (Sánchez-Sáez et al., 2009a) and its underlying operation protocol. In parsing, a syntactic tree t, attached to a string $\boldsymbol{x} = x_1 \dots x_{|x|}$ is composed by substructures called constituents. A constituent c_{ij}^A is defined by the nonterminal symbol (either a *syntactic label* or a *POS tag*) A and its span ij (the starting and ending indexes which delimit the part of the input sentence encompassed by the constituent).

Here follows a general formulation for the non-interactive syntactic parsing scenario, which will allow us to better introduce the IPP formulation. Assume that using a given parsing model G, the parser analyzes the input sentence $\boldsymbol{x}$ and produces the most probable parse tree

$$\hat{t} = \arg\max_{t \in \mathcal{T}} p_G(t|\boldsymbol{x}), \tag{1}$$

where $p_G(t|\boldsymbol{x})$ is the probability of the parse tree t given the input string $\boldsymbol{x}$ using model G, and $\mathcal{T}$ is the set of all possible parse trees for $\boldsymbol{x}$.

In the IPP framework, the manual corrector provides feedback to the system by correcting any of the constituents c_{ij}^A from $\hat{t}$. The system reacts to each of the corrections performed by the human annotator by proposing a new $\hat{t}'$ that takes into account the correction.

Within the IPP framework, the user reviews the constituents contained in the tree to assess their correctness. When the user finds an incorrect constituent he modifies it, setting the correct span and label. This action implicitly validates what it is called the *validated prefix tree* t_p.

We define the validated prefix tree to be composed by the partially corrected constituent, all of its ancestor constituents, and all constituents whose end span is lower than the start span of the corrected constituent. When the user replaces the constituent c_{ij}^A with the correct one $c_{ij}'^A$, the validated prefix tree is

$$\begin{aligned} t_p(c_{ij}'^A) = \{ & c_{mn}^B \ : m \leq i,\ n \geq j\,, \\ & \qquad d(c_{mn}^B) \leq d(c_{ij}'^A)\} \cup \\ & \{c_{pq}^D \ : q < i\ \} \end{aligned} \tag{2}$$

with $d(c^Z_{ab})$ being the depth (distance from root) of constituent c^Z_{ab}.

The validated prefix tree is parallel to the validated sentence prefix commonly used in Interactive Machine Translation or Interactive Handwritten Recognition, and is established after each user action.

This particular definition of the prefix tree determines the fact that the user is expected to review the parse tree in a preorder fashion (left-to-right depth-first). Note that this specific exploration order allows us to simulate the user interaction for the experimentation, as we will explain below. Also note that other types of prefixes could be defined, allowing for different tree review orders.

Within the IPP formulation, when a constituent correction is performed, the prefix tree $t_p(c'^A_{ij})$ is validated and a new tree $\hat{t}'$ that takes into account the prefix is proposed. Incorporating this new evidence into expression (1) yields the following equation

$$\hat{t}' = \arg\max_{t \in \mathcal{T}} p_G(t|\boldsymbol{x}, t_p(c'^A_{ij})). \quad (3)$$

Given the properties of Probabilistic Context-Free Grammars (PCFG) the only subtree that effectively needs to be recalculated is the one starting from the parent of the corrected constituent. This way, just the descendants of the newly introduced constituent, as well as its right hand siblings (along with their descendants) are calculated.

2.1 User Interaction Operation

The IPP formulation allows for a very straightforward operation protocol that is performed by the manual corrector, in which he validates or corrects the successive output parse trees:

1. The IPP system proposes a full parse tree t for the input sentence.

2. Then, the user finds the first incorrect constituent exploring the tree in a certain ordered manner (preorder in our case, given by the tree prefix definition) and amends it, by modifying its span and/or label (implicitly validating the prefix tree t_p).

3. The IPP system produces the most probable tree that is compatible with the validated prefix tree t_p as shown in expression (3).

4. These steps are iterated until a final, perfect parse tree is produced by the system and validated by the user.

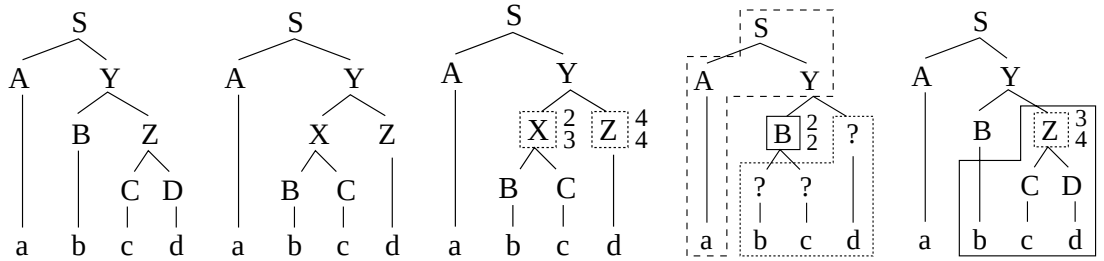
It is worth noting that within this protocol, constituents can be automatically deleted or inserted at the end of any subtree in the syntactic structure by adequately modifying the span of the left-neighbouring constituent.

The IPP interaction process is similar to the ones already established in HTR and SMT. In these fields, the user reads the output sentence from left to right. When the user finds and corrects an erroneous word, he is implicitly validating the prefix sentence up to that word. The remaining suffix sentence is recalculated by the system taking into account the validated prefix sentence.

Fig. 1 shows an example that intends to clarify the Interactive Predictive process. First, the system provides a tentative parse tree (Fig. 1.b). Then the user, which has the correct reference tree (Fig. 1.a) in mind, notices that it has two wrong constituents (c^X_{23} and c^Z_{44}) (Fig. 1.c), and chooses to replace c^X_{23} by c^B_{22} (Fig. 1.d). Here, c^B_{22} corresponds to c'^A_{ij} of expression (3). As the user does this correction, the system automatically validates the prefix (dashed line in Fig. 1.d, $t_p(c'^A_{ij})$ of expression (2)). The system also invalidates the subtrees outside the prefix (dotted line line in Fig. 1.d). Finally, the system automatically predicts a new subtree (Fig. 1.e). Notice how c^Z_{34} changes its span and c^D_{44} is introduced which provides the correct reference parse.

For further exemplification, Sánchez-Sáez et al. (2010) demonstrate an IPP based annotation tool that can be accessed at `http://cat.iti.upv.es/ipp/`.

Within the IPP scenario, the user has to manually review all the system output and correct or validate it, which is still a considerable amount of effort. CMs can ease this work by helping to spot the erroneous constituents.



(a) Reference tree (b) Iteration 0: *Proposed output tree 1* (c) Iteration 0: *Erroneous constituents* (d) Iteration 1: *User corrected constituent* (e) Iteration 1: *Proposed output tree 2*

Figure 1: Synthetic example of user interaction with the IPP system.

3 Confidence Measures

Probabilistic calculation of Confidence Measures (Sánchez-Sáez et al., 2009b) for all tree constituents can be introduced within the IPP process.

The CM of each constituent is its posterior probability, which can be considered as a measure of the degree to which the constituent is believed to be correct for a given input sentence $\boldsymbol{x}$. This is formulated as follows

$$p_G(c_{ij}^A|\boldsymbol{x}) = \frac{p_G(c_{ij}^A, \boldsymbol{x})}{p_G(\boldsymbol{x})} = \frac{\sum_{t' \in \mathcal{T};\, c_{ij}'^A \in t'} \delta(c_{ij}^A, c_{ij}'^A)\; p_G(t'|\boldsymbol{x})}{p_G(\boldsymbol{x})} \tag{4}$$

with $\delta()$ being the Kronecker delta function. Numerator in expression (4) stands for the probability of all parse trees for $\boldsymbol{x}$ that contain the constituent c_{ij}^A (see Fig. 2).

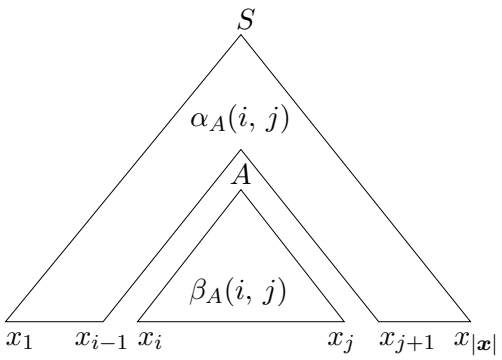


Figure 2: The product of the inside and outside probabilities for each constituent comprises the upper part of expression (5)

The posterior probability is computed with the inside β and outside α probabilities (Baker, 1979)

$$\mathcal{C}(t_{ij}^A) = p_G(c_{ij}^A|\boldsymbol{x}) = \frac{p_G(c_{ij}^A, \boldsymbol{x})}{p_G(\boldsymbol{x})} = \frac{\beta_A(i,\, j)\; \alpha_A(i,\, j)}{\beta_S(1,\, |\boldsymbol{x}|)}\ . \tag{5}$$

It should be clear that the calculation of confidence measures reviewed here is generalizable for any problem that employs PCFGs, and not just NLP tasks. In the experiments presented in the following section we show that CMs are increasingly discriminant when used within the IPP framework to detect erroneous constituents.

4 Experiments

Evaluation of the quality of CMs within the IPP framework is done in a completely automatic fashion by simulating user interaction. Section 4.1 introduces the evaluation protocol and metrics measuring CM quality (i.e., their ability to detect incorrect constituents). The experimentation framework and the results are discussed in section 4.2.

4.1 Evaluation Methods

4.1.1 IPP Evaluation

A good measure of the performance of an Interactive Predictive System is the amount of effort saved by the users of such a system. It is subjective and expensive to test an IPS with real users, so these systems are usually evaluated using automatically calculated metrics that assess the amount of effort saved by the user.

As already mentioned, the objective of an IPP based system is to be employed by annotators to construct correct syntactic trees with less effort. Evaluation of an IPP system was previously done by comparing the IPP usage effort (the number of corrections using the IPP system) against the estimated effort required to manually post-edit the trees after obtaining them with a traditional automatic parsing system (the amount of incorrect constituents) (Sánchez-Sáez et al., 2009a).

In the case of IPP, the gold reference trees are used to simulate system interaction by a human corrector and provide a comparable benchmark. This automatic evaluation protocol is similar to the one presented in section 2.1:

1. The IPP system proposes a full parse tree t for the input sentence.

2. The user simulation subsystem finds the first incorrect constituent by exploring the tree in the order defined by the prefix tree definition (preorder) and comparing it with the *reference*. When the first erroneous constituent is found, it is amended by being replaced in the output tree by the correct one, operation which implicitly validates the prefix tree t_p.

3. The IPP system produces the most probable tree that is compatible with the validated prefix tree t_p.

4. These steps are iterated until a final, perfect parse tree is produced by the IPP system and validated against the *reference* by the user simulation subsystem.

In this work, metrics assessing the quality of CM are introduced within this automatic protocol. We calculate and report them after each of the iterations in the IPP process.

4.1.2 Confidence Measure Evaluation Metrics

The CM of each tree constituent, computed as shown in expression (4) can be seen as its probability of being correct. Once all CM are calculated, a confidence threshold $\tau \in [0, 1]$ can be chosen. Constituents are then marked using τ: the ones with a confidence above this threshold are marked as correct, and the rest as incorrect. Comparing the confidence marks in the output tree with the reference, we obtain the *false rejection* $N_f(\tau) \in [0, N_c]$ (number of correct constituents in the output tree wrongly marked as incorrect by their CM) and the *true rejection* $N_t(\tau) \in [0, N_i]$ (number of incorrect constituents in the output tree that are indeed detected as incorrect by their confidence).

The amount of correct and incorrect constituents in each tree is N_c and N_i respectively. In the ideal case of perfectly error discriminant CM, using the best threshold would yield $N_f(\tau) = 0$ and $N_t(\tau) = N_i$.

A evaluation metric that assess the ability of CMs in telling apart correct constituents from incorrect ones is the *Confidence Error Rate (CER)*:

$$CER(\tau) = \frac{N_f(\tau) + (N_i - N_t(\tau))}{N_c + N_i}. \quad (6)$$

The CER is the number of errors incurred by the CMs divided by the total number of constituents.

The CER can be compared with the Absolute Constituent Error Rate (ACER), which is the CER obtained assuming that all constituents are marked as correct (the only possible assumption when CM are not available):

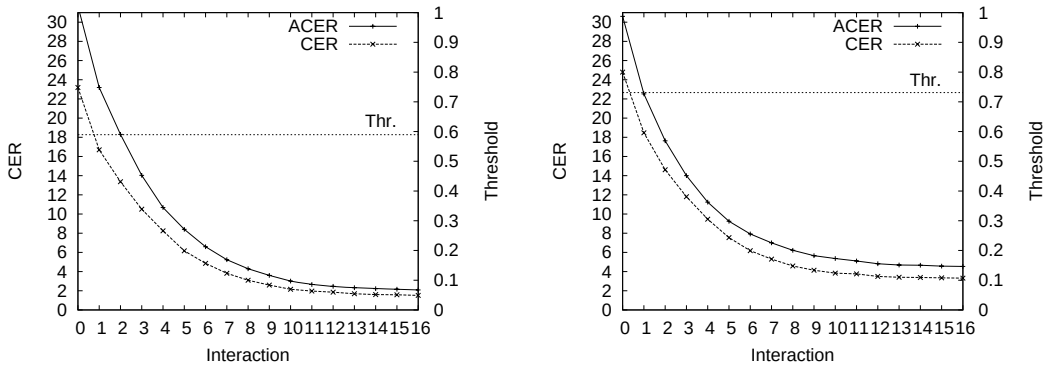
$$ACER = CER(0) = \frac{N_i}{N_c + N_i}. \quad (7)$$

4.2 Experimental Framework

Our experiments were carried out over the Wall Street Journal Penn Treebank (PTB) manually annotated corpus. Three sets were defined over the PTB: train (sections 2 to 21), test (section 23), and development (the first 346 sentences of section 24). Before carrying out experimentation, the *NoEmpties* transformation was applied to all sets (Klein and Manning, 2001).

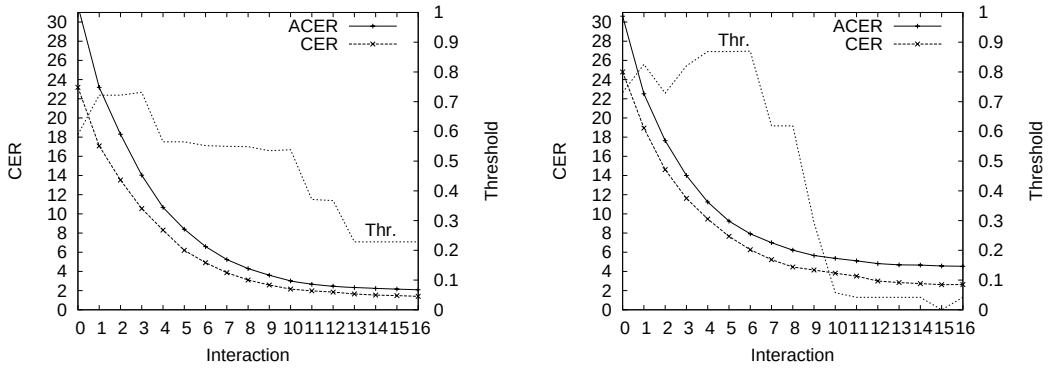
We implemented the CYK-Viterbi parsing algorithm as the parse engine within the IPP framework. This algorithm uses grammars in the Chomsky Normal Form (CNF) so we employed the open source Natural Language Toolkit[2] (NLTK) to obtain several right-factored binary

[2] `http://www.nltk.org/`



(a) PCFG: h=0,v=1 (b) PCFG: h=0,v=2

Figure 3: CER results over IPP system interaction. Threshold fixed at before the interactive process.



(a) PCFG: h=0,v=1 (b) PCFG: h=0,v=2

Figure 4: CER results over IPP system interaction. Threshold optimized for each step of the interactive process.

grammars with different markovization parameters from the training set (Klein and Manning, 2003).

The purpose of our experimentation is to determine if CMs can successfully discriminate erroneous constituents from correct ones within an IPP process, that is, if they help the user to find errors in a hastier manner. For this we need to assess if there exists discriminant information in the CMs corresponding to the constituents of the unvalidated part of the successive IPP-proposed trees.

With this objective in mind, we introduced a CM calculation step after each user interaction within the IPP process. CMs for all constituents in each tree were obtained as described in section 3. After each simulated interaction, we also calculated the ACER and CER over all the syntactic constituents of the whole test set.

Each IPP user interaction yields a parse tree which can be seen as the concatenation of two parts: the validated prefix tree (which is known to be correct because the user, or the user simulation subsystem in this case, has already reviewed it) and a new suffix tree which is calculated by the IPP system based on the validated prefix, as shown in section 2.

The fact that the validated prefix is already known to be correct is taken into account by the CM calculation process, and the confidence of the constituents in the prefix tree is automatically set to their maximum score, equal to 1. This fact causes that the CMs become more discriminant after each interaction, because a larger part of the

tree (the prefix) has a completely correct confidence. The key point here is to measure if this increasingly reduced CER (CM error rate) maintains its advantage over the also increasingly reduced ACER (absolute constituent error rate without taking CMs into account) which would mean that the CMs retain their discriminant power and can be useful as an aid for a human annotator using an IPP system.

Two batches of experiments were performed and, in each of them, two different markovizations of the vanilla PCFG were tested as the parsing model.

In the first battery of experiments, the confidence threshold τ was optimized over the development set before starting the IPP process, remaining the same during the user interaction. The results can be seen in Fig. 3, which shows the obtained baseline ACER and the CER (the confidence assessing metric) for the test set after each user interaction. We see how CMs retain all of their error detection capabilities during the IPP process: in the h0v1 PCFG they are able to discern about 25% of incorrect constituents at most stages of the IPP process, with a slight bump up to 27% after about 7 user interactions; for the h0v2 PCFG they are able to detect about 18% of incorrect constituents at the first interactions, but go up to detect 27% of errors after about 7 or more interactions.

In the second experimental setup, a different threshold for each interaction step was calculated by performing the IPP user simulation process over the development set and optimizing the threshold value. The results can be seen in Fig. 4. We observe improvements in the discriminant ability of confidence values after 8 user interactions, with them being capable to detect more errors towards the end of each IPP session: about 34% of errors for h0v1, and 49% of them for h0v2.

The calculated thresholds have also been plotted in the aforementioned figures. For the per-interaction threshold experimentation, we can see how the threshold gets fine-tuned as the IPP process advances. The lower threshold values for the last interactions were expected due to the fact that more constituents have been validated and have the maximum confidence. This method for precalculating one specific threshold for each of the iterations could be useful when incorporating CM to a real IPP based annotator.

5 Conclusions and Future Work

We have proved that using Confidence Measures can be used to discriminate incorrect constituents from correct ones over an Interactive Predictive Parsing process. We have show two methods for calculating the threshold used to mark constituents as correct/incorrect, showing the advantage of precalculating a specific threshold for each of the interaction steps.

Immediate future work involves implementing CMs as a visual aid in a real IPP system like the one presented in (Sánchez-Sáez et al., 2010). Through he use of CMs, all constituents in the successive trees could be color-coded according to their correctness confidence, so the user could focus and make corrections faster.

Future research paths can deal with applying CMs to improve the output of completely automatic parsers, for example, using them as a component of an n-best re-ranking system.

Additionally, the IPP framework is also suitable for studying and applying training algorithms within the Active Learning and Adaptative/Online Parsing paradigms. This kind of systems could improve their models at operating time, by incorporating new ground truth data as it is provided by the user.

References

Baker, JK. 1979. Trainable grammars for speech recognition. *Journal of the Acoustical Society of America*, 65:132.

Barrachina, S., O. Bender, F. Casacuberta, J. Civera, E. Cubel, S. Khadivi, A. Lagarda, H. Ney, J. Tomás, E. Vidal, and J.M. Vilar. 2009. Statistical approaches to computer-assisted translation. *Computational Linguistics*, 35(1):3–28.

Benedí, J.M., J.A. Sánchez, and A. Sanchís. 2007. Confidence measures for stochastic parsing. In *Proc. of RANLP*, pages 58–63, Borovets, Bulgaria, 27-29 September.

Carter, D. 1997. The TreeBanker. A tool for supervised training of parsed corpora. In *Proc. of ENVGRAM Workshop*, pages 9–15.

Collins, M. 2003. Head-driven statistical models for natural language parsing. *Computational Linguistics*, 29(4):589–637.

de la Clergerie, E.V., O. Hamon, D. Mostefa, C. Ayache, P. Paroubek, and A. Vilnat. 2008. Passage: from French parser evaluation to large sized treebank. *Proc. of LREC*, 100:2.

Hiroshi, I., N. Masaki, H. Taiichi, T. Takenobu, and T. Hozumi. 2005. eBonsai: An integrated environment for annotating treebanks. In *Proc. of IJCNLP*, pages 108–113.

Huang, L. 2008. Forest reranking: Discriminative parsing with non-local features. In *Proc. of ACL*.

Klein, D. and C.D. Manning. 2001. Parsing with treebank grammars: Empirical bounds, theoretical models, and the structure of the Penn treebank. In *Proc. of ACL*, pages 338–345, Morristown, USA. ACL.

Klein, D. and C.D. Manning. 2003. Accurate unlexicalized parsing. In *Proc. of ACL*, volume 1, pages 423–430, Morristown, USA. ACL.

Marcus, M.P., B. Santorini, and M.A. Marcinkiewicz. 1994. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

McClosky, D., E. Charniak, and M. Johnson. 2006. Effective self-training for parsing. In *Proc. of NAACL-HLT*, pages 152–159.

Oepen, S., D. Flickinger, K. Toutanova, and C.D. Manning. 2004. LinGO Redwoods. *Research on Language & Computation*, 2(4):575–596.

Petrov, S. 2010. Products of Random Latent Variable Grammars. *Proc. of NAACL-HLT*.

Sánchez-Sáez, R., J.A. Sánchez, and J.M. Benedí. 2009a. Interactive predictive parsing. In *Proc. of IWPT'09*, pages 222–225, Paris, France, October. ACL.

Sánchez-Sáez, R., J.A. Sánchez, and J.M. Benedí. 2009b. Statistical confidence measures for probabilistic parsing. In *Proc. of RANLP*, pages 388–392, Borovets, Bulgaria, September.

Sánchez-Sáez, R., L.A. Leiva, J.A. Sánchez, and J.M. Benedí. 2010. Interactive predictive parsing using a web-based architecture. In *Proc. of NAACL-HLT*, Los Angeles, United States of America, June.

Serrano, N., A. Sanchis, and A. Juan. 2010. Balancing error and supervision effort in interactive-predictive handwriting recognition. In *Proc. of IUI*, pages 373–376. ACM.

Tarazón, L., D. Pérez, N. Serrano, V. Alabau, O. Ramos Terrades, A. Sanchis, and A. Juan. 2009. Confidence Measures for Error Correction in Interactive Transcription of Handwritten Text. In *Proc. of ICIAP*, pages 567–574, Vietri sul Mare, Italy, September. LNCS.

Toselli, A.H., V. Romero, and E. Vidal. 2008. Computer assisted transcription of text images and multimodal interaction. In *Proc. MLMI*, volume 5237, pages 296–308. Springer.

Ueffing, N. and H. Ney. 2007. Word-level confidence estimation for machine translation. *Computational Linguistics*, 33(1):9–40.

Vidal, E., F. Casacuberta, L. Rodríguez, J. Civera, and C. Martínez. 2006. Computer-assisted translation using speech recognition. *IEEE TASLP*, 14(3):941–951.

Wessel, F., R. Schluter, K. Macherey, and H. Ney. 2001. Confidence measures for large vocabulary continuous speech recognition. *IEEE TSAP*, 9(3):288–298.

Yamamoto, R., S. Sako, T. Nishimoto, and S. Sagayama. 2006. On-line recognition of handwritten mathematical expressions based on stroke-based stochastic context-free grammar. In *Proc of ICFHR*, pages 249–254.

Learning Web Query Patterns for Imitating Wikipedia Articles

Shohei Tanaka[†]
tanaka@mi.ci.i.u-tokyo.ac.jp

Naokaki Okazaki[‡]
okazaki@is.s.u-tokyo.ac.jp

Mitsuru Ishizuka[†]
ishizuka@i.u-tokyo.ac.jp

[†]Graduate School of Information Science and Technology
University of Tokyo

[‡]Interfaculty Initiative in Information Studies
University of Tokyo

Abstract

This paper presents a novel method for acquiring a set of query patterns to retrieve documents containing important information about an entity. Given an existing Wikipedia category that contains the target entity, we extract and select a small set of query patterns by presuming that formulating search queries with these patterns optimizes the overall precision and coverage of the returned Web information. We model this optimization problem as a weighted maximum satisfiability (weighted Max-SAT) problem. The experimental results demonstrate that the proposed method outperforms other methods based on statistical measures such as frequency and point-wise mutual information (PMI), which are widely used in relation extraction.

1 Introduction

Wikipedia[1] is useful for obtaining comprehensive information of entities and concepts. However, even with 3.3 million English articles, Wikipedia does not necessarily include articles about an entity and concept of interest to a user. The ultimate goal of this study is to generate articles about an entity of a specified category from the Web by using Wikipedia articles in the same entity category as *exemplars*.

This study follows previous work of the other authors on query-biased/focused summarization (Tombros and Sanderson, 1998; Berger and Mittal, 2000) for modeling the target article generation process. In that model, when a user inputs an entity of interest, Web pages are retrieved that describe the entity by issuing queries to an information retrieval system. Using the retrieved pages as an information source, an article (summary) can be produced specialized for the target entity. From the application point of view, the article should include the concepts that best describe the target entity. In addition, articles concerning the entities of a category should cover concepts that are typical of the category. For example, an article about an actor is expected to mention his nationality, date of birth, movie credits, awards, etc.

A great number of researchers have addressed the problem of query-focused summarization (Carbonell and Goldstein, 1998; White et al., 2003; Dang, 2005; Daumé and Marcu, 2006; Varadarajan and Hristidis, 2006; Fuentes et al., 2007; Gupta et al., 2007; Wang et al., 2007; Kanungo et al., 2009). However, these studies assume that a document collection is provided for the summarization systems. In other words, collecting source documents that include important concepts for the target entity is not in the scope of these studies. For example, queries such as "(actor) was born in," "(actor) born on," "(actor) plays," and "(actor) won" may be more suitable than the simple query "(actor)" for obtaining concepts concerning the actor.

Source documents can be collected by a similar idea in relation extraction, which extracts entities having specific relations with the target entity (Hearst, 1992; Brin, 1999; Agichtein and Gravano, 2000; Turney, 2001; Pantel and Pennac-

[1] http://en.wikipedia.org/

Coling 2008: Poster Volume, pages 1229–1237,
Beijing, August 2010

chiotti, 2006; Blohm et al., 2007). These studies typically use statistical measures, such as frequency and point-wise mutual information (PMI), to assess the scores of the query patterns. However, these studies cannot eliminate the redundancy of concepts retrieved by a query set because they are designed to extract entities for each relation independently. For example, the query "(actor) born on" would not be necessary if the query "(actor) was born in" could gather documents referring to both the actor's nationality and date of birth.

In this paper, we propose a novel method for acquiring a set of high-quality query patterns that can gather source documents referring to important concepts about a specified entity. Given a category in which the entity is expected to be included, we use existing Wikipedia articles in this category to extract query patterns so that, when used together with the entity, they can retrieve important concepts related to the entity. We then select a small subset of query patterns that maximize the coverage and precision of the query result by modeling the query selection task as a weighted maximum satisfiability (weighted Max-SAT) problem.

2 Proposed method

First, let us define the terminology used in this paper. An *entity* is a topic for which we need to obtain an article (summary). Note that this definition is different from that used in other studies (e.g., named entity recognition). A *concept* is a noun phrase that has a specific relation to an entity. A *query pattern* is a lexical pattern that contains a slot filled by an entity. Used with an entity, a query pattern instantiates a query that collects related concepts. For example, "X was born in" is a query pattern in which X is a slot. When replacing X with an entity (e.g., "Dustin Hoffman"), the query pattern instantiates a query that may return the birthplace.

The goal of this study is, for a given entity category (e.g., *American actor*), to acquire a set of query patterns (*template*) for collecting related concepts from the Web. We learn the template by using Wikipedia articles of the category as supervision data. The method consists of three steps.

1. **Triplet extraction** identifies, for each Wikipedia article, entity mentions, concepts, and phrases that form a bridge between the entity mentions and concepts. In the context of learning query patterns from Wikipedia, we assume that a Wikipedia article is written for an entity. By identifying entity mentions and concepts in the article, we obtain bridging phrases between entity mentions and concepts as candidates for query patterns.

2. **Pattern assessment** verifies whether each candidate query pattern can actually retrieve concepts from the Web. This step issues queries of the form "(entity) (pattern)" to an information retrieval system, analyzes the retrieved Web pages, and examines whether each concept is found in the same sentence as the query expressions.

3. **Pattern selection** obtains a template by choosing a small subset of patterns such that the retrieved Web pages contain as many kinds of concepts as possible. We also eliminate query patterns that can retrieve descriptions other than concepts. We formalize this step as a weighted Max-SAT problem.

2.1 Triplet extraction

We first analyze Wikipedia articles to extract triplets of entities, query patterns, and concepts. Because a Wikipedia article usually describes a single entity, we identify the entity from the title of the article. We then search for occurrences of the entity in the body of the article. However, we might need to resolve coreference expressions because the entity might be described by a number of surface variations. For example, the Wikipedia article titled "Dustin Hoffman" might refer to the entity using "he" and "Hoffman" as well as "Dustin Hoffman"; the entity "Microsoft Corporation" might be described by "Microsoft" and "the company" in the article.

In general, coreference resolution is a non-trivial NLP task. Fortunately, Wikipedia articles are written for target entities. Therefore, we replace the occurrences of the following expressions in the body with the entity name:

Hoffman *was born in* [Los Angeles], [California], the second and youngest son of Lillian and Harry Hoffman, a [Russian]-born father who worked as a prop supervisor/set decorator at [Columbia Pictures] before becoming a furniture salesman. Hoffman *is from a* [Jewish] family, although he did not have a religious upbringing. He *graduated from* [Los Angeles High School] in 1955. He *enrolled at* [Santa Monica College] with the intention of studying medicine but left after a year to join the [Pasadena Playhouse].

Figure 1: A snippet of a Wikipedia article about "Dustin Hoffman."

1. Any token (split by spaces) that appears in the title of the article.

2. The phrase that appears the most frequently with the four anaphoric expressions "he," "she," "they," and "the *noun*."

The first rule deals with anaphoric expression caused by an ellipsis, e.g., "Dustin Hoffman" is referred to by "Dustin" and "Hoffman." The second rule resolves the coreference expressions caused by pronouns and definite noun phrases.

After detecting the entity mentions, we identify the concepts concerning the entity in the article. In this study, we employ WikiLink texts (anchor texts linked with other Wikipedia articles) that co-occur with the entity mentions in the same sentences. Finally, we identify a candidate of a query pattern as a phrase that satisfies the following conditions:

1. It consists of alphanumeric letters and hyphens only.

2. Its length is no longer than 6 tokens.

3. It appears between an entity mention and a concept in a sentence.

Figure 1 shows a snippet of the Wikipedia article about "Dustin Hoffman." The underlined expressions are identified as entity mentions; the text in square brackets represents a WikiLink text. Because all WikiLink texts appear in sentences with the entity mentions, we identify all expressions with square brackets as concepts. Italic texts are candidates of the query patterns.

Finally, we extract triplets of the form $\langle E_k, P_i, C_j \rangle$ from the Wikipedia article, where E_k ($k \in \{1,...,L\}$) denotes the entity, P_i ($i \in \{1,...,M\}$) denotes a query pattern, and C_j ($j \in \{1,...,N\}$) denotes a concept. For each concept C_j found in the Wikipedia article, we build a triplet by setting E_k as the entity of the article and P_j as the query pattern that precedes the concept C_j. Repeating this process for L Wikipedia articles in the same category, we obtain triplets with M query patterns and N concepts.

Table 1: Triplets extracted from Figure 1.

Entity	Query pattern	concept
Dustin Hoffman	was born in	Los Angeles
Dustin Hoffman	was born in	California
Dustin Hoffman	was born in	Russian
Dustin Hoffman	was born in	Columbia Pictures
Dustin Hoffman	is from a	Jewish
Dustin Hoffman	graduated from	Los Angeles High School
Dustin Hoffman	enrolled at	Santa Monica College
Dustin Hoffman	enrolled at	Pasadena Playhouse

Table 1 shows the eight triplets obtained from Figure 1. Here, it might not be clear whether the indefinite article *a* is necessary in the pattern *is from a*. Although we do not address this issue directly in this paper, we determine the popularity and usefulness of the pattern by analyzing Wikipedia articles in the same category. Similarly, some concepts (e.g., *Russian*) are not so important for the entity. It may be better to filter out the concept, but we expect that errors in concept identification are negligible when selecting the query patterns.

2.2 Pattern assessment

In this step, we verify whether each pattern P_i can actually retrieve concepts of the entities from the Web. More specifically, for every combination of an entity mention E_k and a pattern P_i, we issue a query "E_k P_i" (e.g., *"Dustin Hoffman graduated from"*) to Yahoo! Search BOSS[2]. We download the top 10 Web pages retrieved by each query and examine whether any of the concepts, C_j, appear in the same sentence as the query phrase. To describe the capability of the patterns for retrieving concepts, we introduce an $m \times n$ matrix called R,

$$R_{ij} = \begin{cases} 1 & \text{(pattern } P_i \text{ can retrieve concept } C_j\text{)} \\ 0 & \text{(otherwise)} \end{cases}.$$

[2]http://developer.yahoo.com/search/boss/

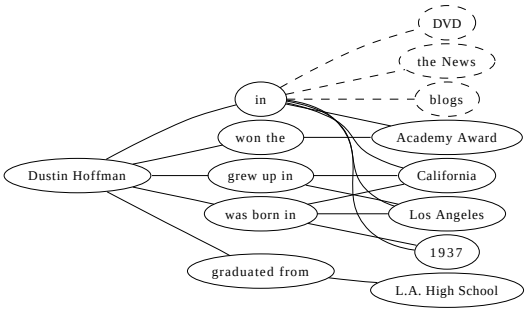


Figure 2: Patterns – collected concepts graph.

Figure 2 illustrates a bipartite graph between the patterns and concepts with R as the biadjacency matrix. Here, nodes with dotted lines are expressions other than concepts but are retrieved by the patterns. For example, the pattern "in" can retrieve four concepts *Academy Award*, *California*, *Los Angeles*, and *1937*, but it also retrieves non-concepts such as *DVD*, *the News*, and *blogs*. In other words, this pattern can retrieve sentences with a number of concepts, but it also gathers unnecessary sentences. Thus, we define the error rate of pattern P_i

$$\varepsilon_i = 1 - \frac{(\text{\# sentences with concepts})}{(\text{\# total sentences retrieved by } P_i)}.$$

2.3 Pattern selection

Based on the pattern assessment in Section 2.2, this step chooses a small set of query patterns as the template. Let $w_1, ..., w_m$ denote m Boolean (0–1 integer) variables, each of which (w_i) indicates whether the corresponding query pattern P_i is selected (1) or unselected (0). Choosing a subset of query patterns is equivalent to assigning Boolean values to the variables $w_1, ..., w_m$. The number of selected patterns is $\sum_{i=1}^{m} w_i$.

Given an assignment of variables $w_1, ..., w_m$ for the query patterns, we can examine whether the concept C_j is retrieved from the patterns by using the logical sum,

$$c_j = w_1 R_{1j} \vee w_2 R_{2j} \vee ... \vee w_m R_{mj} = \bigvee_{i=1}^{m} w_i R_{ij}.$$

Here, c_j is a Boolean (0–1) variable indicating that concept C_j is retrieved (1) or not retrieved (0) by the template. In Figure 2, if either the "in" or "was born in" pattern is selected, we can retrieve the concept "*1937*" from the Web search.

To choose a set of query patterns, we maximize the number of concept coverages $\sum_{j=1}^{n} c_j$ as well as minimize the number of patterns selected $\sum_{i=1}^{m} w_i$ and the total of the error rates of the selected patterns $\sum_{i=1}^{m} \varepsilon_i w_i$. This is achieved by solving the following problem.

Problem 1.

$$\begin{aligned} &\textit{Maximize} \sum_{j=1}^{n} c_j - \alpha \sum_{i=1}^{m} w_i - \beta \sum_{i=1}^{m} \varepsilon_i w_i, \\ &\textit{subject to: } c_1 = \bigvee_{i=1}^{m} w_i R_{i1} \\ &\qquad \cdots \\ &\qquad c_n = \bigvee_{i=1}^{m} w_i R_{in}, \\ &\qquad w_i \in \{0,1\}. \end{aligned}$$

Here, α and β are the parameters for controlling the preference of a smaller number of patterns (α) and the preference of accurate patterns (β).

To solve this problem, we rewrite it as a weighted maximize satisfiability (weighted Max-SAT) problem.

Problem 2.

$$\begin{aligned} &\textit{Maximize} \sum_{k=1}^{n+m} \lambda_k x_k \\ &\textit{Subject to: } x_1 = \bigvee_{i=1}^{m} w_i R_{i1} \quad (\lambda_1 = 1) \\ &\qquad \cdots \quad (\ldots) \\ &\qquad x_n = \bigvee_{i=1}^{m} w_i R_{in} \quad (\lambda_n = 1) \\ &\qquad x_{n+1} = \neg w_1 \quad (\lambda_{n+1} = \alpha + \beta\varepsilon_1) \\ &\qquad \cdots \quad (\ldots) \\ &\qquad x_{n+m} = \neg w_m \quad (\lambda_{n+m} = \alpha + \beta\varepsilon_m) \\ &\qquad w_i \in \{0,1\} \end{aligned}$$

Instead of subtracting the penalty terms from the objective value, we give bonus weights ($\alpha + \beta\varepsilon_i$) if the pattern P_i is not selected. This is achieved by introducing additional clauses x_{n+1}, ..., x_{n+m} that are satisfied by $\neg w_1$, ..., $\neg w_m$, respectively. Therefore, the optimization process tries to find a compromise between selecting patterns (clauses x_1, ..., x_n) and rejecting patterns (clauses x_{n+1}, ..., x_{n+m}). Although the complexity of the weighted Max-SAT problem is NP-hard, we use MiniMaxSAT (Heras et al., 2008) to solve the problem.

3 Evaluation

To verify the performance of the proposed method, we compare the precision, coverage, and F'-score of the information retrieval process by using the template obtained by the proposed method with that by three other baseline methods.

3.1 Experimental Settings

3.1.1 Data

We use articles of five categories in Wikipedia as the data for evaluation: *American actors*, *Genetic disorders*, *American tennis players*, *Software companies*, and *Operas*. Among these categories, the first two (*American actors*, *Genetic disorders*) have been commonly used as evaluation data in previous research on text summarization (Sauper and Barzilay, 2009). The other three (*American tennis players*, *Software companies*, *Operas*) are categories about three distinct topics (*sport*, *business*, *entertainment*).Table 2 shows information about these categories.

We divide the article set of a given category into six subsets. We use one subset as the *development set* for tuning the parameters α and β in the proposed method. The remaining five subsets are the *training set* and the *test set*, which are used for the 5-fold cross-validation. We create the template by using the training set and evaluate it with the test set. For evaluation of the baseline methods, we use only the *training set* and the *test set*.

3.1.2 Baselines

Random Selection

The baseline "*Random Selection*" randomly selects 10 query patterns from the candidate query patterns as the template for the category.

Frequency

In this baseline method, we sort the query patterns for each category in the order of frequency of occurrences in the category. We then select the top 10 frequent query patterns as the template for the category.

PMI-Web

The baseline *PMI-Web* chooses the query patterns that are the most "reliable." Following *KnowItNow* (Cafarella et al., 2005) and *Espresso* (Pantel and Pennacchiotti, 2006), the "reliability" of a pattern is defined by using the strength of the association of the pattern with the entities and concepts co-occurring with the pattern. In *KnowItNow* and *Espresso*, PMI (point-wise mutual information) is used to measure the strength of this association. PMI is estimated with the Web search hit counts as follows:

$$pmi(E_k, P_i, C_j) \approx \frac{\mathrm{hit}(E_k, P_i, C_j)}{\mathrm{hit}(E_k, P_i) \cdot \mathrm{hit}(C_j)},$$

where $\mathrm{hit}(E_k P_i)$, $\mathrm{hit}(C_j)$ are the Web search hit counts for the query "E_k, P_i," "C_j"(E_k, P_i, C_j is *entity, pattern, concept*), and $\mathrm{hit}(E_k, P_i, C_j)$ is the hit count for the query "E_k P_i" and "C_j." The reliability score of the query pattern is defined as the following formula:

$$Score(P_i) = \frac{1}{|S|} \sum_{(E_k, C_j) \in S} pmi(E_k, P_i, C_j),$$

where S is the set of pairs of *entity* E_k and *concept* C_j co-occurring with the *pattern* P_i in a sentence. The method *PMI-Web* chooses the top 10 patterns that have the highest reliability scores.

3.1.3 Experiments

We use each method to generate a template and retrieve information of the entities by using the

Table 2: The five categories used for evaluation.

Category	#Articles	#Patterns	#Concepts
American actors	1864	2951	10495
American tennis players	444	1039	2826
Software companies	1890	1992	5087
Genetic disorders	657	1087	2400
Operas	1425	2125	6365

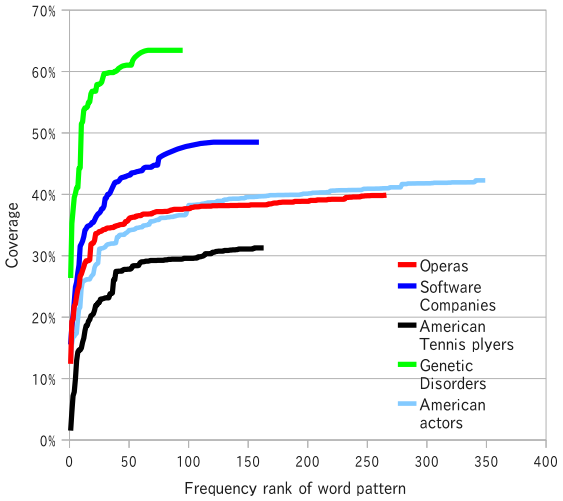


Figure 3: Relation between coverage and query pattern frequency.

query patterns in the template. We remove the query patterns occurring only once in each category from the candidate patterns because these patterns may be too entity-specific or noisy.

Figure 3 shows the coverage of the concept retrieval process when we use the top N frequently appearing patterns in the candidate pattern set. We observe that the coverage does not reach 100% even if we use all the query patterns. This is because some concepts cannot be retrieved by any query pattern. Moreover, we consider a Wikilink as a concept, even though some Wikilink texts do not actually represent a concept.

We use query patterns that occur no less than 3 times (for American actors, American tennis players, Software companies, and Operas) or twice (for Genetic disorders) in the corresponding Wikipedia articles so that the query patterns reach 95% of the upper bound of the coverage. This small subset comprises the final candidate patterns. For the candidate patterns, we use the proposed method (solving the weighted Max-SAT problem) and the three baselines described above to choose N query patterns.

The precision, coverage and quasi F-score (F'-score) of the information retrieval process by each template are defined as follows:

$$\text{precision} = \frac{\text{freq}(\text{E}_\text{k}, \text{P}_\text{i}, \text{C}_\text{j})}{\text{freq}(\text{E}_\text{k}, \text{P}_\text{i})}, \text{coverage} = \frac{\text{C}_\text{collected}}{\text{C}_\text{total}},$$

$$F' = \frac{2 \cdot \text{precision} \cdot \text{coverage}}{\text{precision} + \text{coverage}},$$

where $\text{freq}(E_k, P_i)$ is the frequency of the phrase "E_k P_i" in the retrieved documents, $\text{freq}(E_k, P_i, C_j)$ is the frequency of co-occurrence of the phrase "E_k P_i" and "C_j" in the sentences. C_total is the total number of distinct concepts in the data set, and $\text{C}_\text{collected}$ is the number of distinct concepts which can be collected by the template.

3.2 Result

Table 5 shows the average of the precision, coverage and F' scores of the five categories when we choose 10 query patterns (N=10). The proposed method obtains the highest score of all methods. Moreover, the proposed method outperforms the baselines not only for the average of all categories, but also for each category. This result indicates that the proposed method is able to choose query patterns that precisely and comprehensively collect the target concepts.

Table 5: Performance of the templates produced by the proposed method and the three baselines (N=10).

Method	Precision	Coverage	F' score
Random	16.56	11.40	13.19
Frequency	21.43	29.29	24.40
PMI-Web	22.55	22.08	21.42
Proposed	**27.34**	**30.77**	**27.95**

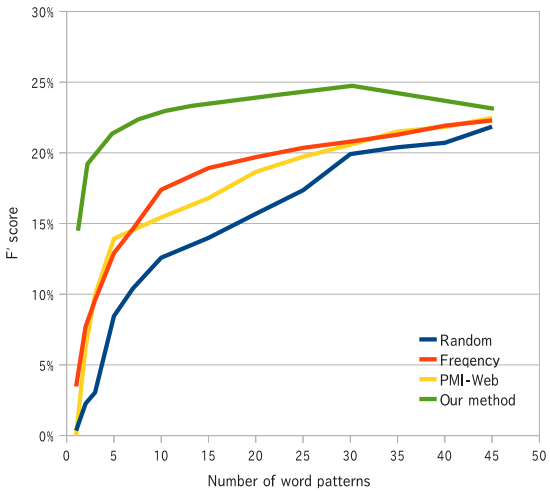


Figure 4: Number of query patterns (N) in template and F' score in *American tennis players*.

Table 3 shows some example templates produced by the proposed method. In this table, the number in the parentheses next to a pattern is the frequency rank of the pattern. We observe that the proposed method generates templates with two types of query patterns: *generic patterns* and *specific patterns*. Generic patterns such as "is a" and "is an" are patterns that can appear in every category. These patterns cover various kinds of concepts (high coverage), but may retrieve sentences that do not describe any concept (low precision). Specific patterns, such as "has a star on the" and "was nominated for a," can retrieve concepts that have specific relations with the entity. Therefore, queries with specific patterns retrieve a small number of concepts with high precision. The proposed method chooses query patterns in both of these types to achieve both high precision and high coverage. Therefore, it is able to retrieve

Table 3: Templates generated by the proposed method (N=10): (n) is the frequency rank.

Category	Template
American actors	"is an"(1), "was an"(2), "was a"(7), "graduated from"(9), "died of"(18), "has a star on the"(24), "was nominated for a"(28), "was married to"(47), "was born on"(56), "has appeared in"(92)
American tennis players	"defeated"(3), "beat"(5), "is a former"(12), "is an"(16), "graduated from"(17), "reached the"(18), "played"(24), "of"(30), "was"(38), "won"
Software companies	"is a"(1), "acquired"(3), "is"(9), "is headquartered in"(11), "was founded by"(15), "has offices in"(22), "was"(29), "include"(36), "introduced"(41), "is an international"(41)
Genetic disorders	"is a"(1), "is an"(2), "has an autosomal recessive pattern of"(3), "has an autosomal dominant pattern of"(9), "is named after"(11), "is a form of"(13), "is caused by"(16), "include"(18), "appears to be inherited in an"(41), "is considered an"(41)
Operas	"is an"(1), "is a"(2), "is an opera by"(17), "was"(18), "is a comic"(20), "premiered at the"(25), "was first performed at"(31), "is the second"(38), "opera"(46), "libretto by"(62)

Table 4: Templates generated by different methods for the *Opera* category (N= 10).

Method	Template	Pre.	Cov.	F'
Random	"was an," "of the complete operas of the," "was on," "was commissioned by," "by," "is," "popular," "for the," "New York," "the same name by"	15.79	8.12	10.73
Frequency	"is an," "is a," "the," "by," "of the," "in," "and," "of," "was a," "a"	16.83	27.12	20.77
PMI-Web	"was created by," "is a three act," "premiered at the," "was an," "is an," "is an opera composed by," "is a Hindi language," "premiered on" "was commissioned by," "was first performed at the,"	30.25	18.29	22.80
Proposed	"is an," "is a," "is an opera by," "was," "is a comic," "premiered at the," "was first performed at," "is the second," "opera," "libretto by"	31.18	27.28	29.10

various types of concepts. This implies that the method achieves high coverage even for concepts that cannot be retrieved by generic patterns.

The baseline *Frequency* obtains the second highest F'-score. It achieves high coverage but low precision. This is because this method chooses high-frequency patterns that can appear with every concept. Therefore, it is able to retrieve concepts with high coverage. However, these patterns do not retrieve specific information concerning a concept. Moreover, some high-frequency patterns, such as "the " and "by," lead to sentences that do not describe any concept.

Table 4 shows the patterns generated for the category *Opera* by each method. We can observe that the method *Frequency* chooses very generic patterns, such as "is a," "and," and "the," which are not specific to *Opera*.

In contrast, the method *PMI-Web* achieves high precision but low coverage. This is because this method chooses highly reliable patterns (e.g., "was commissioned by"), which are strongly associated with a specific kind of concept. However, these patterns cannot retrieve a broad range of concepts related to the target entity. This explains why the method cannot achieve high coverage.

Figure 4 shows the F' scores when we vary the number of selected query patterns (N) for the category *American tennis players*. We observe that the templates generated by the proposed method achieve the highest F'-score at every value of N. The maximum F'-score is 24.7, which is achieved when N is 30. Moreover, the proposed method requires only five query patterns to achieve the F'-score of 21.4. Therefore, the proposed method achieves a high F'-score by using only a small number of patterns. This implies that the method achieves high performance in a short query processing time.

4 Related Work

Many studies have addressed the problem of pattern extraction from Wikipedia (or other large corpora). Filatova et al. (2006) presented an approach for automatically extracting important word patterns from a large corpus. They analyzed the BBC corpus to extract word patterns containing verbs that are supposed to be important for a specific domain. Biadsy et al. (2008) described a system for producing biographies for a given target name. They used Wikipedia to learn the document structures of a biography. Ye et al. (2009)

explored a method for generating a series of summaries of various lengths by using information from Wikipedia.

Sauper and Barzilay (2009) proposed an approach for creating a summary of many chunks of text that are related to an entity and retrieved from the Web. They used Wikipedia not only for producing the template, but also for improving the summaries. Although the target of their work is very close to that of our study, the focus of each study is different. They address the method for selecting appropriate sentences for summarization, whereas we consider the method for selecting query patterns that can generate a comprehensive summary of an entity.

Various studies have addressed Web page summarization and query-focused summarization, from search result summarization (Kanungo et al., 2009) to query biased summarization (Wang et al., 2007). Furthermore, Fujii and Ishikawa (2004) presented a method to automatically compile encyclopedic knowledge from the Web.

Similar to relation extraction, the proposed method retrieves information concerning an entity by using query patterns. This is because query patterns for relation extraction are also appropriate in sentence extraction for multi-document summarization (Hachey, 2009). However, the relation extraction task primarily obtains query patterns that retrieve instances of a specific relation. This is different from the goal of this study, which is obtaining a set of patterns that are able to retrieve a large range of topics related to an entity.

5 Conclusion

We present a novel method to acquire a set of query patterns for retrieving documents that contain important information regarding an entity. Especially, we concentrate on the method for selecting query patterns that are able to comprehensively and precisely retrieve important concepts concerning an entity. The experimental results demonstrate that the proposed method outperforms methods based on statistical measures such as frequency and point-wise mutual information (PMI), which are widely used in relation extraction.

Currently, we use the text between an entity and a WikiLink as a candidate for a query pattern. In the future, we plan to use the text between two noun phrases as query patterns to increase the number of candidates for the pattern selection process. Moreover, we intend to build a text summarization application based on the proposed method to confirm that the selected pattern set is able to generate a comprehensive summary for an entity.

References

Agichtein, Eugene and Luis Gravano. 2000. Snowball: Extracting relations from large plain-text collections. In *Proc. of the fifth ACM conference on Digital libraries*, pages 85–94.

Berger, Adam and Vibhu O. Mittal. 2000. Query-relevant summarization using FAQs. In *Proc. of the 38th Annual Meeting on Association for Computational Linguistics*, pages 294–301.

Biadsy, Fadi, Julia Hirschberg, and Elena Filatova. 2008. An unsupervised approach to biography production using Wikipedia. In *Proc. of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies*, pages 807–815.

Blohm, Sebastian, Philipp Cimiano, and Egon Stemle. 2007. Harvesting relations from the Web: quantifiying the impact of filtering functions. In *Proc. of the 22nd national conference on Artificial intelligence*, pages 1316–1321.

Brin, Sergey. 1999. Extracting patterns and relations from the World Wide Web. *Selected papers from the International Workshop on The World Wide Web and Databases*, pages 172–183.

Cafarella, Michael J., Doug Downey, Stephen Soderland, and Oren Etzioni. 2005. KnowItNow: Fast, scalable information extraction from the Web. In *Proc. of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 563–570.

Carbonell, Jaime and Jade Goldstein. 1998. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proc. of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 335–336.

Dang, Hoa Trang. 2005. Overview of DUC 2005. In *Document Understanding Conference (DUC) 2005*.

Daumé, III, Hal and Daniel Marcu. 2006. Bayesian query-focused summarization. In *Proc. of the 21st*

International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics, pages 305–312.

Filatova, Elena, Vasileios Hatzivassiloglou, and Kathleen McKeown. 2006. Automatic creation of domain templates. In *Proc. of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 207–214.

Fuentes, Maria, Enrique Alfonseca, and Horacio Rodríguez. 2007. Support vector machines for query-focused summarization trained and evaluated on pyramid data. In *Proc. of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, pages 57–60.

Fujii, Atsushi and Tetsuya Ishikawa. 2004. Summarizing encyclopedic term descriptions on the Web. In *Proc. of the 20th international conference on Computational Linguistics*, pages 645–651.

Gupta, Surabhi, Ani Nenkova, and Dan Jurafsky. 2007. Measuring importance and query relevance in topic-focused multi-document summarization. In *Proc. of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, pages 193–196.

Hachey, Ben. 2009. Multi-document summarisation using generic relation extraction. In *Proc. of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 420–429.

Hearst, Marti A. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proc. of the 14th conference on Computational linguistics*, pages 539–545.

Heras, Federico, Javier Larrosa, and Albert Oliveras. 2008. MiniMaxSat: An efficient weighted Max-SAT solver. *Journal of Artificial Intelligence Research*, 31:1–32.

Kanungo, Tapas, Nadia Ghamrawi, Ki Yuen Kim, and Lawrence Wai. 2009. Web search result summarization: Title selection algorithms and user satisfaction. In *Proceeding of the 18th ACM conference on Information and knowledge management*, pages 1581–1584.

Pantel, Patrick and Marco Pennacchiotti. 2006. Espresso: Leveraging generic patterns for automatically harvesting semantic relations. In *Proc. of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 113–120.

Sauper, Christina and Regina Barzilay. 2009. Automatically generating Wikipedia articles: a structure-aware approach. In *Proc. of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 208–216.

Tombros, Anastasios and Mark Sanderson. 1998. Advantages of query biased summaries in information retrieval. In *Proc. of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 2–10.

Turney, Peter D. 2001. Mining the Web for synonyms: PMI-IR versus LSA on TOEFL. In *Proc. of the 12th European Conference on Machine Learning*, pages 491–502.

Varadarajan, Ramakrishna and Vagelis Hristidis. 2006. A system for query-specific document summarization. In *Proc. of the 15th ACM international conference on Information and knowledge management*, pages 622–631.

Wang, Changhu, Feng Jing, Lei Zhang, and Hong-Jiang Zhang. 2007. Learning query-biased Web page summarization. In *Proc. of the sixteenth ACM conference on information and knowledge management*, pages 555–562.

White, Ryen W., Joemon M. Jose, and Ian Ruthven. 2003. A task-oriented study on the influencing effects of query-biased summarisation in Web searching. *Information Processing and Management*, 39(5):707–733.

Ye, Shiren, Tat-Seng Chua, and Jie Lu. 2009. Summarizing definition from Wikipedia. In *Proc. of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1*, pages 199–207.

Semi-Supervised WSD in Selectional Preferences with Semantic Redundancy

Xuri TANG[1,5] , Xiaohe CHEN[1] , Weiguang QU[2,3] and Shiwen YU[4]
1. School of Chinese Language and Literature, Nanjing Normal University
{xrtang,chenxiaohe5209}@126.com
2. Jiangsu Research Center of Information Security & Privacy Technology
3. School of Computer Science, Nanjing Normal University
wgqu_nj@163.com
4. Institute of Computational Linguistics , Peking University
yusw@pku.edu.cn
5. College of Foreign Studies, Wuhan Textile University

Abstract

This paper proposes a semi-supervised approach for WSD in Word-Class based selectional preferences. The approach exploits syntagmatic and paradigmatic semantic redundancy in the semantic system and uses association computation and minimum description length for the task of WSD. Experiments on Predicate-Object collocations and Subject-Predicate collocations with polysemous predicates in Chinese show that the proposed approach achieves a precision which is 8% higher than the semantic-association based baseline. The semi-supervised nature of the approach makes it promising for constructing large scale selectional preference knowledge base.

1 Introduction

This paper addresses word sense disambiguation (WSD) which is required in the construction of selectional preference (SP) knowledge database. In previous literature of SP, four different types of formalization models are explicitly or implicitly employed. Two types are distinguished in Li and Abe(1998):

Word Model: $P(n \mid v,r) = \sigma$ (1)

Class Model: $P(C \mid v,r) = \sigma$ (2)

where *v* stands for verb, *n* for noun, *C* for the semantic class of *n*, *r* for the grammatical relation between *v* and *n,* and P for the preference strength. Most of the researches(Resnik 1996; Li and Abe 1998; Ciaramita and Johnson 2000; Brockmann and Lapata 2003; Light and Greiff 2002) uses the class model, and a few(Erk 2007) uses the word model. The other two types of model are given as below:

Class-Only Model: $P(C_n \mid C_v,r) = \sigma$ (3)

Word-Class Model: $P(n,C_n \mid v,C_v,r) = \sigma$ (4)

where C_n , C_v are semantic classes for the noun and verb respectively. Class-Only model considers solely the semantic classes, while Word-Class model considers both words and semantic classes. Agirre and Martinez(2001) and Zheng et al(2007) adopted the Class-only Model in research, while in McCarthy and Carroll(2003) and Merlo and Stevenson(2001) the Word-Class Model is employed.

Among the four models, the Word-Class Model is the type which possesses the most granulated knowledge and is the most potential in applications. McCarthy and Carroll(2003) reports that the Word-Class Model performs well in unsupervised WSD. In other NLP tasks such as metaphor recognition, this model may be indispensable. For instance, to distinguish the predicate verb "浮动(float)" in Ex(1a) as

Ex. 1

a. 树 叶 浮 动 (leaf floats)　　b. 价 格 浮 动 (price floats)

Coling 2008: Poster Volume, pages 1238–1246,
Beijing, August 2010

literal and Ex(1b) as metaphorical requires different interpretations of the verb.

The present research is concerned with WSD as in the Word-Class model. Particularly, it aims at disambiguating predicates in subject-predicate (Subj-Pred) and predicate-object (Pred-Obj) constructions. The motivations behind the research are two folds. Firstly, semi-supervised and unsupervised WSD in SP are not fully explored. Merlo and Stevenson(Merlo and Stevenson 2001) employs supervised learning from large annotated corpus, which is difficult to obtain. One known unsupervised learning approach for WSD in SP is McCarthy and Carroll(2003) which addresses the issue via conditional probability. The other motivation derives from the fact few research is done on selectional preferences in languages other than English, as is stated in Brockmann and Lapata(2003). For instance, studies on construction of SP knowledge database in Chinese can only be found in Wu et al(2005), Zhen et al(2007), Jia and Yu(2008) and some others.

The basic idea of the approach proposed for WSD in the paper is that the most acceptable interpretation of senses for a given construction is the pair of senses which encodes the most redundant information in the semantic system of the language. Two principles, namely Syntagmatic Redundancy Principle and Paradigmatic Redundancy Principle, are proposed in the paper to capture the intuition. Two corresponding devices are employed to model the two principles: Association for Syntagmatic Redundancy Principle and Minimum Description Length for Paradigmatic Redundancy Principle. Two experiments are conducted in the paper. The first is based on semantic association, achieving a 61.98% precision for predicates in Subj-Preds and 62.54% in Pred-Objs. This experiment is used as baseline as the approach is also used in McCarthy and Carroll(2003) for verb and adjective disambiguation. In the second experiment, both semantic association and MDL are employed, the precision of WSD amounts to 69.88% and 69.09% for predicates in Subj-Preds and Pred-Objs respectively, indicating that a combination of the two devices are fairly effective in disambiguating word senses for SP.

The rest of the paper is organized as below. The second part gives further illustration of the rationale for the approach. The third part describes the procedure and the fourth part discusses the experiment result. The thesis concludes with some speculations in further researches.

2 Rationale

2.1 Task Formalization

Consider a Subj-Pred or Pred-Obj collocation C=$<W_{pred}, W_{\arg}>$, where W_{pred} is the word of predicate and $W_{\arg}$ is the word of argument. W_{pred} has M senses, denoted by set S_{pred}. $W_{\arg}$ has N senses, denoted by $S_{\arg}$. The possible interpretation of C has M*N possibilities, denoted by $S_C = S_{pred} \times S_{\arg} = \{ \varsigma^i_j \mid \varsigma^i_j = < s^i_{pred}, s^j_{\arg} > \}$, where ς^i_j is called a sense collocation. The task of WSD is to search for a particular sense collocation in S_C and assign it to C as its interpretation. At the initial stage, each sense collocation in S_C is considered to have an even number of frequency, namely $f(\varsigma^i_j) = 1/(N \times M)$. Accordingly, for each $s^i_{pred} \in S_{pred}$, $f(s^i_{pred}) = 1/M$, For each $s^i_{\arg} \in S_{\arg}$, $f(s^i_{\arg}) = 1/N$.

2.2 Syntagmatic Redundancy Principle

Syntagmatic Redundancy Principle (SRP) can be stated as following: among all possible sense collocations for a word collocation, the most appropriate is the one in which senses exhibit the most redundant information between each other.

The syntagmatic redundancy between words has been noticed very early by linguists and has been applied in WSD. Firth(1957) argues that there exists “mutual expectancy” between words in collocations, and the meaning of word is partially encoded in its juxtaposition. Lyons(1977:261) comments that Porzig has noticed in 1934 the “essential meaning relation” between words of collocations like “dog barks” and “tree fells”

and emphasizes that the meanings of collocationally restricted lexemes such as "bark" and "fell" can only be explained by taking into account the collocates they occur with. This notion is also employed in Yarowsky(1995) for WSD, in which the key is the "one-sense-per-collocation" statement. McCarthy and Carroll(2003) also uses this type of redundancy for disambiguation in SP.

SRP can be explained as a statistic correlation between s_{pred} and $s_{\arg}$. The more co-relevant these two senses are, the more likely the pair is to be accepted as the appropriate interpretation. This can be described as below:

$$\overline{\varsigma_j^i} = \arg\max Assoc(s_{pred}^i, s_{\arg}^j) \qquad (5)$$

where $Assoc(s_{pred}^i, s_{\arg}^j)$ is the function for sense association. Four methods can be considered for association computation: conditional probability (Formula 6 and 7), Lift(Han and Kamber 2006:261) (Formula 8), All-Confidence(Han and Kamber 2006:263) (Formula 9) and cosine (Formula 10). Note that two versions of conditional probability are considered, as are denoted in Formula 6 and 7. The first version, Cond-Prob 1, takes argument sense as condition, while the second version Cond-Prob 2 takes predicate sense as condition.

$$p(s_{pred}^i \mid s_{\arg}^j) = \frac{p(s_{pred}^j, s_{\arg}^i)}{p(s_{\arg}^j)} \qquad (6)$$

$$p(s_{\arg}^i \mid s_{pred}^j) = \frac{p(s_{pred}^j, s_{\arg}^i)}{p(s_{pred}^i)} \qquad (7)$$

$$lift(s_{pred}^i, s_{\arg}^j) = \frac{p(s_{pred}^i, s_{\arg}^j)}{p(s_{pred}^i) * p(s_{\arg}^j)} \qquad (8)$$

$$all\_conf(s_{pred}^i, s_{\arg}^j) = \frac{f(s_{pred}^i, s_{\arg}^j)}{\max(f(s_{\arg}^i), f(c_{pred}^j))} \qquad (9)$$

$$\cos ine(s_{pred}^i, s_{\arg}^j) = \frac{p(s_{pred}^i, s_{\arg}^j)}{\sqrt{p(s_{pred}^i) * p(s_{\arg}^j)}} \qquad (10)$$

2.3 Paradigmatic Redundancy Principle

Paradigmatic Redundancy Principle (PRP) can be stated as following: among all possible sense collocations for a word collocation, the most appropriate is the one which is also implicitly or explicitly expressed by other synonymous, metonymic or metaphorical word collocations.

Ex(2) illustrates the explicit redundancy in synonymous and metaphorical ways, in which the sense collocation "[Price|价格] [QuantityChange|量变]" is expressed by five word collocations, each with a different predicate：变化(change), 浮动(float), 调整(adjust), 起伏(go up and down), 变动(alter).

Ex 2.

a. 价格变化 price changes
b. 价格浮动 price floats
c. 价格调整 price adjusts
d. 价格起落 price goes up and down
e. 价格变动 price alters

Ex(3) reveals the implicit redundancy in metonymic way, in which the meaning "人(human) 安心(is eased)" is implicitly expressed in all the six collocations, established by semantic relatedness among the arguments "马拉多纳(Maradona)", "学生(student)", "工作(work)", "劳动(labour)", "行车(driving)", and "生活(life)".

Ex 3.

a. 马拉多纳 安心 了 Maradona is eased
b. 学生 安心 了 Student is eased
c. 工作 安心 了 work is eased
d. 劳动 安心 了 labour is eased
e. 行车 安心 了 driving is eased
f. 生活 安心 了 Life is eased

To apply PRP, WSD in SP is casted as an issue of model selection. Given a set of word collocations Θ, the process of WSD is to assign to each word collocation one sense collocation from a number of possibilities. Those assigned sense collocations form a set, or a model for Θ. The goal of WSD in SP is to select from all those models the one which best interprets Θ. For this purpose, Miminum Description Length(Barron et al. 1998; Michell 2003; MacKay 2003) can be used. MDL selects models by relying on induction bias based on Occam's Razor, which stipulates that the simplest solution is usually the correct one. One way to interpret MDL in Bays' analysis is as below(Michell 2003:124):

$$m' = \arg\min L_M(m) + L_D(D \mid m) \qquad (11)$$

In (11), $L_M(m)$ is the model description length when model m is considered, $L_D(D \mid m)$ is the data description length when model m is used for description. The model with minimum length is the best model.

For model description length, we have adopted the method used in (Li and Abe 1998) which considers only the size of the model:

$$L_M(m) = \frac{size(m)-1}{2}\log(N) \qquad (12)$$

where size(m) is the number of sense collocation contained in model m, and N is the number of word collocation in consideration. In this study, the set of word collocation with the same predicate word, denoted by Θ, is used as the unit for model description length calculation instead of the whole corpus, so as to reduce computation complexity. Accordingly, each word collocation in Θ can be assigned one and only one sense collocation in the model m, out of all the potential sense collocations as is explained in section 2.1.

Data description length is calculated on model m and Θ, as is denoted in formulas (13), (14) and (15) below. The calculation is

$$L(\Theta \mid m) = -\sum \log(p(\varsigma_j^i)) = -\sum \log(\frac{\overline{f}(\varsigma_j^i)}{Num_\Theta{}^2}) \qquad (13)$$

$$\overline{f}(\varsigma_j^i) = f(\varsigma_j^i) + f(\varsigma_j^i) * \sum_{\varsigma_j^i, \varsigma_l^k \in m} w(\varsigma_j^i, \varsigma_l^k) \qquad (14)$$

$$w(\varsigma_j^i, \varsigma_l^k) = rel(< s_{pred}^i, s_{\arg}^j >, < s_{pred}^k, s_{\arg}^l >) = \begin{cases} rel(s_{\arg}^j, s_{\arg}^l) & \text{if } s_{\text{pred}}^i = s_{pred}^k \\ 0 & \text{if } s_{\text{pred}}^i \neq s_{pred}^k \end{cases} \qquad (15)$$

based on the probability of sense collocation $\varsigma_j^i =< s_{pred}^i, s_{\arg}^j >$, which in turn is calculated on a modified frequency of the collocation $\overline{f}(\varsigma_j^i)$. The frequency is modified by counting the explicit occurrence of the sense collocation itself and the implicit occurrence expressed by other sense collocations in Θ. This idea is equivalent to enlarge the corpus by 1 fold, thus the overall collocation number is the two times of the original number.

The modified frequency is a sum of two parts, denoted in formula (14). The first part is $f(\varsigma_j^i)$, the frequency of ς_j^i. The second part is the weighted frequency of ς_j^i. The weight is determined by the relatedness of the sense collocation ς_j^i and all the other sense collocation ς_l^k in the model ***m***. According to this formula, if the sense collocation is found to be more similar to other sense collocations, it should obtain a higher modified frequency, and thus more likely to be the correct one for the word collocation.

The way to calculate the weight is given in formula (15). If two sense collocations have identical predicate sense, namely $s_{\text{pred}}^i = s_{pred}^k$, then the weight between the two sense collocations is measured by $rel(s_{\arg}^j, s_{\arg}^l)$, the semantic relatedness between the argument sense $s_{\arg}^j$ and $s_{\arg}^l$. Otherwise, 0 is returned. There are different ways to measure sense relatedness. The present study has used semantic similarity based on HowNet(Liu and Li 2002) to calculate the semantic relatedness.

3 Procedure

Figure 1 maps out the procedure for WSD in SP in the present study. The procedure is divided into two phases: data collection and disambiguation. The collocation data are collected from three sources: Sketch Engine, Collocation Dictionary and HowNet Examples. Two types of collocation data are collected: subject-predicate collocations (Subj-Pred) and predicate-object collocations (Pred-Obj) from Sketch Engine and Collocation Dictionary. Collocation Retriever reduces HowNet examples into Subj-Preds and Pred-Objs using simple heuristic methods. As a result, about 70,000 subject-predicate collocations and 106,000 predicate-object collocations are obtained.

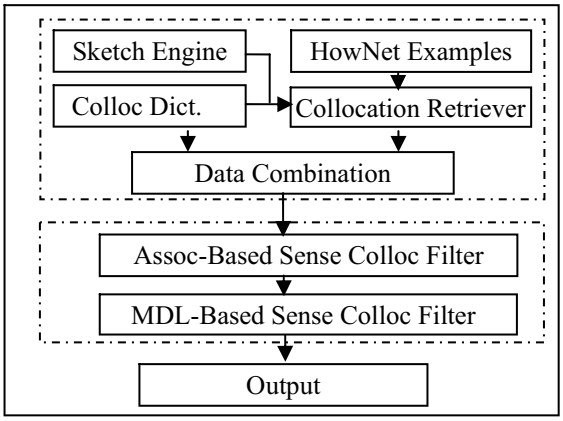


Figure 1. WSD Procedure

In disambiguation phase, two devices are employed to filter out unlikely sense collocations: Association-Based Sense Collocation Filter, following SRP, and MDL-Based Sense Collocation Filter, following PRP.

In this phase, Subj-Preds and Pred-Objs are processed independently but following the same route.

Each phase alone can perform WSD independently. Accordingly, two experiments are conducted to evaluate the method proposed in this paper. The first experiment uses association-based filter for word sense disambiguation, which is also used as the baseline. The approach is also used in (McCarthy and Carroll 2003) to disambiguate verbs and adjectives in collocations. To be particular, the method used by McCarthy and Carroll(2003) is formula (6). The second experiment is based on the result of the first one so as to observe the improvement obtained by MDL-Based approach. In the second experiment, unsupervised and semi-supervised WSD are also investigated by including some annotated collocations in the evaluation data.

Two corpora are constructed for evaluation. One corpus is a set of 1034 subject-predicate constructions. The other is a set of 1841 predicate-object constructions. Both are manually annotated by the authors with sense definitions defined in HowNet(Dong 2006). All together there are 52 highly ambiguous predicates involved in the study.

4 Experiments and Discussion

4.1 Collocation Retriever

The major task in data collocation is in Collocation Retriever, which retrieves collocations from HowNet examples. Ex(4) gives a partial entry structure in HowNet,

```
Ex 4.
W_C=浮动
E_C=人心~，开始~，工作指标不停地~
DEF=[change|变]
```

in which W_C stands for Chinese Word, DEF for definition, E_C for Examples of Chinese, and the wave "~" for the word in question. From E_C, possible Subj-Preds such as "人心(public opinion) 浮动(floats)", "指标(index) 浮动(floats)" can be retrieved, in which the sense of "浮动(float)" is annotated with DEF. But there are also noises. A simple heuristic method is applied to automatically filter out unwanted collocations. The heuristic method checks whether the collocation retrieved from HowNet share possible sense collocations with collocations in Collocation Dictionary. If yes, it is accepted as a collocation of the type, otherwise, it is rejected. Procedures are given below:

(a) Use Subj-Pred collocations and Pred-Obj collocations in Collocation Dictionary to build sense collocation set $\Gamma_{Subj-Pred}$ and $\Gamma_{Pred-Obj}$;

(b) For each example sentence in E_C, segment it using ICTCLAS[1] to obtain an array of words. Words before "~" forms potential Subj-Pred collocations $A_{Subj-Pred}$ and Words after form potential Pred-Obj collocations $B_{Pred-Obj}$.

(c) For each $a \in A_{Subj-Pred}$ or $b \in B_{Subj-Pred}$, construct possible sense collocation set Γ_a or Γ_b, if $\Gamma_a \cap \Gamma_{Subj-Pred} \neq \phi$ or $\Gamma_b \cap \Gamma_{Pred-Obj} \neq \phi$, add it as a Subj-Pred collocation or Pred-Obj collocation.

Evaluation on partial retrieved collocations shows that about 70% of obtained collocations are valid collocations, while about 30% are errors. Thus manual edition has been applied to rid those invalid collocations.

4.2 Association-Based Filter

Association-Based Sense Collocation Filter filters out those sense collocations that are very unlikely to be the right interpretation for a word collocation. Table 1 gives association computation result for the six senses related to the predicate "粗(rough)" in Subj-Pred collocation "性格(personality) 粗(rough)". The 2nd, 3rd, 4th, and 6th are very unlikely interpretations and should be filtered, while the 5th seems to be the most appropriate.

Table 1. Association-Based Filter Example

No.	Pred Sense	Arg Sense	Assoc. Dgr
1	[Behavior|举止]	[careless|粗心]	0.0019
2	[Behavior|举止]	[coarse|糙]	0.0002
3	[Behavior|举止]	[hoarse|沙哑]	0.0004
4	[Behavior|举止]	[roughly|大概]	0.0002
5	[Behavior|举止]	[vulgar|俗]	0.0071
6	[Behavior|举止]	[widediameter|粗]	0.0002

Following the procedure in Figure 1, to filter out those unlikely sense collocations, average

[1] A Chinese segmentation system, please refer to http://www.ictclas.org for further information.

association value is used as the filter and those below the average are dropped and those above are chosen for MDL-Based Filter. In Table 1, the average is 0.0017, and the 1rd and 5th are chosen.

However, in order to obtain a baseline and to decide which association computation model to use, we have followed the definition in Formula 5 and perform WSD test by choosing the sense collocation with highest association as the correct sense tags. for used this step solely for WSD, as is defined in Formula 4. Table 2 gives the experiment results for Subj-Pred and Pred-Obj collocations with all the association computation models denoted in Formula 6-10.

Table 2. WSD Result by Association

	Subj-Pred(%)	Pred-Obj(%)
Cond-Prob 1	61.98	62.54
Cond-Prob 2	55.15	42.4
Lift	63.09	40.84
All_Conf	56.16	48.54
Cosine	58.83	55.72

One interesting phenomenon about all the five models is null-invariance. In selecting models for association computation, null-invariance is an important feature to be considered(Han and Kamber 2006). A model with null-invariance is not influenced by additional irrelevant data and thus is more stable. In the experiment, the model Lift is the only one not featured with null-invariance. The experiments show that Lift is not stable in different collocation types, achieving high precision in Subj-Pred but low precision in Pred_Obj.

A second interesting phenomenon is collocation directionality exposed by the experiments, which can be observed in the two models of conditional probability: Cond-Prob 1, with argument as condition, and Cond-Prob 2, with predicate as condition. Directionality in collocation has been noticed earlier in some researches, for example Qu(2008). Our experiment shows that when using Cond-Prob 1, we are able to get a precision of 61.98% and 62.54% for Subj-Pred and Pred-Obj respectively, while Cond-Prob 2 gets a much lower precision. This fact can be interpreted that arguments tend to have a stronger selectional preference strength, and the possible selection range is comparatively narrower, while predicates have weaker selectional preference strength and a wider selectional range.

4.3 MDL-Based Filter

MDL-Based Filter takes as input result from Association-Based Filter using Cond-Prob 1 for association computation and average association as filter. Table 3 and 4 give the final experiment outcome for Pred-Obj and Subj-Pred constructions and individual predicates.

It can be seen in Table 3 that MDL-Based Filter Several inferences can be made from the experiments. Firstly, comparison between Association-Based WSD (Table 2) and MDL WSD (Table 3) shows that MDL can improve overall performance up to 8%. As is mentioned earlier, Association-Based WSD is used as baseline in the present study. Given the fact that the average number of senses for word in question is fairly high, the improvement is considered as significant.

Table 3. General WSD Results[2]

	Ave. N.O.S.	Assoc. WSD (%)	MDL WSD (%)
Subj-Pred	4.16	61.98	69.09
Pred-Obj	5.03	62.54	69.88

Analysis on the individual predicates in Table 4 gives a clearer picture of WDL-based WSD. Firstly, it can be seen that MDL is especially effective when the demarcation of word senses is clear-cut. Predicate words such as “安静(quiet)”, “肮脏(dirty)”, “困难(difficult)” in Subj-Preds and “锤炼(beat)”, “触动(touch)” and “打断(break)” in Pred-Objs are successfully disambiguated in Table 4. These words generally have 2 or 3 senses, and the senses generally differ in terms of abstractness and concreteness, as is indicated in table 5. This is due to the fact that the arguments in these collocations are clearly delimitated in HowNet and this delimitation is well captured by the modified frequency calculation defined in formula (14). Via the formula, the concrete sense collocations can

[2] In Table 3 and 4, Ave. N.O.S stands for average number of senses of predicates, N.O.S stands for number of senses of the predicate, Assoc. WSD stands for Association-based WSD, and MDL WSD stands for MDL-based WSD.

Table 4. Detailed WSD Experiment Results

Results for Pred-Obj.				Results for Subj-Pred.			
Pred.	N.O.S	Assoc. WSD (%)	MDL WSD (%)	Pred.	N.O.S.	Assoc. WSD (%)	MDL WSD (%)
追(v)	5	69.23	80.77	困难(a)	2	61.14	92.00
上(v)	14	70.59	70.59	垮(v)	2	72.73	86.36
交(v)	6	56.25	90.62	细腻(a)	2	47.83	58.7
介绍(v)	3	72.22	88.89	错(a/v)	5	52.17	78.26
倒(v)	9	50	60.53	肮脏(a)	3	56.76	81.08
停(v)	8	86.67	93.33	浓(a)	5	40	40
套(v)	5	68.75	62.5	破裂(v)	2	55.17	41.38
展开(v)	3	73.91	81.16	安静(a)	3	75.76	93.94
开(v)	17	55.93	44.07	沉(a)	4	96.3	66.67
把握(v)	3	80.36	78.57	沉闷(a)	3	47.37	42.11
打断(v)	2	66.67	92.31	淡(a)	6	88.24	88.24
洗刷(v)	2	57.14	80.95	低(a)	6	46	60
挨(v)	6	76.27	79.66	开发(v)	3	44.44	44.44
锤炼(v)	3	83.33	100	开阔(a)	2	38.46	65.38
磨(v)	8	63.64	63.64	麻木(a)	2	93.33	53.33
排(v)	3	77.14	80	漂浮(v)	3	85.19	88.89
触动(v)	2	88.24	100	轻(a)	10	50	50
散布(v)	2	83.87	80.65	陷落(v)	2	60.53	63.16
看(v)	9	61.84	68.42	长(a/v)	9	39.66	53.45
陷入(v)	3	40.28	51.39	粗(a)	6	59.46	51.35
带(v)	4	48.08	53.85	掉(v)	6	48.72	74.36
破坏(v)	3	73.49	73.49	动摇(v)	3	48.15	44.44
恢复(v)	2	15.32	40	黑暗(a)	2	88.57	57.14
糟蹋(v)	2	84.91	83.02	厚(a)	6	68.18	40.91
找(v)	3	86.54	85.58	坏(v)	8	52.03	65.04
爱(v)	4	72.51	72.99	结实(a)	2	95.35	95.35

Table 5. Word Sense Distinction

Pred	Concrete Sense	Abstract Sense(s)
安静	[quiet|静]	[calm|镇静], [peaceful|宁]
肮脏	[dirty|龌]	[despicable|卑劣], [immoral|不道德]
困难	[difficult|难]	[poor|穷]
锤炼	[beat|打]	[MakeBetter|优化], [cultivate|培养]
触动	[touch|触]	[excite|感动]
打断	[break|折断]	[obstruct|阻止]

increase the modified frequency of concrete sense collocations, and the abstract sense collocation can increase the modified frequency of abstract sense collocations, thus leading to the clear demarcation of abstract senses and concrete senses.

The role of semantic relevance can also be clearly noticed in the predicates which have a decreased precision in MDL in Table 4. Via Paradigmatic Redundancy Principle, the information encoded in one collocation are diffused to other collocations. Consequently, errors can be diffused. This explains why the precisions of some predicates such as "沉(sink)", "麻木(dumb)", "黑暗(dark)" in Subj-Pred and "开(open)", "套(harness)" in Pred-Objs decrease after MDL. Further analysis shows that this is because MDL has diffused the errors produced by Association Filter. For instance, at Association Filter phase, the collocation "箱子(box) 沉(sink)" is assigned with the only sense collocation "[tool|用具] [very|很]" and all other potential sense collocations are filtered. When MDL is applied, other collocations such as "机器(machine) 沉(heavy)", "镐头(pick) 沉(heavy)", "镢头(chaw) 沉(heavy)", "篮子(basket) 沉(heavy)", "盒子(box) 沉(heavy)", "家具(furniture) 沉(heavy)", in which the arguments are tightly correlated with that of "箱子(box) 沉(sink)", all takes the sense "[very|很]", thus leading to the decrease of precision.

The diffusion of senses can also best seen in the comparison between those predicates whose WSD are semi-supervised and those whose WSD are not supervised. Some predicates have collocations successfully retrieved from HowNet examples in which the word sense is already identified. These collocations are diffused in MDL filtering and play important roles in improving precision, while some other predicates do not have such resource. In Table 4, those unsupervised predicates are "陷落(fall)", "垮(collapse)", "细腻(exquisite)", "麻木(dumb)", "开阔(wide)", "开发(develop)" in Subj-Preds and "展开(spread)", "洗刷(brush)", "陷入(get into)", "带(bring)", and "糟蹋(mar)" in Pred-Objs. The other predicates are semi-supervised. As can be seen in Table 4, most of these unsupervised predicates generally have a precision of 40%-60%, while those semi-supervised predicates enjoy are much higher precision between 50%-100%. The explanation

for the result is straight forward. When one sense collocation of one word collocation is correctly identified, by way of Paradigmatic Redundancy Principle, the sense collocation which is similar to the correctly identified will have a higher modified frequency and is thus singled out as the best choice. This feature of MDL has great significance in the process of annotating large scale collocation data. With only a small number of annotated collocations for each predicate, a fairly high precision can be achieved for all the rest of the data through MDL.

5 Conclusion

The present paper believes that the Word-Class Model gives the fullest description for selectional preference and thus makes efforts to disambiguate predicates in selectional preferences. From the perspective of semantic system, two principles of semantic redundancy, namely the Syntagmatic Redundancy Principle and Paradigmatic Redundancy Principle, are proposed in the paper and are applied in WSD in SP via Association Computation and Minimum Description Length. The experiments show that the approach proposed is fairly encouraging in disambiguation of polysemous predicates, especially under semi-supervised conditions when a small portion of data is annotated. With such a tool, we are able to build large scale selectional preference knowledge database based on Word-Class Models, which can be applied in various tasks, of which metaphor recognition is the particular one we bear in mind.

Acknowledgement

This work is supported by Chinese National Fund of Social Science under Grant 07BYY050 and Chinese National Science Fund under Grant 60773173 and Chinese National Fund of Social Science under Grant 10CYY021. We are also grateful to the autonomous reviewers for their valuable advice and suggestions.

References

Agirre, E., and D. Martínez. 2001. Learning class-to-class selectional preferences. Paper read at Proceedings of the Conference on Natural Language Learning, at Toulouse, France.

Barron, A. R., J. Rissanen, and B. Yu. 1998. The Minimum Description Length Principle in coding and modeling. IEEE Transactions on Information Theory 44 (6):2743-2760.

Brockmann, C., and M. Lapata. 2003. Evaluating and combining approaches to selectional preference acquisition. Paper read at Proceedings of the European Association for Computational Linguistics, at Budapest, Hungary.

Ciaramita, M., and M. Johnson. 2000. Explaining away ambiguity: Learning verb selectional preference with Bayesian networks. In Proceedingsofthe18thInternationalConferenceon ComputationalLinguistics (COLING 2000), 187-193.

Dong, Z. 2006. HowNet and the Computation of Meaning. River Edge, NJ: World Scientific.

Erk, K. 2007. A Simple, Similarity-based Model for Selectional Preferences. Paper read at Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, at Prague, Czech Republic.

Firth, J. R. 1957. A Synopsis of Linguistic Theory, 1930-1955. In Studies in Linguistic Analysis. Oxford: Blackwell, 1-32.

Han, J., and M. Kamber. 2006. Data Ming: Concepts and Techniques. Singapore: Elsevier.

Jia Yuxiang and Yu Shiwen. 2008. Automatic Acquisition of Selectional Preference and Its Application to Metaphor Processing. Paper read at the Fourth National Student Conference on Computationl Linguistics, at Taiyuan, Shangxi, China.

Li, H., and N. Abe. 1998. Generalizing Case Frames Using a Thesaurus and the MDL Principle. Computational Linguistics 24 (2):217-244.

Light, M., and W. Greiff. 2002. Statistical models for the induction and use of selectional preferences. Cognitive Science 87:1-13.

Liu, Qun and Li Sujian. 2002. Word Similarity Computation Based on HowNet. In Proceedings of the 3rd Chinese Lexical Semantics. Taibei, China.

Lyons, J. 1977. Semantics. Cambridge: Cambridge University Press.

MacKay, D. J. C. 2003. Information Theory, Inference, and Learning Algorithms. Cambridge: Cambridge University Press.

McCarthy, D., and J. Carroll. 2003. Disambiguating Nouns, Verbs, and Adjectives Using Automatically Acquired Selectional Preferences. Computational Linguistics 29 (4):639-654.

Merlo, P., and S. Stevenson. 2001. Automatic Verb Classification Based on Statistical Distributions of Argument Structure. Computational Linguistics 27 (3):374-408.

Michell, Tom M.. Machine Learning. Translated by Zen Huajun and Zhang Yinkui. Beijing: China Machine Press.

Resnik, P. 1996. Selectional constraints: an information-theoretic model and its computational realization. Cognition 61:127-159.

Qu, Weiguang. 2008. Lexical Sense Disambiguation in Modern Chinese. Beijing: Science Press.

Wu, Yunfang, Duan Huiming and Yu Shiwen. 2005. Verb's Selectional Preference on Object. Spoken and Written Language in Practice 2005(2):121-128.

Yarowsky, D. 1995. Unsupervised Word Sense Disambiguation Rivaling Supervised Methods. Paper read at Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics, at Cambridge, MA.

Zheng, Xuling, Zhou Changle, Li Tangqiu and Chen Yidong. 2007. Automatic Acquisition of Chinese Semantic Collocation Rules Based on Association Rule Mining Technique. Journal of Xiamen University (Natural Science) 46(3):331-336.

A Comparison of Models for Cost-Sensitive Active Learning

Katrin Tomanek and **Udo Hahn**
Jena University Language & Information Engineering (JULIE) Lab
Friedrich-Schiller-Universität Jena
http://www.julielab.de

Abstract

Active Learning (AL) is a selective sampling strategy which has been shown to be particularly cost-efficient by drastically reducing the amount of training data to be manually annotated. For the annotation of natural language data, cost efficiency is usually measured in terms of the number of tokens to be considered. This measure, assuming uniform costs for all tokens involved, is, from a linguistic perspective at least, intrinsically inadequate and should be replaced by a more adequate cost indicator, *viz.* the time it takes to manually label selected annotation examples. We here propose three different approaches to incorporate costs into the AL selection mechanism and evaluate them on the MUC7$_\mathcal{T}$ corpus, an extension of the MUC7 newspaper corpus that contains such annotation time information. Our experiments reveal that using a cost-sensitive version of semi-supervised AL, up to 54% of true annotation time can be saved compared to random selection.

1 Introduction

Active Learning (AL) is a selective sampling strategy for determining those annotation examples which are particularly informative for classifier training, while discarding those that are already easily predictable for the classifier given previous training experience. While the efficiency of AL has already been shown for many NLP tasks based on measuring the number of tokens or sentences that are saved in comparison to random sampling (e.g., Engelson and Dagan (1996), Tomanek et al. (2007) or Settles and Craven (2008)), it is obvious that just counting tokens under the assumption of *uniform* annotation costs for each token is empirically questionable, from a linguistic perspective, at least.

As an alternative, we here explore annotation costs that incur for AL based on an empirically more plausible cost metric, *viz.* the time it takes to annotate selected linguistic examples. We investigate three approaches to incorporate costs into the AL selection mechanism by modifying the standard (fully supervised) mode of AL and a non-standard semi-supervised one according to cost considerations. The empirical backbone of this comparison is constituted by MUC7$_\mathcal{T}$, a re-annotation of a part of the MUC7 newspaper corpus that contains annotation time information (Tomanek and Hahn, 2010).

2 Active Learning

Unlike random sampling, AL is a selective sampling technique where the learner is in control of the data to be chosen for training. By design, the intention behind AL is to reduce annotation costs, usually considered as the amount of labeled training material required to achieve a particular target performance of the model. The latter is yielded by querying labels only for those examples which are assumed to have a high training utility. In this section, we introduce different AL frameworks – the default, fully supervised AL approach (Section 2.1), as well as a semi-supervised variant of it (Section 2.2). In Section 2.3 we then propose three methods how these approaches to AL can be made cost-sensitive without further modifications.

Coling 2008: Poster Volume, pages 1247–1255,
Beijing, August 2010

2.1 Fully Supervised AL (FuSAL)

As we consider AL for the NLP task of Named Entity Recognition (NER), some design decisions have to be made. Firstly, the selection granularity is set to complete sentences – a reasonable linguistic annotation unit which still allows for fairly precise selection. Second, a batch of examples instead of a single example is selected per AL iteration to reduce the computational overhead of the sampling process.

We base our approach to AL on Conditional Random Fields (CRFs), which we employ as base learners (Lafferty et al., 2001). For observation sequences $\vec{x} = (x_1, \ldots, x_n)$ and label sequences $\vec{y} = (y_1, \ldots, y_n)$, a linear-chain CRF is defined as

$$P_\theta(\vec{y}|\vec{x}) = \frac{1}{Z_\theta(\vec{x})} \cdot \prod_{i=1}^{n} \exp \sum_{j=1}^{k} \lambda_j f_j\Big(y_{i-1}, y_i, \vec{x}, i\Big)$$

where $Z_\theta(\vec{x})$ is the normalization factor, and k feature functions $f_j(\cdot)$ with feature weights $\theta = (\lambda_1, \ldots, \lambda_k)$ appear.

The core of any AL approach is a utility function $u(p, \theta)$ which estimates the informativeness of each example p, a complete sentence $p = (\vec{x})$, drawn from the pool P of all unlabeled examples, for model induction. For our experiments, we employ two alternative utility functions which have produced the best results in previous experiments (Tomanek, 2010, Chapter 4). The first utility function is based on the confidence of a CRF model θ in the predicted label sequence $\vec{y}^*$ which is given by the probability distribution $P_\theta(\vec{y}^*|\vec{x})$. The utility function based on this probability boils down to

$$u_{\mathrm{LC}}(p, \theta) = 1 - P_\theta(\vec{y}^*|\vec{x})$$

so that sentences for which the predicted label sequence $\vec{y}^*$ has a low probability is granted a high utility. Instead of calculating the model's confidence on the complete sequence, we might alternatively calculate the model's confidence in its predictions on single tokens. To obtain an overall confidence for the complete sequence, the average over the single token-confidence values can be computed by the marginal probability $P_\theta(y_i|\vec{x})$. Now that we are calculating the confidence on the token level, we might also obtain the performance of the second best label and calculate the margin between the first and second best label as a confidence score so that the final utility function is obtained by

$$u_{\mathrm{MA}}(p, \theta) = -\frac{1}{n} \sum_{i=1}^{n} \Big(\max_{y' in \mathcal{Y}} P_\theta(y_i = y'|\vec{x}) - \max_{\substack{y'' in \mathcal{Y} \\ y' \neq y''}} P_\theta(y_i = y''|\vec{x}) \Big)$$

Algorithm 1 formalizes our AL framework. Depending on the utility function, the best b examples are selected per round, manually labeled, and then added to the set of labeled data $\mathcal{L}$ which feeds the classifier for the next training round.

Algorithm 1 NER-specific AL Framework

Given:
b: number of examples to be selected in each iteration
$\mathcal{L}$: set of labeled examples $l = (\vec{x}, \vec{y}) \in \mathcal{X}^n \times \mathcal{Y}^n$
$\mathcal{P}$: set of unlabeled examples $p = (\vec{x}) \in \mathcal{X}^n$
$T(\mathcal{L})$: a learning algorithm
$u(p, \theta)$: utility function

Algorithm:
loop until stopping criterion is met

1. learn model: $\theta \leftarrow T(\mathcal{L})$
2. sort $p \in \mathcal{P}$: let $S \leftarrow (p_1, \ldots, p_m) : u(p_i, \theta) \geq u(p_{i+1}, \theta), i \in [1, m], p \in \mathcal{P}$
3. select b examples p_i with highest utility from S: $\mathcal{B} \leftarrow \{p_1, \ldots, p_b\}, b \leq m, p_i \in \mathcal{S}$
4. query labels for all $p \in \mathcal{B}$: $\mathcal{B}' \leftarrow \{l_1, \ldots, l_b\}$
5. $\mathcal{L} \leftarrow \mathcal{L} \cup \mathcal{B}', \mathcal{P} \leftarrow \mathcal{P} \setminus \mathcal{B}$

return $\mathcal{L}^* \leftarrow \mathcal{L}$ and $\theta^* \leftarrow T(\mathcal{L}^*)$

The specification is still not cost-sensitive as the selection of examples depends only on the utility function. Using u_{LC} will result in a reduction of the number of examples (i.e., sentences) selected irrespective of the sentence length so that a model learns the most from it. As a result, we observed that the selected sentences are quite long which might even cause higher annotation costs per sentence (Tomanek, 2010, Chapter 4). As for u_{MA} there is at least a slight normalization sensitive to costs since the sum over all token-level utility scores is normalized by the length of the selected sentence.

2.2 Semi-supervised AL (SeSAL)

Tomanek and Hahn (2009) extendeded this standard fully supervised AL framework by a semi-supervised variant (SeSAL). The selection of sentences is performed in a standard manner, i.e., similarly to the procedure in Algorithm 1. However, once selected, rather than manually annotating the complete sentence, only (uncertain) subsequences of each selected sentence are manually labeled, while the remaining (certain) ones are automatically annotated using the current version of the classifier.

After the selection of an informative example $p = (\vec{x})$ with $\vec{x} = (x_1, \ldots, x_n)$, the subsequences $\vec{x}' = (x_a, \ldots, x_b)$, $1 \leq a \leq b \leq n$, with low local uncertainty have to be identified. For reasons of simplicity, only sequences of length 1, i.e., single tokens, are considered. For a token x_i from a selected sequence $\vec{x}$ the model's confidence $C_\theta(y_i^*)$ in label y_i^* is estimated. Token-level confidence for a CRF is calculated as the marginal probability so that

$$C_\theta(y_i^*) = P_\theta(y_i = y_i^* | \vec{x})$$

where y_i^* specifies the label at the respective position of the predicted label sequence $\vec{y}^*$ (the one which is obtained by the Viterbi algorithm). If $C_\theta(y_i^*)$ exceeds a confidence threshold t, y_i^* is assigned as the putatively correct label. Otherwise, manual annotation of this token is required.

Employing SeSAL, savings of over 80 % of the tokens compared to random sampling are reported by Tomanek and Hahn (2009). Even when compared to FuSAL, still 60 % of the number of tokens are eliminated. A crucial question, however, not answered in these experiments, is whether this method actually reduces the overall annotation expenses in time rather than just in the number of tokens. Also SeSAL does not incorporate labeling costs in the selection process.

2.3 Cost-Sensitive AL (CoSAL)

In this section, we turn to an extension of FuSAL and SeSAL which incorporates cost sensitivity into the AL selection process (CoSAL). Three different approaches of CoSAL will be explored. The challenge we now face is that two contradictory criteria – utility and costs – have to be balanced.

2.3.1 Cost-Constrained Sampling

CoSAL can be realized in the most straightforward way by simply constraining the sampling to a particular maximum cost c_{max} per example. Therefore, in a pre-processing step all examples $p \in \mathcal{P}$ for which $\text{cost}(p) > c_{\text{max}}$ are removed from $\mathcal{P}$. The unmodified NER-specific AL framework can then be applied.

An obvious shortcoming of Cost-Constrained Sampling (CCS) is that it precludes any form of compensation between utility and costs. Thus, an exceptionally useful example with a cost factor slightly above c_{max} will be rejected. Another critical issue is how to fix c_{max}. If chosen too low, the pre-filtering of $\mathcal{P}$ results in a much too strong restriction of selection options when only few examples remain inside $\mathcal{P}$. If chosen too high, the cost constraint becomes ineffective.

2.3.2 Linear Rank Combination

A general solution to fit different criteria into a single one is by way of linear combination. If, however, different units of measurement are used, a transformation function for the alignment of benefit, or utility, and costs must be found. This can be difficult to determine. In our scenario, benefits measured by utility scores and costs measured in seconds are clearly incommensurable. As it is not immediately evident how to express utility in monetary terms (or vice versa), we transform utility and cost information into ranks $R\big(u(p,\theta)\big)$ and $R'\big(\text{cost}(p)\big)$ instead. As for utility, higher utility leads to higher ranks. As for costs, lower costs lead to higher ranks. The linear rank combination (LRK) is defined as

$$\phi_{\text{LRK}}(\vec{v}(p)) = \alpha R\big(u(p,\theta)\big) + (1-\alpha) R'\big(\text{cost}(p)\big)$$

where α is a weighting term. In a CoSAL scenario, where utility is the primary criterion, $\alpha > 0.5$ seems a reasonable choice. Alternatively, as costs and utility are contradictory, allowing equal influence for both criteria, as with $\alpha = 0.5$, it may be difficult to find appropriate examples in a medium-sized corpus. Thus, the choice of α depends on size and diversity with respect to combinations of utility and costs within $\mathcal{P}$.

2.3.3 Benefit-Cost Ratio

Our third approach to CoSAL is based on the Benefit-Cost Ratio (BCR). Given equal units of measurement for benefits and costs, the benefit-cost ratio indicates whether a scenario is profitable (ratio > 1). BCR can also be applied when units are incommensurable and a transformation function is available, as is the case for the combination of utility and cost. This holds as long as benefit and costs can be expressed in the same units by a linear transformation function, i.e., $u(p, \theta) = \beta \cdot \text{cost}(p) + b$. If such a transformation function exists, one can refrain from finding proper values for the above variables b and β and instead calculate BCR as

$$\phi_{\text{BCR}}(p) = \frac{u(p, \theta)}{\text{cost}(p)}$$

Since annotation costs are usually expressed on a linear scale, this is also required for utility, if we want to use BCR. But when utility is based on model confidence as we do it here, this property gets lost.[1] Hence a non-linear transformation function is needed to fit the scales of utility and costs. Assuming a linear relationship between utility and costs, BCR has already been applied by Haertel et al. (2008) and Settles et al. (2008). Our approach provides a crucial extension as we explicitly consider scenarios where such a linear relationship is not given and a non-linear transformation function is required instead.

In a direct comparison of LRK with BCR, LRK may be used when such a transformation function would be needed but is unknown and hard to find. Choosing LRK over BCR is also motivated by findings in the context of data fusion in information retrieval where Hsu and Taksa (2005) remark that, given incommensurable units and scales, one would do better when ranks rather than the actual scores or values were combined.

3 Experiments

In the following, we study possible benefits of CoSAL, relative to FuSAL and SeSAL, in the light of real annotation times as a cost measure (instead of the standard, yet inadequate one, *viz.* the number of tokens being selected). Such timing data is available in the MUC7$_\mathcal{T}$ corpus (Tomanek and Hahn, 2010), a re-annotation of the MUC7 corpus containing the ENAMEX types (persons, locations, and organizations) and a time stamp reflecting the time it took annotators to decide on each entity type. The MUC7$_\mathcal{T}$ corpus contains 3,113 sentences (76,900 tokens).

The results we report on are averaged over 20 independent runs. For each run, we split the MUC7$_\mathcal{T}$ corpus randomly into a pool to select from (90%) and an evaluation set (10%). AL was started from a random seed set of 20 sentences. As utility scores to estimate benefits we applied u_{MA} and u_{LC} as defined in Section 2.1.

The plots in the following sections depict costs in terms of annotation time (in seconds) relative to annotation quality (expressed via F1-scores). Learning curves are only shown for early AL iterations. Later on, in the convergence phase, due to the two conflicting criteria now considered simultaneously, selection options become more and more scarce so that CoSAL necessarily performs sub-optimally.

3.1 Parametrization of CoSAL Approaches

Preparatory experiments were run to analyze how different parameters affected different CoSAL settings. For the CCS and LRK experiments, we used the u_{LC} utility function.

For CCS, we tested three c_{max} values, *viz.* 7.5, 10, and 15, to determine the maximum performance attainable on MUC7$_\mathcal{T}$ when only examples below the chosen threshold were included. Our choices of the maximum were based on the distributions of annotation times over the sentences (see Figure 1) where 7.5s marks the 75% quantile and 15s is just above the 90% quantile. For 7.5s, we peaked at $F_{max} = 0.84$, for 10s at $F_{max} = 0.86$, and for 15s at $F_{max} = 0.88$. Figure 2 (top) shows the learning curves of CoSAL with CCS and different c_{max} values. With $c_{max} = 15$, as could be expected from the boxplot in Figure 1, no difference can be observed compared to cost-*in*sensitive FuSAL. CCS with lower values for c_{max} stagnates at the maximum perfor-

[1] Though normalized to $[0, 1]$, confidence estimates, especially for sequence classification, are often not on a linear scale so that confidence values that are twice as high do not necessarily mean that the benefit in training a model on such an example is doubled.

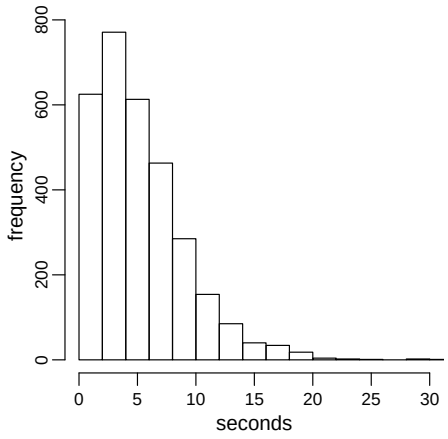


Figure 1: Distribution of annotation times per sentence in MUC7$_\mathcal{T}$.

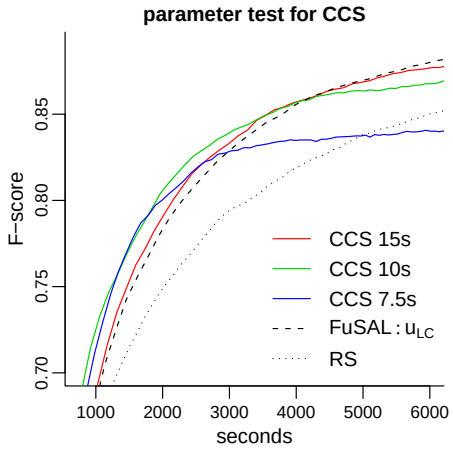
mance reported above, but still improves upon cost-*in*sensitive FuSAL in early AL iterations.

At some point in time all economical examples, with costs below c_{max} but high utility, have been consumed from the corpus. Even in a corpus much larger than MUC7$_\mathcal{T}$ this effect will only occur with some delay. Indeed, any choice of a restrictive value for c_{max} will cause similar exhaustion effects. Unfortunately, it is unclear how to tune c_{max} suitably in a real-life annotation scenario where pretests for maximum performance for a particular c_{max} are not possible. For further experiments, we chose $c_{max} = 10$.

For LRK, we tested three different weights α, *viz.* 0.5, 0.75, and 0.9. Figure 2 (bottom) shows their effects on the learning curves. Similar tendencies as for c_{max} for CCS can be observed. With $\alpha = 0.9$, CoSAL does not fall below default FuSAL, at least in the observed range. A lower weight of $\alpha = 0.75$ results in larger improvements in earlier AL iterations but then falls back to FuSAL and in later AL iterations (not shown here) even below FuSAL. If the time parameter is granted too much influence, as with $\alpha = 0.5$, performance even drops to random selection level. This might also be due to corpus exhaustion. For further experiments, we chose $\alpha = 0.75$ because of its potential to improve upon FuSAL in early iterations.

For BCR with u_{MA}, we change this utility function to $n \cdot u_{\mathrm{MA}}$ to compensate for the normalization by token length which is otherwise already contained in u_{MA}(n is the length of the respective sentence). For u_{LC}, the preparatory experiments already showed that this utility function does not behave on a linear scale. This is so because u_{LC} is based on $P_\theta(\vec{y}|\vec{x})$ for confidence estimation of the complete label sequence $\vec{y}$. Hence, a u_{LC} score twice as high does not indicate doubled benefit for classifier training. Thus, we need a non-linear calibration function to transform u_{LC} into a proper utility estimator on a linear scale so that BCR can be applied.

To determine such a non-linear calibration function, the *true* benefit of an example p would



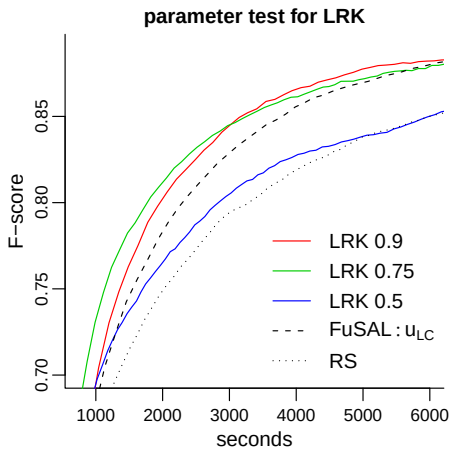


Figure 2: Different parameter settings for CCS and LRK based on FuSAL with u_{LC} as utility function. FuSAL: u_{LC} refers to cost-*in*sensitive FuSAL, CCS and LRK to the cost-sensitive versions of FuSAL with the respective parameters.

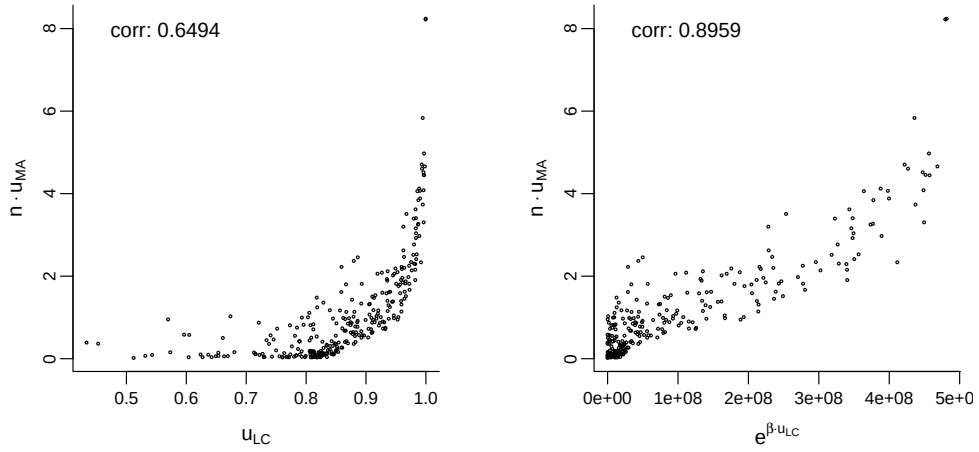


Figure 3: Scatter plots for (a) u_{LC} versus $n \cdot u_{\mathrm{MA}}$ and (b) $e^{\beta \cdot u_{\mathrm{LC}}}$ versus $n \cdot u_{\mathrm{MA}}$

be needed. In the absence of such information, we consider $n \cdot u_{\mathrm{MA}}$ as a good approximation. To identify the relationship between u_{LC} and $n \cdot u_{\mathrm{MA}}$, we trained a model on a random subsample from $P' \subset \mathcal{P}$ and used this model to obtain the scores for u_{LC} and $n \cdot u_{\mathrm{MA}}$ for each example from the test set $\mathcal{T}$.[2] Figure 3 (left) shows a scatter plot of these scores which provides ample evidence that the relationship between u_{LC} and benefit is indeed non-linear. As calibration function for u_{LC} we propose $f(p) = e^{\beta \cdot u_{\mathrm{LC}}(p)}$. Experimentally, we determined $\beta = 20$ as a good value. Figure 3 (right) reveals that $e^{\beta \cdot u_{\mathrm{LC}}(p)}$ is a better utility estimator; the correlation with $n \cdot u_{\mathrm{MA}}$ is now $corr = 0.8959$ and the relationship is close to being linear.

In Figure 4, learning curves for BCR with the utility function u_{LC} and the calibrated function $e^{\beta \cdot u_{\mathrm{LC}}(p)}$ are compared. BCR with the uncalibrated utility function u_{LC} fails miserably (the performance falls even below random selection). This adds credibility to our claim that while u_{LC} may be appropriate for *ranking* examples (as for standard, cost-*in*sensitive AL), it is inappropriate for *estimating* true benefit/utility which is needed when costs are to be incorporated with the BCR method. BCR with the calibrated utility $e^{\beta \cdot u_{\mathrm{LC}}(p)}$, in contrast, outperforms cost-*in*sensitive FuSAL. For further experiments with BCR, we either apply $n \cdot u_{\mathrm{MA}}$ or $e^{\beta \cdot u_{\mathrm{LC}}(p)}$ as utility functions.

[2] We experimented with different sizes for P', with almost identical results.

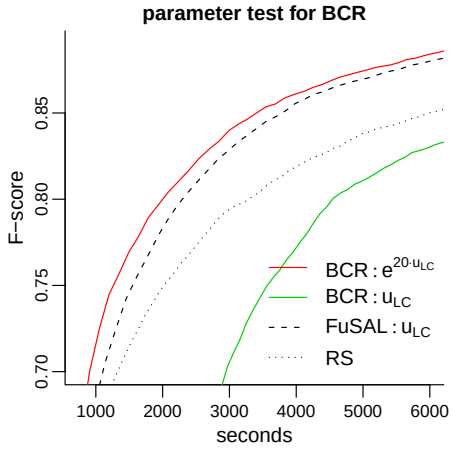


Figure 4: Different parameter settings for BCR

3.2 Comparison of CoSAL Approaches

We compared all three approaches to CoSAL in the parametrization chosen above for the utility functions u_{MA} and u_{LC}. Learning curves are shown in Figure 5. Improvements over cost-*in*sensitive AL are only achieved in early AL iterations up to 2,500s (for CoSAL based on u_{MA}) or 4,000s (for CoSAL based on u_{LC}) of annotation time. This exclusiveness of early improvements can be explained by the size of the corpus and, by this, the limited number of good selection options. Since AL selects with respect to two conflicting criteria, the pool $\mathcal{P}$ should be much larger to increase the chance for examples that are favorable with respect to both criteria.

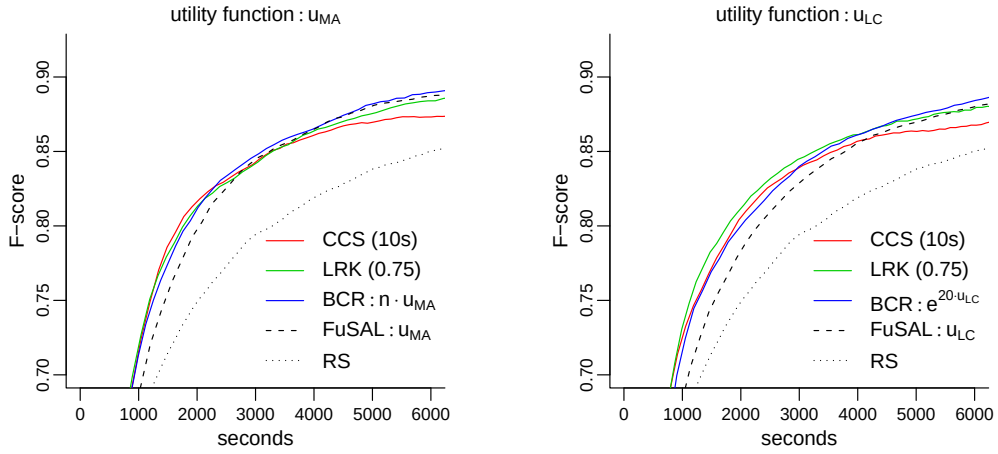


Figure 5: Comparison of CoSAL approaches for the utility functions u_{MA} and u_{LC}. Baseline given by random selection (RS) and standard FuSAL with either u_{MA} or u_{LC}.

Improvements for CoSAL based on u_{LC} are generally higher than for u_{MA}. Moreover, cost-*in*sensitive AL based on u_{LC} does not exhibit any normalization where, in contrast, u_{MA} is normalized at least to the number of tokens per example. In CoSAL, both u_{LC} and u_{MA} are normalized by costs, which is methodologically a more substantial enhancement for u_{LC} than for u_{MA}.

For CoSAL based on u_{MA} we cannot proclaim a clear winner among the different approaches. All three CoSAL approaches improve upon cost-*in*sensitive AL. For CoSAL based on u_{LC}, LRK performs best, while CCS and BCR perform similarly well. Given this result, we might prefer LRK or CCS over BCR. A disadvantage of the first two approaches is that they require corpus-specific parameters which may be difficult to find for a new learning problem for which no data for experimentation is at hand. Though not the best performer, BCR does not require further parametrization and appears more appropriate for real-life annotation projects – as long as utility is an appropriate estimator for benefit. CoSAL with BCR has already been studied by Settles et al. (2008). They also applied a utility function based on sequence-confidence estimation which presumably, as with our u_{LC} utility function, is not a good benefit estimator. The fact that Settles et al. did not explicitly treat this issue might explain why cost-sensitive AL based on BCR often performed worse than cost-*in*sensitive AL in their experiments.

3.3 CoSAL Applied to SeSAL

We looked at a cost-sensitive version of SeSAL by applying the cost-sensitive FuSAL approach together with BCR and the transformation function for the utility as discussed above. On top of this selection, we ran the standard SeSAL approach – only tokens below a confidence threshold were selected for annotation. The following experiments are all based on the u_{LC} utility function (and the transformation function of it).

Figure 6 depicts learning curves for cost-*in*sensitive and cost-sensitive SeSAL and FuSAL which reveal that cost-sensitive SeSAL consid-

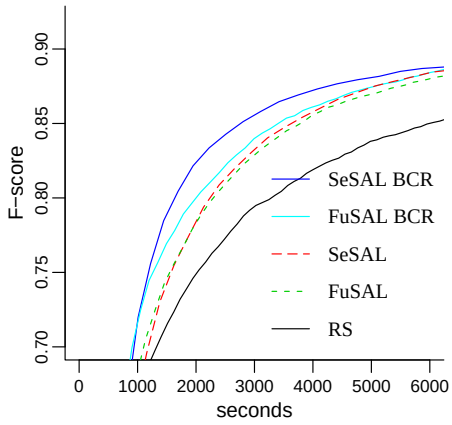


Figure 6: Cost-sensitive (BCR variants) *vs.* cost-*in*sensitive FuSAL and SeSAL with u_{LC} as utility function.

erably outperforms cost-sensitive FuSAL. Cost-sensitive SeSAL attains a target performance of F=0.85 with only 2806s, while cost-sensitive FuSAL needs 3410s, and random selection consumes over 6060s. Thus, cost-sensitive SeSAL here reduces true annotation time by about 54 % compared to random selection, whereas cost-sensitive FuSAL reduces annotation time by only 44 %.

4 Related Work

Although the reduction of data acquisition costs that result from human labeling efforts have always been the main driver for AL studies, *cost-sensitive AL* is a new branch of AL. In an early study on cost metrics for AL, Becker and Osborne (2005) examined whether AL, while decreasing the sample size on the one hand, on the other hand increased annotation efforts. For a real-world AL annotation project, they demonstrated that the actual sampling efficiency measure for an AL approach depends on the cost metric being applied. In a companion paper, Hachey et al. (2005) studied how sentences selected by AL affected the annotators' performance both in terms of the time needed and the annotation accuracy achieved. They found that selectively sampled examples are, on the average, more difficult to annotate than randomly sampled ones. This observation, for the first time, questioned the widespread assumption that all annotation examples can be assigned a uniform cost factor.

Making a standard AL approach cost-sensitive by normalizing utility in terms of annotation time has been proposed before by Haertel et al. (2008), Settles et al. (2008), and Donmez and Carbonell (2008). CoSAL based on the net-benefit (costs subtracted from utility) was proposed by Vijayanarasimhan and Grauman (2009) for object recognition in images and Kapoor et al. (2007) for voice message classification.

5 Conclusions

We investigated three approaches to incorporate the notion of cost into the AL selection mechanism, including a fixed maximal cost budget per example, a linear rank combination to express net-benefit, and a benefit-cost ratio. The cost metric we applied was the *time* needed by human coders for annotating particular annotation examples.

Among the three approaches to cost-sensitive AL, we see a slight advantage for benefit cost ratios in real-world settings because they do not require additional corpus-specific parametrization, once a proper calibration function is found.

Another observation is that advantages of the three cost-sensitive AL models over cost-*in*sensitive ones consistently occur only in early iteration rounds – a result we attribute to corpus exhaustion effects since cost-sensitive AL selects for two criteria (utility and cost) and thus requires a extremely large pool to be able to pick up really advantageous examples. Consequently, applied to real-world annotation settings where the pools may be extremely large, we expect cost-sensitive approaches to be even more effective in terms of the reduction of annotation time.

To be applicable in real-world scenarios, annotation costs which, in our experiments, were directly traceable in the MUC7$_\mathcal{T}$ corpus have to be estimated since they are not known prior to annotation. In Tomanek et al. (2010), we investigated the reading behavior during named entity annotation using eye-tracking technology. With the insights gained from this study on crucial factors influencing annotation time we were able to induce such a much needed *predictive* model of annotation costs. In future work, we plan to incorporate this empirically founded cost model into our approaches to cost-sensitive AL and to investigate whether our positive findings can be reproduced with estimated costs as well.

Acknowledgements

This work was partially funded by the EC within the CALBC (FP7-231727) project.

References

Becker, Markus and Miles Osborne. 2005. A two-stage method for active learning of statistical grammars. In *IJCAI'05 – Proceedings of the 19th International Joint Conference on Artificial Intelligence*, pages 991–996. Edinburgh, Scotland, UK, July 31 - August 5, 2005.

Donmez, Pinar and Jaime Carbonell. 2008. Proactive learning: Cost-sensitive active learning with multiple imperfect oracles. In *CIKM'08 – Proceeding of the 17th ACM conference on Information and Knowledge Management*, pages 619–628. Napa Valley, CA, USA, October 26-30, 2008.

Engelson, Sean and Ido Dagan. 1996. Minimizing manual annotation cost in supervised training from corpora. In *ACL'96 – Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics*, pages 319–326. Santa Cruz, CA, USA, June 24-27, 1996.

Hachey, Ben, Beatrice Alex, and Markus Becker. 2005. Investigating the effects of selective sampling on the annotation task. In *CoNLL'05 – Proceedings of the 9th Conference on Computational Natural Language Learning*, pages 144–151. Ann Arbor, MI, USA, June 29-30, 2005.

Haertel, Robbie, Kevin Seppi, Eric Ringger, and James Carroll. 2008. Return on investment for active learning. In *Proceedings of the NIPS 2008 Workshop on Cost-Sensitive Machine Learning*. Whistler, BC, Canada, December 13, 2008.

Hsu, Frank and Isak Taksa. 2005. Comparing rank and score combination methods for data fusion in information retrieval. *Information Retrieval*, 8(3):449–480.

Kapoor, Ashish, Eric Horvitz, and Sumit Basu. 2007. Selective supervision: Guiding supervised learning with decision-theoretic active learning. In *IJCAI'07 – Proceedings of the 20th International Joint Conference on Artifical Intelligence*, pages 877–882. Hyderabad, India, January 6-12, 2007.

Lafferty, John, Andrew McCallum, and Fernando Pereira. 2001. Conditional Random Fields: Probabilistic models for segmenting and labeling sequence data. In *ICML'01 – Proceedings of the 18th International Conference on Machine Learning*, pages 282–289. Williamstown, MA, USA, June 28 - July 1, 2001.

Settles, Burr and Mark Craven. 2008. An analysis of active learning strategies for sequence labeling tasks. In *EMNLP'08 – Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 1069–1078. Waikiki, Honolulu, Hawaii, USA, October 25-27, 2008.

Settles, Burr, Mark Craven, and Lewis Friedland. 2008. Active learning with real annotation costs. In *Proceedings of the NIPS 2008 Workshop on Cost-Sensitive Machine Learning*. Whistler, BC, Canada, December 13, 2008.

Tomanek, Katrin and Udo Hahn. 2009. Semi-supervised active learning for sequence labeling. In *ACL/IJCNLP'09 – Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing*, pages 1039–1047. Singapore, August 2-7, 2009.

Tomanek, Katrin and Udo Hahn. 2010. Annotation time stamps: Temporal metadata from the linguistic annotation process. In *LREC'10 – Proceedings of the 7th International Conference on Language Resources and Evaluation*. La Valletta, Malta, May 17-23, 2010.

Tomanek, Katrin, Joachim Wermter, and Udo Hahn. 2007. An approach to text corpus construction which cuts annotation costs and maintains corpus reusability of annotated data. In *EMNLP-CoNLL'07 – Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Language Learning*, pages 486–495. Prague, Czech Republic, June 28-30, 2007.

Tomanek, Katrin, Udo Hahn, Steffen Lohmann, and Jürgen Ziegler. 2010. A cognitive cost model of annotations based on eye-tracking data. In *ACL'10 – Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Uppsala, Sweden, July 11-16, 2010.

Tomanek, Katrin. 2010. *Resource-Aware Annotation through Active Learning*. Ph.D. thesis, Technical University of Dortmund.

Vijayanarasimhan, Sudheendra and Kristen Grauman. 2009. What's it going to cost you? predicting effort vs. informativeness for multi-label image annotations. *CVPR'09 – Proceedings of the 2009 IEEE Computer Vision and Pattern Recognition Conference*.

Extraction of Multi-word Expressions from Small Parallel Corpora

Yulia Tsvetkov
Department of Computer Science
University of Haifa
yulia.tsvetkov@gmail.com

Shuly Wintner
Department of Computer Science
University of Haifa
shuly@cs.haifa.ac.il

Abstract

We present a general methodology for extracting multi-word expressions (of various types), along with their translations, from small parallel corpora. We automatically align the parallel corpus and focus on *mis*alignments; these typically indicate expressions in the source language that are translated to the target in a non-compositional way. We then use a large monolingual corpus to rank and filter the results. Evaluation of the quality of the extraction algorithm reveals significant improvements over naïve alignment-based methods. External evaluation shows an improvement in the performance of machine translation that uses the extracted dictionary.

1 Introduction

Multi-word Expressions (MWEs) are lexical items that consist of multiple orthographic words (e.g., *ad hoc*, *by and large*, *New York*, *kick the bucket*). MWEs are numerous and constitute a significant portion of the lexicon of any natural language. They are a heterogeneous class of constructions with diverse sets of characteristics, distinguished by their idiosyncratic behavior. Morphologically, some MWEs allow some of their constituents to freely inflect while restricting (or preventing) the inflection of other constituents. In some cases MWEs may allow constituents to undergo non-standard morphological inflections that they would not undergo in isolation. Syntactically, some MWEs behave like words while other are phrases; some occur in one rigid pattern (and a fixed order), while others permit various syntactic transformations. Semantically, the compositionality of MWEs is gradual, ranging from fully compositional to idiomatic (Bannard et al., 2003).

Because of their prevalence and irregularity, MWEs must be stored in lexicons of natural language processing applications. Handling MWEs correctly is beneficial for a variety of applications, including information retrieval, building ontologies, text alignment, and machine translation.

Identifying MWEs and extracting them from corpora is therefore both important and difficult. In Hebrew (which is the subject of our research), this is even more challenging due to two reasons: the rich and complex morphology of the language; and the dearth of existing language resources, in particular parallel corpora, semantic dictionaries and syntactic parsers.

We propose a novel algorithm for identifying MWEs in bilingual corpora, using automatic word alignment as our main source of information. In contrast to existing approaches, we do not limit the search to one-to-many alignments, and propose an error-mining strategy to detect misalignments in the parallel corpus. We also consult a large monolingual corpus to rank and filter out the expressions. The result is fully automatic extraction of MWEs of various types, lengths and syntactic patterns, along with their translations. We demonstrate the utility of the methodology on Hebrew-English MWEs by incorporating the extracted dictionary into an existing machine translation system.

The main contribution of the paper is thus a new alignment-based algorithm for MWE extraction that focuses on misalignments, augmented by validating statistics computed from a monolingual corpus. After discussing related work, we detail in Section 3 the methodology we propose. Section 4 provides a thorough evaluation of the results. We then extract translations of the identified MWEs and evaluate the contribution of the extracted dictionary in Section 5. We conclude with suggestions for future research.

Coling 2008: Poster Volume, pages 1256–1264,
Beijing, August 2010

2 Related Work

Early approaches to identifying MWEs concentrated on their collocational behavior (Church and Hanks, 1989). Pecina (2008) compares 55 different association measures in ranking German Adj-N and PP-Verb collocation candidates. This work shows that combining different collocation measures using standard statistical classification methods improves over using a single collocation measure. Other results (Chang et al., 2002; Villavicencio et al., 2007) suggest that some collocation measures (especially PMI and Log-likelihood) are superior to others for identifying MWEs. Soon, however, it became clear that mere co-occurrence measurements are not enough to identify MWEs, and their linguistic properties should be exploited as well (Piao et al., 2005). Hybrid methods that combine word statistics with linguistic information exploit morphological, syntactic and semantic idiosyncrasies to extract idiomatic MWEs.

Semantic properties of MWEs can be used to distinguish between compositional and non-compositional (idiomatic) expressions. Katz and Giesbrecht (2006) and Baldwin et al. (2003) use Latent Semantic Analysis for this purpose. They show that compositional MWEs appear in contexts more similar to their constituents than non-compositional MWEs. Van de Cruys and Villada Moirón (2007) use unsupervised learning methods to identify non-compositional MWEs by measuring to what extent their constituents can be substituted by semantically related terms. Such techniques typically require lexical semantic resources that are unavailable for Hebrew.

An alternative approach to using semantics capitalizes on the observation that an expression whose meaning is non-compositional tends to be translated into a foreign language in a way that does not result from a combination of the literal translations of its component words. Alignment-based techniques explore to what extent word alignment in parallel corpora can be used to distinguish between idiomatic expressions and more transparent ones. A significant added value of such works is that MWEs can thus be both identified in the source language and associated with their translations in the target language.

Villada Moirón and Tiedemann (2006) focus on Dutch expressions and their English, Spanish and German translations in the Europarl corpus (Koehn, 2005). To extract the candidates, they use syntactic properties (based on full parsing of the Dutch text) and statistical association measures. This approach requires syntactic resources that are unavailable for Hebrew.

Some recent works concentrate on exploiting translational correspondences of MWEs from (small) parallel corpora. MWE candidates and their translations are extracted as a by-product of automatic word alignment of parallel texts. Unlike Villada Moirón and Tiedemann (2006), who use aligned parallel texts to *rank* MWE candidates, Caseli et al. (2009) actually use them to extract the candidates. After the texts are word-aligned, Caseli et al. (2009) extract sequences of length 2 or more in the source language that are aligned with sequences of length 1 or more in the target. Candidates are then filtered out of this set if they comply with pre-defined part-of-speech patterns, or if they are not sufficiently frequent in the parallel corpus. Even with the most aggressive filtering, precision is below 40% and recall is extremely low (F-score is below 10 for all experiments). Our setup is similar, but we extract MWE candidates from the aligned corpus in a very different way; and we use statistics collected from a *monolingual* corpus to filter and rank the results.

Zarrieß and Kuhn (2009) also use aligned parallel corpora but only focus on one-to-many word alignments. To restrict the set of candidates, they focus on specific syntactic patterns as determined by parsing both sides of the corpus (again, using resources unavailable to us). The results show high precision but very low recall.

3 Methodology

We propose an alternative approach to existing alignment-based techniques for MWE extraction. Using a small bilingual corpus, we extract MWE candidates from noisy word alignments in a novel way. We then use statistics from a large monolingual corpus to rank and filter the list of candidates. Finally, we extract the translation of candidate MWEs from the parallel corpus and use them in a machine translation (MT) system.

3.1 Motivation

Parallel texts are an obvious resource from which to extract MWEs. By definition, idiomatic expressions have a non-compositional meaning, and hence may be translated to a single word (or to an expression with a different meaning) in a foreign language. The underlying assumption of alignment-based approaches to MWE extraction is that MWEs are aligned across languages in a way that differs from compositional expressions; we share this assumption. However, existing approaches focus on the results of word alignment in their quest for MWEs, and in particular consider 1:n and n:m alignments as potential areas in which to look for MWEs. This is problematic for two reasons: first, word alignment algorithms have difficulties aligning MWEs, and hence 1:n and n:m alignments are often noisy; while these environments provide cues for identifying MWEs, they also include much noise. Second, our experimental scenario is such that our parallel corpus is particularly small, and we cannot fully rely on the quality of word alignments, but we have a bilingual dictionary that compensates for this limitation. In contrast to existing approaches, then, we focus on *misalignments*: we trust the quality of 1:1 alignments, which we verify with the dictionary; and we search for MWEs exactly in the areas that word alignment *failed* to properly align, not relying on the alignment in these cases.

Moreover, in contrast to existing alignment-based approaches, we also make use of a large monolingual corpus from which statistics on the distribution of word sequences in Hebrew are drawn. This has several benefits: of course, monolingual corpora are easier to obtain than parallel ones, and hence tend to be larger and provide more accurate statistics. Furthermore, this provides validation of the MWE candidates that are extracted from the parallel corpus: rare expressions that are erroneously produced by the alignment-based technique can thus be eliminated on account of their low frequency in the monolingual corpus.

Specifically, we use pointwise mutual information (PMI) as our association measure. While PMI has been proposed as a good measure for identifying MWEs, it is also known not to discriminate accurately between MWEs and other frequent collocations. This is because it promotes collocations whose constituents rarely occur in isolation (e.g., typos and grammar errors), and expressions consisting of some word that is very frequently followed by another (e.g., *say that*). However, such cases do not have idiomatic meanings, and hence at least one of their constituents is likely to have a 1:1 alignment in the parallel corpus; we only use PMI *after* such alignments have been removed.

An added value of our methodology is the automatic production of an MWE translation dictionary. Since we start with a parallel corpus, we can go back to that corpus after MWEs have been identified, and extract their translations from the parallel sentences in which they occur.

Finally, alignment-based approaches can be symmetric, and ours indeed is. While our main motivation is to extract MWEs in Hebrew, a by-product of our system is the extraction of *English* MWEs, along with their translations to Hebrew. This, again, contributes to the task of enriching our existing bilingual dictionary.

3.2 Resources

Our methodology is in principle language-independent and appropriate for medium-density languages (Varga et al., 2005). We assume the following resources: a small bilingual, sentence-aligned parallel corpus; large monolingual corpora in both languages; morphological processors (analyzers and disambiguation modules) for the two languages; and a bilingual dictionary. Our experimental setup is Hebrew-English. We use a small parallel corpus (Tsvetkov and Wintner, 2010) consisting of 19,626 sentences, mostly from newspapers. The corpus consists of 271,787 English tokens (14,142 types) and 280,508 Hebrew tokens (12,555 types), and is similar in size to that used by Caseli et al. (2009).

We also use data extracted from two monolingual corpora. For Hebrew, we use the morphologically-analyzed MILA corpus (Itai and Wintner, 2008) with part-of-speech tags produced by Bar-Haim et al. (2005). This corpus is much larger, consisting of 46,239,285 tokens (188,572 types). For English we use Google's Web 1T corpus (Brants and Franz, 2006).

Finally, we use a bilingual dictionary consist-

ing of 78,313 translation pairs. Some of the entries were collected manually, while others are produced automatically (Itai and Wintner, 2008; Kirschenbaum and Wintner, 2010).

3.3 Preprocessing the corpora

Automatic word alignment algorithms are noisy, and given a small parallel corpus such as ours, data sparsity is a serious problem. To minimize the parameter space for the alignment algorithm, we attempt to reduce language specific differences by pre-processing the parallel corpus. The importance of this phase should not be underestimated, especially for alignment of two radically different languages such as English and Hebrew (Dejean et al., 2003).

Hebrew,[1] like other Semitic languages, has a rich, complex and highly productive morphology. Information pertaining to gender, number, definiteness, person, and tense is reflected morphologically on base forms of words. In addition, prepositions, conjunctions, articles, possessives, etc., may be concatenated to word forms as prefixes or suffixes. This results in a very large number of possible forms per lexeme. We therefore tokenize the parallel corpus and then remove punctuation. We analyze the Hebrew corpus morphologically and select the most appropriate analysis in context. Adopting this selection, the surface form of each word is reduced to its base form, and bound morphemes (prefixes and suffixes) are split to generate stand-alone "words". We also tokenize and lemmatize the English side of the corpus, using the Natural Language Toolkit package (Bird et al., 2009).

Then, we remove some language-specific differences automatically. We remove frequent function words: in English, the articles *a*, *an* and *the*, the infinitival *to* and the copulas *am*, *is* and *are*; in Hebrew, the accusative marker *at*. These forms do not have direct counterparts in the other language.

For consistency, we pre-process the monolingual corpora in the same way. We then compute the frequencies of all word bi-grams occurring in each of the monolingual corpora.

[1] To facilitate readability we use a transliteration of Hebrew using Roman characters; the letters used, in Hebrew lexicographic order, are *abgdhwzxTiklmns'pcqršt*.

3.4 Identifying MWE candidates

The motivation for our MWE identification algorithm is the assumption that there may be three sources to misalignments (anything that is not a 1:1 word alignment) in parallel texts: either MWEs (which trigger 1:n or n:m alignments); or language-specific differences (e.g., the source language lexically realizes notions that are realized morphologically, syntactically or in some other way in the target language); or noise (e.g., poor translations, low-quality sentence alignment, and inherent limitations of word alignment algorithms).

This motivation induces the following algorithm. Given a parallel, sentence-aligned corpus, it is first pre-processed as described above, to reduce the effect of language-specific differences. We then use Giza++ (Och and Ney, 2003) to word-align the text, employing *union* to merge the alignments in both directions. We look up all 1:1 alignments in the dictionary. If the pair exists in our bilingual dictionary, we remove it from the sentence and replace it with a special symbol, '*'. Such word pairs are not parts of MWEs. If the pair is not in the dictionary, but its alignment score is very high (above 0.5) and it is sufficiently frequent (more than 5 occurrences), we add the pair to the dictionary but also retain it in the sentence. Such pairs are still candidates for being (parts of) MWEs.

Example 1 *Figure 1-a depicts a Hebrew sentence with its word-by-word gloss, and its English translation in the parallel corpus. Here,* bn adm *"person" is a MWE that cannot be translated literally. After pre-processing (Section 3.3), the English is represented as "and i tell her keep away from person" (note that* to *and* the *were deleted). The Hebrew, which is aggressively segmented, is represented as in Figure 1-b. Note how this reduces the level of (morphological and orthographic) difference between the two languages. Consequently, Giza++ finds the alignment depicted in Figure 1-c. Once 1:1 alignments are replaced by '*', the alignment of Figure 1-d is obtained.*

If our resources were perfect, i.e., if word alignment made no errors, the dictionary had perfect coverage and our corpora induced perfect statis-

a. *wamrti* *lh* *lhzhr* *mbn* *adm* *kzh*
and-I-told to-her to-be-careful from-child man like-this
"and I told her to keep away from the person"

b. *w* *ani* *amr* *lh* *lhzhr* *m* *bn* *adm* *k* *zh*
and I tell to-her to-be-careful from child man like this

c. *w* *ani* *amr* *lh* *lhzhr* *m* *bn adm* *k* *zh*
and I told her keep away from person {} {}

d. * * * * *lhzhr* * *bn adm* *k* *zh*
* * * * keep away * person

Figure 1: Example sentence pair (a); after pre-processing (b); after word alignment (c); and after 1:1 alignments are replaced by '*' (d)

tics, then all remaining text (other than the special symbol) in the parallel text would be part of MWEs. In other words, all sequences of remaining source words, separated by '*', are MWE candidates. As our resources are far from perfect, further processing is required in order to prune these candidates. For this, we use association measures computed from the monolingual corpus.

3.5 Ranking and filtering MWE candidates

The algorithm described above produces sequences of Hebrew word forms (free and bound morphemes produced by the pre-processing stage) that are not 1:1-aligned, separated by '*'s. Each such sequence is a MWE candidate. In order to rank the candidates we use statistics from a large *monolingual* corpus. We do *not* rely on the alignments produced by Giza++ in this stage.

We extract all word bi-grams from the remaining candidates. Each bi-gram is associated with its PMI-based score,[2] computed from the monolingual corpus. Interestingly, about 20,000 candidate MWEs are removed in this stage because they do not occur at all in the monolingual corpus.

We then experimentally determine a threshold (see Section 4). A word sequence *of any length* is considered MWE if all the adjacent bi-grams it contains score above the threshold. Finally, we restore the original forms of the Hebrew words in the candidates, combining together bound morphemes that were split during pre-processing; and we restore the function words. Many of the candidate MWEs produced in the previous stage are eliminated now, since they are not genuinely multi-word in the original form.

Example 2 *Refer back to Figure 1-d. The sequence* bn adm k zh *is a MWE candidate. Two bi-grams in this sequence score above the threshold:* bn adm*, which is indeed a MWE, and* k zh*, which is converted to the original form* kzh *and is hence not considered a candidate. We also consider* adm k*, whose score is low. Note that the same aligned sentence can be used to induce the* English *MWE keep away, which is aligned to a single Hebrew word.*

3.6 Results

As an example of the results obtained with this setup, we list in Table 1 the 15 top-ranking extracted MWEs. For each instance we list an indication of the type of MWE: person name (PN), geographical term (GT), noun-noun compound (NNC) or noun-adjective combination (N-ADJ). Of the top 100 candidates, 99 are clearly MWEs,[3] including *mzg awir* (*temper-of air*) "weather", *kmw kn* (*like thus*) "furthermore", *bit spr* (*house-of book*) "school", *šdh t'wph* (*field-of flying*) "airport", *tšwmt lb* (*input-of heart*) "attention", *ai apšr* (*not possible*) "impossible" and *b'l ph*

[2] PMI^k is a heuristic variant of the PMI measure, proposed and studied by Daille (1994), where k, the exponent, is a frequency-related factor, used to demote collocations with low-frequency constituents. The value of the parameter k can be chosen freely ($k > 0$) in order to tune the properties of the PMI to the needs of specific applications. We conducted experiments with k = 0, 0.1, ... , 3 and found k = 2.7 to give the best results for our application.

[3] This was determined by two annotators.

(*in-on mouth*) "orally". Longer MWEs include *ba lidi biTwi* (*came to-the-hands-of expression*) "was expressed"; *xzr ‘l ‘cmw* (*returned on itself*) "recurred"; *ixd ‘m zat* (*together with it*) "in addition"; and *h‘crt hkllit šl haw”m* (*the general assembly of the UN*) "the UN general assembly".

Hebrew	Gloss	Type
xbr hknst	MP	NNC
tl abib	Tel Aviv	GT
gwš qTip	Gush Katif	NNC-GT
awpir pins	Ophir Pines	PN
hc‘t xwq	Legislation	NNC
axmd Tibi	Ahmad Tibi	PN
zhwh glawn	Zehava Galon	PN
raš hmmšlh	Prime Minister	NNC
abšlwm wiln	Avshalom Vilan	PN
br awn	Bar On	PN
mair šTrit	Meir Shitrit	PN
limwr libnt	Limor Livnat	PN
hiw‘c hmšpTi	Attorney General	N-ADJ
twdh rbh	thanks a lot	N-ADJ
rcw‘t ‘zh	Gaza Strip	NNC-GT

Table 1: Results: extracted MWEs

4 Evaluation

MWEs are notoriously hard to define, and no clear-cut criteria exist to distinguish between MWEs and other frequent collocations. In order to evaluate the utility of our methodology, we conducted three different types of evaluations that we detail below and in Section 5.

First, we use a small annotated corpus of Hebrew noun-noun constructions that was made available to us (Al-Haj and Wintner, 2010). The corpus consists of 463 high-frequency bi-grams of the same syntactic construction; of those, 202 are tagged as MWEs (in this case, noun compounds) and 258 as non-MWEs. This corpus consolidates the annotation of three annotators: only instances on which all three agreed were included. Since it includes both positive and negative instances, this corpus facilitates a robust evaluation of precision and recall. Of the 202 positive examples, only 121 occur in our parallel corpus; of the 258 negative examples, 91 occur in our corpus. We therefore limit the discussion to those 212 examples whose MWE status we can determine, and ignore other results produced by the algorithm we evaluate.

On this corpus, we compare the performance of our algorithm to four baselines: using only PMI to rank the bi-grams in the parallel corpus; using PMI computed from the monolingual corpus to rank the bi-grams in the parallel corpus; and using Giza++ 1:n alignments, ranked by their PMI (with bi-grams statistics computed once from parallel and once from monolingual corpora). ‘MWE’ refers to our algorithm. For each of the above methods, we set the threshold at various points, and count the number of true MWEs above the threshold (true positives) and the number of non-MWEs above the threshold (false positives), as well as the number of MWEs and non-MWEs below the threshold (false positives and true negatives, respectively). From these four figures we compute precision, recall and their harmonic mean, f-score, which we plot against (the number of results above) the threshold in Figure 2. Clearly, the performance of our algorithm is consistently above the baselines.

Second, we evaluate the algorithm on additional datasets. We compiled three small corpora of Hebrew two-word MWEs. The first corpus, **PN**, contains 785 person names (names of Knesset members and journalists), of which 157 occur in the parallel corpus. The second, **Phrases**, consists of 571 entries beginning with the letter *x* from a dictionary of Hebrew phrases (Rosenthal, 2009), and a set of 331 idioms we collected from internet resources. Of those, 154 occur in the corpus. The third set, **NN**, consists of the positive examples in the annotated corpus of noun-noun constructions described above.

Since we do not have negative examples for these sets, we only evaluate recall, using a threshold reflecting 2750 results. For each of these datasets, we report the number of MWEs in the dataset (which also occur in the parallel corpus, of course) our algorithm detected. We compare in Table 2 the recall of our method (MWE) to Giza++ alignments, as above, and list also the upper bound (UB), obtained by taking all above-threshold bi-grams in the corpus.

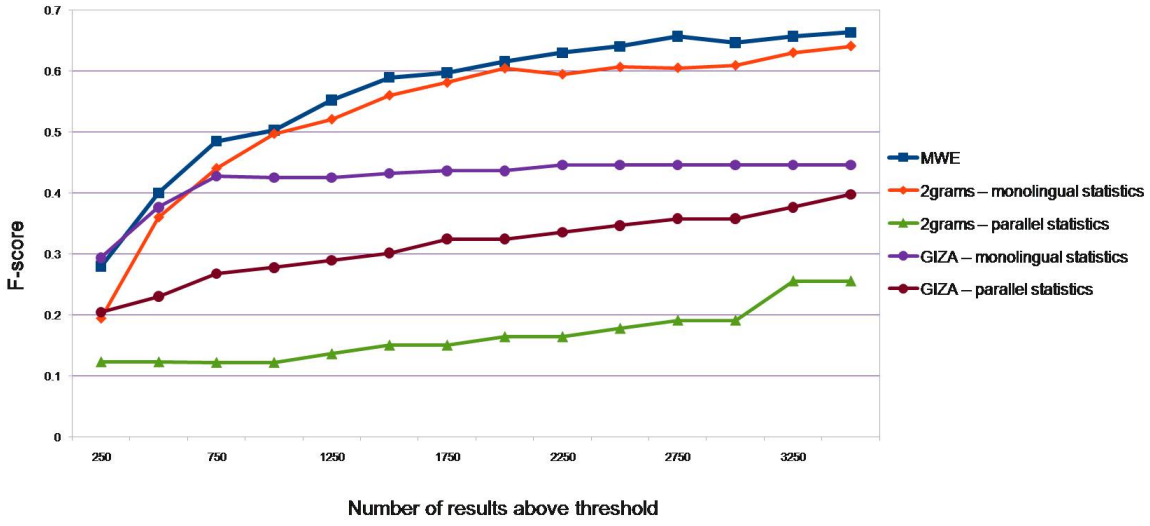


Figure 2: Evaluation results compared with baselines: noun-noun compounds

Method	PN		Phrases		NN	
	#	%	#	%	#	%
UB	74	100	40	100	89	100
MWE	66	89.2	35	87.5	67	75.3
Giza	7	9.5	33	82.5	37	41.6

Table 2: Recall evaluation

5 Extraction of MWE translations

An obvious benefit of using parallel corpora for MWE extraction is that the translations of extracted MWEs are available in the corpus. We use a naïve approach to identify these translations. For each MWE in the source-language sentence, we consider as translation all the words in the target-language sentence (in their original order) that are aligned to the word constituents of the MWE, as long as they form a contiguous string. Since the quality of word alignment, especially in the case of MWEs, is rather low, we remove "translations" that are longer than four words (these are most often wrong). We then associate each extracted MWE in Hebrew with all its possible English translations.

The result is a bilingual dictionary containing 2,955 MWE translation pairs, and also 355 translation pairs produced by taking high-quality 1:1 word alignments (Section 3.4). We used the extracted MWE bilingual dictionary to augment the existing (78,313-entry) dictionary of a transfer-based Hebrew-to-English statistical machine translation system (Lavie et al., 2004b). We report in Table 3 the results of evaluating the performance of the MT system with its original dictionary and with the augmented dictionary. The results show a statistically-significant ($p < 0.1$) improvement in terms of both BLEU (Papineni et al., 2002) and Meteor (Lavie et al., 2004a) scores.

Dictionary	BLEU	Meteor
Original	13.69	33.38
Augmented	13.79	33.99

Table 3: External evaluation

As examples of improved translations, a sentence that was originally translated as "His teachers also hate to the Zionism and besmirch his HRCL and Gurion" (fully capitalized words indicate lexical omissions that are transliterated by the MT system) is translated with the new dictionary as "His teachers also hate to the Zionism and besmirch his Herzl and David Ben-Gurion"; a phrase originally translated as "when so" is now properly translated as "likewise"; and several occurrences of "down spring" and "height of spring" are corrected to "Tel Aviv".

6 Conclusion

We described a methodology for extracting multiword expressions from parallel corpora. The algorithm we propose capitalizes on semantic cues provided by ignoring 1:1 word alignments, and viewing all other material in the parallel sentence as potential MWE. It also emphasizes the importance of properly handling the morphology and orthography of the languages involved, reducing wherever possible the differences between them in order to improve the quality of the alignment. We use statistics computed from a large monolingual corpus to rank and filter the results. We used the algorithm to extract MWEs from a small Hebrew-English corpus, demonstrating the ability of the methodology to accurately extract MWEs of various lengths and syntactic patterns. We also demonstrated that the extracted MWE bilingual dictionary can improve the quality of MT.

This work can be extended in various ways. While several works address the choice of association measure for MWE identification and for distinguishing between MWEs and other frequent collocations, it is not clear which measure would perform best in our unique scenario, where candidates are produced by word (mis)alignment. We intend to explore some of the measures discussed by Pecina (2008) in this context. The algorithm used for extracting the translations of candidate MWEs is obviously naïve, and we intend to explore more sophisticated algorithms for improved performance. Also, as our methodology is completely language-symmetric, it can be used to produce MWE candidates in English. In fact, we already have such a list of candidates, whose quality we will evaluate in the future. Finally, as our main motivation is high-precision, high-recall extraction of Hebrew MWEs, we develop other, non-alignment-based approaches to the task (Al-Haj and Wintner, 2010), and would like to explore the utility of combining different approaches to the same task under a unified framework. We are actively pursuing these research directions.

Acknowledgments

This research was supported by THE ISRAEL SCIENCE FOUNDATION (grants No. 137/06, 1269/07). We are grateful to Hassan Al-Haj for providing the noun compound annotated corpus and to Gennadi Lembersky for his help with the machine translation system.

References

Al-Haj, Hassan and Shuly Wintner. 2010. Identifying multi-word expressions by leveraging morphological and syntactic idiosyncrasy. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING 2010)*, August.

Baldwin, Timothy, Colin Bannard, Takaaki Tanaka, and Dominic Widdows. 2003. An empirical model of multiword expression decomposability. In *Proceedings of the ACL 2003 workshop on Multiword expressions*, pages 89–96. Association for Computational Linguistics.

Bannard, Colin, Timothy Baldwin, and Alex Lascarides. 2003. A statistical approach to the semantics of verb-particles. In Francis Bond, Anna Korhonen, Diana McCarthy and Aline Villavicencio, editors, *Proceedings of the ACL 2003 Workshop on Multiword Expressions: Analysis, Acquisition and Treatment*, pages 65–72.

Bar-Haim, Roy, Khalil Sima'an, and Yoad Winter. 2005. Choosing an optimal architecture for segmentation and POS-tagging of Modern Hebrew. In *Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages*, pages 39–46, Ann Arbor, Michigan, June. Association for Computational Linguistics.

Bird, Steven, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python*. O'Reilly Media, Sebastopol, CA.

Brants, Thorsten and Alex Franz. 2006. Web 1T 5-gram version 1.1. LDC Catalog No. LDC2006T13.

Caseli, Helena, Aline Villavicencio, André Machado, and Maria José Finatto. 2009. Statistically-driven alignment-based multiword expression identification for technical domains. In *Proceedings of the Workshop on Multiword Expressions: Identification, Interpretation, Disambiguation and Applications*, pages 1–8, Singapore, August. Association for Computational Linguistics.

Chang, Baobao, Pernilla Danielsson, and Wolfgang Teubert. 2002. Extraction of translation unit from Chinese-English parallel corpora. In *Proceedings of the first SIGHAN workshop on Chinese language processing*, pages 1–5, Morristown, NJ, USA. Association for Computational Linguistics.

Church, Kenneth. W. and Patrick Hanks. 1989. Word association norms, mutual information and lexicography (rev). *Computational Linguistics*, 19(1):22–29.

Daille, Béatrice. 1994. *Approche mixte pour l'extraction automatique de terminologie : statistiques lexicales et filtres linguistiques*. Ph.D. thesis, Université Paris 7.

Dejean, Herve, Eric Gaussier, Cyril Goutte, and Kenji Yamada. 2003. Reducing parameter space for word alignment. In *Proceedings of the HLT-NAACL 2003 Workshop on Building and using parallel texts*, pages 23–26, Morristown, NJ, USA. Association for Computational Linguistics.

Itai, Alon and Shuly Wintner. 2008. Language resources for Hebrew. *Language Resources and Evaluation*, 42:75–98, March.

Katz, Graham and Eugenie Giesbrecht. 2006. Automatic identification of non-compositional multiword expressions using latent semantic analysis. In *Proceedings of the Workshop on Multiword Expressions: Identifying and Exploiting Underlying Properties*, pages 12–19, Sydney, Australia, July. Association for Computational Linguistics.

Kirschenbaum, Amit and Shuly Wintner. 2010. A general method for creating a bilingual transliteration dictionary. In *Proceedings of The seventh international conference on Language Resources and Evaluation (LREC-2010)*, May.

Koehn, Philipp. 2005. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of the MT Summit X, Phuket, Thailand*.

Lavie, Alon, Kenji Sagae, and Shyamsundar Jayaraman. 2004a. The significance of recall in automatic metrics for mt evaluation. In Frederking, Robert E. and Kathryn Taylor, editors, *AMTA*, volume 3265 of *Lecture Notes in Computer Science*, pages 134–143. Springer.

Lavie, Alon, Shuly Wintner, Yaniv Eytani, Erik Peterson, and Katharina Probst. 2004b. Rapid prototyping of a transfer-based Hebrew-to-English machine translation system. In *Proceedings of TMI-2004: The 10th International Conference on Theoretical and Methodological Issues in Machine Translation*, Baltimore, MD, October.

Och, Franz Josef and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *ACL '02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 311–318, Morristown, NJ, USA. Association for Computational Linguistics.

Pecina, Pavel. 2008. A machine learning approach to multiword expression extraction. In *Proceedings of the LREC Workshop Towards a Shared Task for Multiword Expressions*.

Piao, Scott Songlin, Paul Rayson, Dawn Archer, and Tony McEnery. 2005. Comparing and combining a semantic tagger and a statistical tool for mwe extraction. *Computer Speech and Language*, 19(4):378–397.

Rosenthal, Ruvik. 2009. *Milon HaTserufim (Dictionary of Hebrew Idioms and Phrases)*. Keter, Jerusalem. In Hebrew.

Tsvetkov, Yulia and Shuly Wintner. 2010. Automatic acquisition of parallel corpora from websites with dynamic content. In *Proceedings of The seventh international conference on Language Resources and Evaluation (LREC-2010)*, May.

Van de Cruys, Tim and Begoña Villada Moirón. 2007. Semantics-based multiword expression extraction. In *Proceedings of the Workshop on A Broader Perspective on Multiword Expressions*, pages 25–32, Prague, Czech Republic, June. Association for Computational Linguistics.

Varga, Dániel, Péter Halácsy, András Kornai, Viktor Nagy, László Németh, and Viktor Trón. 2005. Parallel corpora for medium density languages. In *Proceedings of RANLP'2005*, pages 590–596.

Villada Moirón, Begoña and Jörg Tiedemann. 2006. Identifying idiomatic expressions using automatic word alignment. In *Proceedings of the EACL 2006 Workshop on Multi-word-expressions in a multilingual context*. Association for Computational Linguistics.

Villavicencio, Aline, Valia Kordoni, Yi Zhang, Marco Idiart, and Carlos Ramisch. 2007. Validation and evaluation of automatically acquired multiword expressions for grammar engineering. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 1034–1043.

Zarrieß, Sina and Jonas Kuhn. 2009. Exploiting Translational Correspondences for Pattern-Independent MWE Identification. In *Proceedings of the Workshop on Multiword Expressions: Identification, Interpretation, Disambiguation and Applications*, pages 23–30, Singapore, August. Association for Computational Linguistics.

Citation Author Topic Model in Expert Search

Yuancheng Tu, Nikhil Johri, Dan Roth, Julia Hockenmaier
University of Illinois at Urbana-Champaign
{ytu,njohri2,danr,juliahmr}@illinois.edu

Abstract

This paper proposes a novel topic model, Citation-Author-Topic (CAT) model that addresses a semantic search task we define as expert search – given a research area as a query, it returns names of experts in this area. For example, *Michael Collins* would be one of the top names retrieved given the query *Syntactic Parsing*.

Our contribution in this paper is two-fold. First, we model the cited author information together with words and paper authors. Such extra contextual information directly models linkage among authors and enhances the author-topic association, thus produces more coherent author-topic distribution. Second, we provide a preliminary solution to the task of expert search when the learning repository contains exclusively research related documents authored by the experts. When compared with a previous proposed model (Johri et al., 2010), the proposed model produces high quality author topic linkage and achieves over 33% error reduction evaluated by the standard MAP measurement.

1 Introduction

This paper addresses the problem of searching for people with similar interests and expertise, given their field of expertise as the query. Many existing people search engines need people's names to do a "keyword" style search, using a person's name as a query. However, in many situations, such information is insufficient or impossible to know beforehand. Imagine a scenario where the statistics department of a university invited a world-wide known expert in Bayesian statistics and machine learning to give a keynote speech; how can the organizer notify all the people on campus who are interested without spamming those who are not? Our paper proposes a solution to the aforementioned scenario by providing a search engine which goes beyond "keyword" search and can retrieve such information semantically. The organizer would only need to input the research domain of the keynote speaker, i.e. *Bayesian statistics, machine learning*, and all professors and students who are interested in this topic will be retrieved and an email agent will send out the information automatically.

Specifically, we propose a **C**itation-**A**uthor-**T**opic (CAT) model which extracts academic research topics and discovers different research communities by clustering experts with similar interests and expertise. CAT assumes three steps of a hierarchical generative process when producing a document: first, an author is generated, then that author generates topics which ultimately generate the words and cited authors. This model links authors to observed words and cited authors via latent topics and captures the intuition that when writing a paper, authors always first have topics in their mind, based on which, they choose words and cite related works.

Corpus linguists or forensic linguists usually

Coling 2008: Poster Volume, pages 1265–1273,
Beijing, August 2010

identify authorship of disputed texts based on stylistic features, such as vocabulary size, sentence length, word usage that characterize a specific author and the general semantic content is usually ignored (Diederich et al., 2003). On the other hand, graph-based and network based models ignore the content information of documents and only focus on network connectivity (Zhang et al., 2007; Jurczyk and Agichtein, 2007). In contrast, the model we propose in this paper fully utilizes the content words of the documents and combines them with the stylistic flavor contextual information to link authors and documents together to not only identify the authorship, but also to be used in many other applications such as paper reviewer recommendation, research community identification as well as academic social network search.

The novelty of the work presented in this paper lies in the proposal of jointly modeling the cited author information and using a discriminative multinomial distribution to model the co-author information instead of an artificial uniform distribution. In addition, we apply and evaluate our model in a semantic search scenario. While current search engines cannot support interactive and exploratory search effectively, our model supports search that can answer a range of exploratory queries. This is done by semantically linking the interests of authors to the topics of the collection, and ultimately to the distribution of the words in the documents.

In the rest of this paper, we first present some related work on author topic modeling and expert search in Sec. 2. Then our model is described in Sec. 3. Sec. 4 introduces our expert search system and Sec. 5 presents our experiments and the evaluation. We conclude this paper in Sec. 6 with some discussion and several further developments.

2 Related Work

Author topic modeling, originally proposed in (Steyvers et al., 2004; Rosen-Zvi et al., 2004), is an extension of Latent Dirichlet Allocation (LDA) (Blei et al., 2003), a probabilistic generative model that can be used to estimate the properties of multinomial observations via unsupervised learning. LDA represents each document as a mixture of probabilistic topics and each topic as a multinomial distribution over words. The Author topic model adds an author layer over LDA and assumes that the topic proportion of a given document is generated by the chosen author.

Author topic analysis has attracted much attention recently due to its broad applications in machine learning, text mining and information retrieval. For example, it has been used to predict authors for new documents (Steyvers et al., 2004), to recommend paper reviewers (Rosen-Zvi et al., 2004), to model message data (Mccallum et al., 2004), to conduct temporal author topic analysis (Mei and Zhai, 2006), to disambiguate proper names (Song et al., 2007), to search academic social networks (Tang et al., 2008) and to generate meeting status analyses for group decision making (Broniatowski, 2009).

In addition, there are many related works on expert search at the TREC enterprise track from 2005 to 2007, which focus on enterprise scale search and discovering relationships between entities. In that setting, the task is to find the experts, given a web domain, a list of candidate experts and a set of topics [1]. The task defined in our paper is different in the sense that our topics are hidden and our document repositories are more homogeneous since our documents are all research papers authored by the experts. Within this setting, we can explore in depth the influence of the hidden topics and contents to the ranking of our experts. Similar to (Johri et al., 2010), in this paper we apply CAT in a semantic retrieval scenario, where searching people is associated with a set of hidden semantically meaningful topics instead of their personal names.

In recent literature, there are three main lines of work that extend author topic analyses. One line of work is to relax the model's "bag-of-words" assumption by automatically discovering multi-word phrases and adding them into the original model (Johri et al., 2010). Similar work has also been proposed for other topic models such as Ngram topic models (Wallach, 2006; Wang and McCallum, 2005; Wang et al., 2007; Griffiths et al., 2007).

[1] http://trec.nist.gov/pubs.html

Another line of work models authors information as a general contextual information (Mei and Zhai, 2006) or associates documents with network structure analysis (Mei et al., 2008; Serdyukov et al., 2008; Sun et al., 2009). This line of work aims to propose a general framework to deal with collections of texts with an associated networks structure. However, it is based on a different topic model than ours; for example, Mei's works (Mei and Zhai, 2006; Mei et al., 2008) extend probabilistic latent semantic analysis (PLSA), and do not have cited author information explicitly.

Our proposal follows the last line of work which extends author topic modeling with specific contextual information and directly captures the association between authors and topics together with this contextual information (Tang et al., 2008; Mccallum et al., 2004). For example, in (Tang et al., 2008), publication venue is added as one extra piece of contextual information and in (Mccallum et al., 2004), email recipients, which are treated as extra contextual information, are paired with email authors to model an email message corpus. In our proposed method, the extra contextual information consists of the cited authors in each documents. Such contextual information directly captures linkage among authors and cited authors, enhances author-topic associations, and therefore produces more coherent author-topic distributions.

3 The Citation-Author-Topic (CAT) Model

CAT extends previously proposed author topic models by explicitly modelling the cited author information during the generative process. Compared with these models (Rosen-Zvi et al., 2004; Johri et al., 2010), whose plate notation is shown in Fig. 1, CAT (shown in Fig. 2) adds cited author information and generates authors according to the observed author distribution.

Four plates in Fig. 1 represent topic ($\mathcal{T}$), author ($\mathcal{A}$), document ($\mathcal{D}$) and words in each document ($\mathcal{N}_d$) respectively. CAT (Fig. 2) has one more plate, cited-author topic plate, in which each topic is represented as a multinomial distribution over all cited authors (λ_c).

Within CAT, each author is associated with a multinomial distribution over all topics, $\vec{\theta_a}$, and each topic is a multinomial distribution over all words, $\vec{\phi_t}$, as well as a multinomial distribution over all cited authors $\vec{\lambda_c}$. Three symmetric Dirichlet conjugate priors, η, β and γ, are defined for each of these three multinomial distributions in CAT as shown in Fig. 2.

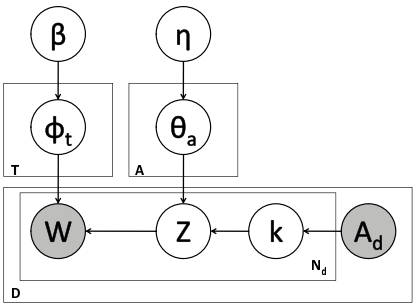


Figure 1: Plate notation of the previously proposed author topic models (Rosen-Zvi et al., 2004; Johri et al., 2010).

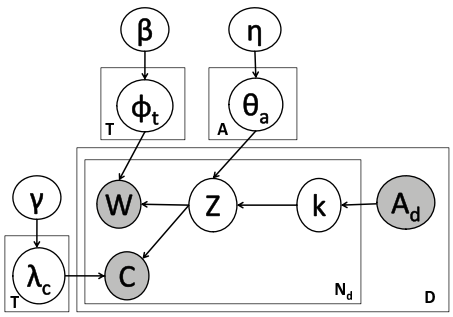


Figure 2: Plate notation of our current model: CAT generates words W and cited authors C independently given the topic.

The generative process of CAT is formally defined in Algorithm 1. The model first samples the word-topic, cited author-topic and the author-topic distributions according to the three Dirichlet hyperparameters. Then for each word in each document, first the author k is drawn from the observed multinomial distribution and that author chooses the topic z_i, based on which word w_i and cited author c_i are generated independently.

CAT differs from previously proposed MAT (Multiword-enhanced Author Topic) model (Johri et al., 2010) in two aspects. First of all, CAT uses

Algorithm 1: CAT: $\mathcal{A}, \mathcal{T}, \mathcal{D}, \mathcal{N}$ are four plates as shown in Fig. 2. The generative process of CAT modeling.

Data: $\mathcal{A}, \mathcal{T}, \mathcal{D}, \mathcal{N}$
for *each topic* $t \in \mathcal{T}$ **do**
  draw a distribution over words:
  $\vec{\phi_t} \sim Dir_{\mathcal{N}}(\beta)$;
  draw a distribution over cited authors:
  $\vec{\lambda_c} \sim Dir_{\mathcal{C}}(\gamma)$;
for *each author* $a \in \mathcal{A}$ **do**
  draw a distribution over topics:
  $\vec{\theta_a} \sim Dir_{\mathcal{T}}(\eta)$;
for *each document* $d \in \mathcal{D}$ *and* k *authors* $\in d$ **do**
  for *each word* $w \in d$ **do**
    choose an author
    $k \sim Multinomial(A_d)$;
    assign a topic i given the author:
    $z_{k,i}|k \sim Multinomial(\theta_a)$;
    draw a word from the chosen topic:
    $w_{d,k,i}|z_{k,i} \sim Multinomial(\phi_{z_{k,i}})$;
    draw a cited author from the topic:
    $c_{d,k,i}|z_{k,i} \sim Multinomial(\lambda_{z_{k,i}})$

cited author information to enhance the model and assumes independence between generating the words and cited authors given the topic. Secondly, instead of an artificial uniform distribution over all authors and co-authors, CAT uses the observed discriminative multinomial distribution to generate authors.

3.1 Parameter Estimation

CAT includes three sets of parameters. The $\mathcal{T}$ topic distribution over words, ϕ_t which is similar to that in LDA. The author-topic distribution θ_a as well as the cited author-topic distribution λ_c. Although CAT is a relatively simple model, finding its posterior distribution over these hidden variables is still intractable due to their high dimensionality. Many efficient approximate inference algorithms have been used to solve this problem including Gibbs sampling (Griffiths and Steyvers, 2004; Steyvers and Griffiths, 2007; Griffiths et al., 2007) and mean-field variational methods (Blei et al., 2003). Gibbs sampling is a special case of Markov-Chain Monte Carlo (MCMC) sampling and often yields relatively simple algorithms for approximate inference in high dimensional models.

In our CAT modeling, we use a collapsed Gibbs sampler for our parameter estimation. In this Gibbs sampler, we integrated out the hidden variables θ, ϕ and λ using the Dirichlet delta function (Heinrich, 2009). The Dirichlet delta function with an M dimensional symmetric Dirichlet prior δ is defined as:

$$\Delta_M(\delta) = \frac{\Gamma(\delta^M)}{\Gamma(M\delta)}$$

Based on the independence assumptions defined in Fig. 2, the joint distribution of topics, words and cited authors given all hyperparameters which originally represented by integrals can be transformed into the delta function format and formally derived in Equation 1.

$$\begin{aligned} & P(\vec{z}, \vec{w}, \vec{c}|\beta, \eta, \lambda) \\ = \; & P(\vec{z}|\beta, \eta, \lambda) P(\vec{w}, \vec{c}|\vec{z}, \beta, \eta, \lambda) \\ = \; & P(\vec{z}) P(\vec{w}|\vec{z}) P(\vec{c}|\vec{z}) \\ = \; & \prod_{a=1}^{A} \frac{\Delta(n_A+\eta)}{\Delta(\eta)} \prod_{z=1}^{T} \frac{\Delta(n_{Z_w}+\beta)}{\Delta(\beta)} \prod_{z=1}^{T} \frac{\Delta(n_{Z_c}+\lambda)}{\Delta(\lambda)} \end{aligned} \tag{1}$$

The updating equation from which the Gibbs sampler draws the hidden variable for the current state j, i.e., the conditional probability of drawing the k^{th} author K_j^k, the i^{th} topic Z_j^i, and the c^{th} cited author C_j^c tuple, given all the hyperparameters and all the observed documents and authors, cited authors except the current assignment (the exception is denoted by the symbol $\forall \neg j$), is defined in Equation 2.

$$\begin{aligned} & P(Z_j^i, K_j^k, C_j^c | W_j^w, \forall \neg j, A_d, \beta, \eta, \gamma) \\ \propto \; & \frac{\Delta(n_Z+\beta)}{\Delta(n_{Z,\neg j}+\beta)} \frac{\Delta(n_K+\eta)}{\Delta(n_{K,\neg j}+\eta)} \frac{\Delta(n_C+\gamma)}{\Delta(n_{C,\neg j}+\gamma)} \\ = \; & \frac{n^w_{i,\neg j}+\beta_w}{\sum_{w=1}^{V} n^w_{i,\neg j}+V\beta_w} \frac{n^i_{k,\neg j}+\eta_i}{\sum_{i=1}^{T} n^i_{k,\neg j}+T\eta_i} \frac{n^c_{i,\neg j}+\lambda_c}{\sum_{c=1}^{C} n^c_{i,\neg j}+C\lambda_c} \end{aligned} \tag{2}$$

The parameter sets ϕ and θ, λ can be interpreted as sufficient statistics on the state variables of the Markov Chain due to the Dirichlet conjugate priors we used for the multinomial distributions.

These three sets of parameters are estimated based on Equations 3 , 4 and 5 respectively, in which n_i^w is defined as the number of times the word w is generated by topic i; n_k^i is defined as the number of times that topic i is generated by author k and n_c^i is defined as the number of times that the cited author c is generated by topic i. The vocabulary size is V, the number of topics is T and the cited-author size is C.

$$\phi_{w,i} = \frac{n_i^w + \beta_w}{\sum_{w=1}^{V} n_i^w + V\beta_w} \quad (3)$$

$$\theta_{k,i} = \frac{n_k^i + \eta_i}{\sum_{i=1}^{T} n_k^i + T\eta_i} \quad (4)$$

$$\lambda_{c,i} = \frac{n_i^c + \lambda_c}{\sum_{c=1}^{C} n_i^c + C\lambda_c} \quad (5)$$

The Gibbs sampler used in our experiments is adapted from the Matlab Topic Modeling Toolbox [2].

4 Expert Search

In this section, we describe a preliminary retrieval system that supports *expert search*, which is intended to identify groups of research experts with similar research interests and expertise by inputting only general domain key words. For example, we can retrieve *Michael Collins* via search for *natural language parsing*.

Our setting is different from the standard TREC expert search in that we do not have a pre-defined list of experts and topics, and our documents are all research papers authored by experts. Within this setting, we do not need to identify the status of our experts, i.e., a real expert or a communicator, as in TREC expert search. All of our authors and cited authors are experts and the task amounts to ranking the experts according to different topics given samples of their research papers.

The ranking function of this retrieval model is derived through the CAT parameters. The search aims to link research topics with authors to bypass the proper names of these authors. Our retrieval function ranks the joint probability of the query words (W) and the target author (a), i.e., $P(W, a)$. This probability is marginalized over all topics, and the probability that an author is cited given the topic is used as an extra weight in our ranking function. The intuition is that an author who is cited frequently should be more prominent and ranked higher. Formally, we define the ranking function of our retrieval system in Equation 6. c_a denotes when the author is one of the cited authors in our corpus. CAT assumes that words and authors, and cited authors are conditionally independent given the topic, i.e., $w_i \perp \text{a} \perp c_a$.

$$\begin{aligned} P(W, a) &= \sum_{w_i} \alpha_i \sum_t P(w_i, a|t, c_a)P(t, c_a) \\ &= \sum_{w_i} \alpha_i \sum_t P(w_i|t)P(a|t)P(c_a|t)P(t) \end{aligned} \quad (6)$$

W is the input query, which may contain one or more words. If a multiword is detected within the query, it is added into the query. The final score is the sum of all words in this query weighted by their inverse document frequency α_i.

In our experiments, we chose ten queries which cover several popular research areas in computational linguistics and natural language processing and run the retrieval system based on three models: the original author topic model (Rosen-Zvi et al., 2004), the MAT model (Johri et al., 2010) and the CAT model. In the original author topic model, query words are treated token by token. Both MAT and CAT expand the query terms with multiwords if they are detected inside the original query. For each query, top 10 authors are returned from the system. We manually label the relevance of these 10 authors based on the papers collected in our corpus.

Two standard evaluation metrics are used to measure the retrieving results. First we evaluate the precision at a given cut-off rank, namely precision at rank k with k ranging from 1 to 10. We then calculate the average precision (AP) for each query and the mean average precision (MAP) for

[2] http://psiexp.ss.uci.edu/research/programs_data/

the queries. Unlike precision at k, MAP is sensitive to the ranking and captures recall information since it assumes the precision of the non-retrieved documents to be zero. It is formally defined as the average of precisions computed at the point of each of the relevant documents in the ranked list as shown in Equation 7.

$$AP = \frac{\sum_{r=1}^{n}(Precision(r) \times rel(r))}{\mid relevant\, documents \mid} \quad (7)$$

To evaluate the recall of our system, we collected a pool of authors for six of our queries returned from an academic search engine, ArnetMiner (Tang et al., 2008)[3] as our reference author pool and evaluate our recall based on the number of authors we retrieved from that pool.

5 Experiments and Analysis

In this section, we describe the empirical evaluation of our model qualitatively and quantitatively by applying our model to the expert search we defined in Sec. 4. We compare the retrieving results with two other models: Multiword- enhanced Author Topic (MAT) model (Johri et al., 2010) and the original author topic model (Rosen-Zvi et al., 2004).

5.1 Data set and Pre-processing

We crawled the ACL anthology website and collected papers from ACL, EMNLP and CONLL over a period of seven years. The ACL anthology website explicitly lists each paper together with its title and author information. Therefore, the author information of each paper can be obtained accurately without extracting it from the original paper. However, many author names are not represented consistently. For example, the same author may have his/her middle name listed in some papers, but not in others. We therefore normalized all author names by eliminating middle names from all authors.

Cited authors of each paper are extracted from the reference section and automatically identified by a named entity recognizer tuned for citation extraction (Ratinov and Roth, 2009). Similar to regular authors, all cited authors are also normalized with their first name initial and their full last name. We extracted about 20,000 cited authors from our corpus. However, for the sake of efficiency, we only keep those cited authors whose occurrence frequency in our corpus is above a certain threshold. We experimented with thresholds of 5, 10 and 20 and retained the total number of 2,996, 1,771 and 956 cited authors respectively.

Conf.	Year	Paper	Author	uni.	Vocab.
ACL	03-09	1,326	2,084	34,012	205,260
EMNLP	93-09	912	1,453	40,785	219,496
CONLL	97-09	495	833	27,312	123,176
Total	93-09	2,733	2,911	62,958	366,565

Table 1: Statistics about our data set. *Uni.* denotes unigram words and *Vocab.* denotes all unigrams and multiword phrases discovered in the data set.

We applied the same strategy to extract multiwords from our corpus and added them into our vocabulary to implement the model described in (Johri et al., 2010). Some basic statistics about our data set are summarized in Table 1 [4].

5.2 Qualitative Coherence Analysis

As shown by other previous works (Wallach, 2006; Griffiths et al., 2007; Johri et al., 2010), our model also demonstrates that embedding multiword tokens into the model can achieve more cohesive and better interpretable topics. We list the top 10 words from two topics of CAT and compare them with those from the unigram model in Table 2. Unigram topics contain more general words which can occur in every topic and are usually less discriminative among topics.

Our experiments also show that CAT achieves better retrieval quality by modeling cited authors jointly with authors and words. The rank of an author is boosted if that author is cited more frequently. We present in Table 3 the ranking of one of our ten query terms to demostrate the high quality of our proposed model. When compared to the model without cited author information, CAT not only retrieves more comprehensive expert list, its ranking is also more reasonable than the model without cited author information.

Another observation in our experiments is that

[3] http://www.arnetminer.org

[4] Download the data and the software package at: http://L2R.cs.uiuc.edu/∼cogcomp/software.php.

Query term: **parsing**				
	Proposed CAT Model		Model without cited authors	
Rank	**Author**	**Prob.**	**Author**	**Prob.**
1	J._Nivre	0.125229	J._Nivre	0.033200
2	C._Manning	0.111252	R._Barzilay	0.023863
3	M._Johnson	0.101342	M._Johnson	0.023781
4	J._Eisner	0.063528	D._Klein	0.018937
5	M._Collins	0.047347	R._McDonald	0.017353
6	G._Satta	0.042081	L._Marquez	0.016003
7	R._McDonald	0.041372	A._Moschitti	0.015781
8	D._Klein	0.041149	N._Smith	0.014792
9	K._Toutanova	0.024946	C._Manning	0.014040
10	E._Charniak	0.020843	K._Sagae	0.013384

Table 3: Ranking for the query term: *parsing*. CAT achieves more comprehensive and reasonable rank list than the model without cited author information.

CAT	**Uni. AT Model**
TOPIC 49	Topic 27
pronoun_resolution	anaphor
antecedent	antecedents
coreference_resolution	anaphoricity
network	anphoric
resolution	is
anaphor	anaphora
pronouns	soon
anaphor_antecedent	determination
semantic_knowledge	pronominal
proper_names	salience
TOPIC 14	Topic 95
translation_quality	hypernym
translation_systems	seeds
source_sentence	taxonomy
word_alignments	facts
paraphrases	hyponym
decoder	walk
parallel_corpora	hypernyms
translation_system	page
parallel_corpus	logs
translation_models	extractions

Table 2: CAT with embedded multiword components achieves more interpretable topics compared with the unigram Author Topic (AT) model.

some experts who published many papers, but on heterogeneous topics, may not be ranked at the very top by models without cited author information. However, with cited author information, those authors are ranked higher. Intuitively this makes sense since many of these authors are also the most cited ones.

5.3 Quantitative retrieval results

One annotator labeled the relevance of the retrieval results from our expert search system. The annotator was also given all the paper titles of each corresponding retrieved author to help make this binary judgment. We experiment with ten queries and retrieve the top ten authors for each query.

Precision@K		
K	CAT Model	Model w/o Cited Authors
1	0.80	0.80
2	0.80	0.70
3	0.73	0.60
4	0.70	0.50
5	0.68	0.48
6	0.70	0.47
7	0.69	0.40
8	0.68	0.45
9	0.73	0.44
10	0.70	0.44

Table 4: Precision at K evaluation of our proposed model and the model without cited author information.

We first used the precision at k for evaluation. We calculate the precision at k for both our proposed CAT model and the MAT model, which does not have the cited author information. The results are listed in Table 4. It can be observed that at every rank position, our CAT model works better. In order to focus more on relevant retrieval results, we also calculated the mean average precision (MAP) for both models. For the given ten queries, the MAP score for the CAT model is 0.78, while the MAT model without cited author information has a MAP score of 0.67. The CAT model with cited author information achieves about 33% error reduction in this experiment.

Query ID	Query Term
1	parsing
2	machine translation
3	dependency parsing
4	transliteration
5	semantic role labeling
6	coreference resolution
7	language model
8	Unsupervised Learning
9	Supervised Learning
10	Hidden Markov Model

Table 5: Queries and their corresponding ids we used in our experiments.

Recall for each query		
Query ID	CAT Model	Model w/o Cite
1	0.53	0.20
2	0.13	0.20
3	0.27	0.13
4	0.13	0.2
5	0.27	0.20
6	0.13	0.26
Average	0.24	0.20

Table 6: Recall comparison between our proposed model and the model without cited author information.

Since we do not have a gold standard experts pool for our queries, to evaluate recall, we collected a pool of authors returned from an academic search engine, ArnetMiner (Tang et al., 2008) as our reference author pool and evaluated our recall based on the number of authors we retrieved from that pool. Specifically, we get the top 15 returned persons from that website for each query and treat them as the whole set of relevant experts for that query and our preliminary recall results are shown in Table 6.

In most cases, the CAT recall is better than that of the compared model, and the average recall is better as well. All the queries we used in our experiments are listed in Table 5. And the average recall value is based on six of the queries which have at least one overlap author with those in our reference recall pool.

6 Conclusion and Further Development

This paper proposed a novel author topic model, CAT, which extends the existing author topic model with additional cited author information.

We applied it to the domain of expert retrieval and demonstrated the effectiveness of our model in improving coherence in topic clustering and author topic association. The proposed model also provides an effective solution to the problem of *community mining* as shown by the promising retrieval results derived in our expert search system.

One immediate improvement would result from extending our corpus. For example, we can apply our model to the ACL ARC corpus (Bird et al., 2008) to check the model's robustness and enhance the ranking by learning from more data. We can also apply our model to data sets with rich linkage structure, such as the TREC benchmark data set or ACL Anthology Network (Radev et al., 2009) and try to enhance our model with the appropriate network analysis.

Acknowledgments

The authors would like to thank Lev Ratinov for his help with the use of the NER package and the three anonymous reviewers for their helpful comments and suggestions. The research in this paper was supported by the Multimodal Information Access & Synthesis Center at UIUC, part of CCICADA, a DHS Science and Technology Center of Excellence.

References

Bird, S., R. Dale, B. Dorr, B. Gibson, M. Joseph, M. Kan, D. Lee, B Powley, D. Radev, and Y. Tan. 2008. The acl anthology reference corpus: A reference dataset for bibliographic research in computational linguistics. In *Proceedings of LREC'08*.

Blei, D., A. Ng, and M. Jordan. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research*.

Broniatowski, D. 2009. Generating status hierarchies from meeting transcripts using the author-topic model. In *In Proceedings of the Workshop: Applications for Topic Models: Text and Beyond*.

Diederich, J., J. Kindermann, E. Leopold, and G. Paass. 2003. Authorship attribution with support vector machines. *Applied Intelligence*, 19:109–123.

Griffiths, T. and M. Steyvers. 2004. Finding scientific topic. In *Proceedings of the National Academy of Science*.

Griffiths, T., M. Steyvers, and J. Tenenbaum. 2007. Topics in semantic representation. *Psychological Review*.

Heinrich, G. 2009. Parameter estimation for text analysis. Technical report, Fraunhofer IGD.

Johri, N., D. Roth, and Y. Tu. 2010. Experts' retrieval with multiword-enhanced author topic model. In *Proceedings of NAACL-10 Semantic Search Workshop*.

Jurczyk, P. and E. Agichtein. 2007. Discovering authorities in question answer communities by using link analysis. In *Proceedings of CIKM'07*.

Mccallum, A., A. Corrada-emmanuel, and X. Wang. 2004. The author-recipient-topic model for topic and role discovery in social networks: Experiments with enron and academic email. Technical report, University of Massachusetts Amherst.

Mei, Q. and C. Zhai. 2006. A mixture model for contextual text mining. In *Proceedings of KDD-2006*, pages 649–655.

Mei, Q., D. Cai, D. Zhang, and C. Zhai. 2008. Topic modeling with network regularization. In *Proceeding of WWW-08:*, pages 101–110.

Radev, D., M. Joseph, B. Gibson, and P. Muthukrishnan. 2009. A Bibliometric and Network Analysis of the field of Computational Linguistics. *Journal of the American Society for Information Science and Technology*.

Ratinov, L. and D. Roth. 2009. Design challenges and misconceptions in named entity recognition. In *Proc. of the Annual Conference on Computational Natural Language Learning (CoNLL)*.

Rosen-Zvi, M., T. Griffiths, M. Steyvers, and P. Smyth. 2004. the author-topic model for authors and documents. In *Proceedings of UAI*.

Serdyukov, P., H. Rode, and D. Hiemstra. 2008. Modeling multi-step relevance propagation for expert finding. In *Proceedings of CIKM'08*.

Song, Y., J. Huang, and I. Councill. 2007. Efficient topic-based unsupervised name disambiguation. In *Proceedings of JCDL-2007*, pages 342–351.

Steyvers, M. and T. Griffiths. 2007. Probabilistic topic models. In *Handbook of Latent Semantic Analysis*. Lawrence Erlbaum Associates.

Steyvers, M., P. Smyth, and T. Griffiths. 2004. Probabilistic author-topic models for information discovery. In *Proceedings of KDD*.

Sun, Y., J. Han, J. Gao, and Y. Yu. 2009. itopicmodel: Information network-integrated topic modeling. In *Proceedings of ICDM-2009*.

Tang, J., J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su. 2008. Arnetminer: Extraction and mining of academic social networks. In *Proceedings of KDD-2008*, pages 990–998.

Wallach, H. 2006. Topic modeling; beyond bag of words. In *International Conference on Machine Learning*.

Wang, X. and A. McCallum. 2005. A note on topical n-grams. Technical report, University of Massachusetts.

Wang, X., A. McCallum, and X. Wei. 2007. Topical n-grams: Phrase and topic discoery with an application to information retrieval. In *Proceedings of ICDM*.

Zhang, J., M. Ackerman, and L. Adamic. 2007. Expertise networks in online communities: Structure and algorithms. In *Proceedings of WWW 2007*.

A Multi-Domain Web-Based Algorithm for POS Tagging of Unknown Words

Shulamit Umansky-Pesin
Institute of computer science
The Hebrew University
pesin@cs.huji.ac.il

Roi Reichart
ICNC
The Hebrew University
roiri@cs.huji.ac.il

Ari Rappoport
Institute of computer science
The Hebrew University
arir@cs.huji.ac.il

Abstract

We present a web-based algorithm for the task of POS tagging of *unknown words* (words appearing only a small number of times in the training data of a supervised POS tagger). When a sentence s containing an unknown word u is to be tagged by a trained POS tagger, our algorithm collects from the web contexts that are partially similar to the context of u in s, which are then used to compute new tag assignment probabilities for u. Our algorithm enables fast multi-domain unknown word tagging, since, unlike previous work, it does not require a corpus from the new domain. We integrate our algorithm into the MXPOST POS tagger (Ratnaparkhi, 1996) and experiment with three languages (English, German and Chinese) in seven in-domain and domain adaptation scenarios. Our algorithm provides an error reduction of up to 15.63% (English), 18.09% (German) and 13.57% (Chinese) over the original tagger.

1 Introduction

Part-of-speech (POS) tagging is a fundamental NLP task that has attracted much research in the last decades. While supervised POS taggers have achieved high accuracy (e.g., (Toutanova et al., 2003) report a 97.24% accuracy in the WSJ Penn Treebank), tagger performance on words appearing a small number of times in their training corpus (*unknown words*) is substantially lower. This effect is especially pronounced in the *domain adaptation* scenario, where the training and test corpora are from different domains. For example, when training the MXPOST POS tagger (Ratnaparkhi, 1996) on sections 2-21 of the WSJ Penn Treebank it achieves 97.04% overall accuracy when tested on WSJ section 24, and 88.81% overall accuracy when tested on the BNC corpus, which contains texts from various genres. For unknown words (test corpus words appearing 8 times or less in the training corpus), accuracy drops to 89.45% and 70.25% respectively.

In this paper we propose an unknown word POS tagging algorithm based on web queries. When a new sentence s containing an unknown word u is to be tagged by a trained POS tagger, our algorithm collects from the web contexts that are partially similar to the context of u in s. The collected contexts are used to compute new tag assignment probabilities for u.

Our algorithm is particularly suitable for *multi-domain* tagging, since it requires no information about the domain from which the sentence to be tagged is drawn. It does not need domain specific corpora or external dictionaries, and it requires no preprocessing step. The information required for tagging an unknown word is very quickly collected from the web.

This behavior is unlike previous works for the task (e.g (Blitzer et al., 2006)), which require a time consuming preprocessing step and a corpus collected from the target domain. When the target domain is heterogeneous (as is the web itself), a corpus representing it is very hard to assemble. To the best of our knowledge, ours is the first paper to provide such an *on-the-fly* unknown word tagging algorithm.

To demonstrate the power of our algorithm as a

Coling 2008: Poster Volume, pages 1274–1282,
Beijing, August 2010

fast multi-domain learner, we experiment in three languages (English, German and Chinese) and several domains. We implemented the MXPOST tagger and integrated it with our algorithm. We show error reduction in unknown word tagging of up to 15.63% (English), 18.09% (German) and 13.57% (Chinese) over MXPOST. The run time overhead is less than 0.5 seconds per an unknown word in the English and German experiments, and less than a second per unknown word in the Chinese experiments.

Section 2 reviews previous work on unknown word Tagging. Section 3 describes our web-query based algorithm. Section 4 and Section 5 describe experimental setup and results.

2 Previous Work

Most supervised POS tagging works address the issue of unknown words. While the general methods of POS tagging vary from study to study – Maximum Entropy (Ratnaparkhi, 1996), conditional random fields (Lafferty et al., 2001), perceptron (Collins, 2002), Bidirectional Dependency Network (Toutanova et al., 2003) – the treatment of unknown words is more homogeneous and is generally based on additional features used in the tagging of the unknown word.

Brants (2000) used only suffix features. Ratnaparkhi (1996) used orthographical data such as suffixes, prefixes, capital first letters and hyphens, combined with a local context of the word. In this paper we show that we improve upon this method. Toutanova and Manning (2000), Toutanova et al. (2003), Lafferty et al. (2001) and Vadas and Curran (2005) used additional language-specific morphological or syntactic features. Huihsin et al. (2005) combined orthographical and morphological features with external dictionaries. Nakagawa and Matsumoto (2006) used global and local information by considering interactions between POS tags of unknown words with the same lexical form.

Unknown word tagging has also been explored in the context of domain adaptation of POS taggers. In this context two directions were explored: a supervised method that requires a manually annotated corpus from the target domain (Daume III, 2007), and a semi-supervised method that uses an unlabeled corpus from the target domain (Blitzer et al., 2006).

Both methods require the preparation of a corpus of target domain sentences and re-training the learning algorithm. Blitzer et al. (2006) used 100K unlabeled sentences from the WSJ (source) domain as well as 200K unlabeled sentences from the biological (target) domain. Daume III (2007) used an 11K words labeled corpus from the target domain.

There are two serious problems with these approaches. First, it is not always realistically possible to prepare a corpus representing the target domain, for example when that domain is the web (e.g., when the POS tagger serves an application working on web text). Second, preparing a corpus is time consuming, especially when it needs to be manually annotated. Our algorithm requires no corpus from the target data domain, no preprocessing step, and it doesn't even need to know the identity of the target domain. Consequently, the problem we address here is more difficult (and arguably more useful) than that addressed in previous work[1].

The domain adaptation techniques above have not been applied to languages other than English, while our algorithm is shown to perform well in seven scenarios in three languages.

Qiu et al. (2008) explored Chinese unknown word POS tagging using internal component and contextual features. Their work is not directly comparable to ours since they did not test a domain adaptation scenario, and used substantially different corpora and evaluation measures in their experiments.

Numerous works utilized web resources for NLP tasks. Most of them collected corpora using data mining techniques and used them offline. For example, Keller et al., (2002) and Keller and Lapata (2003) described a method to obtain frequencies for unseen adjective-noun, noun-noun and verb-object bigrams from the web by query-

[1]We did follow their experimental procedure as much as we could. Like (Blitzer et al., 2006), we compare our algorithm to the performance of the MXPOST tagger trained on sections 2-21 of WSJ. Like both papers, we experimented in domain adaptation from WSJ to a biological domain. We used the freely available Genia corpus, while they used data from the Penn BioIE project (PennBioIE, 2005).

ing a Web engine.

On-line usage of web queries is less frequent and was used mainly in semantic acquisition applications: the discovery of semantic verb relations (Chklovski and Pantel, 2004), the acquisition of entailment relations (Szpektor et al., 2004), and the discovery of concept-specific relationships (Davidov et al., 2007). Chen et al. (2007) used web queries to suggest spelling corrections.

Our work is related to self-training (McClosky et al., 2006a; Reichart and Rappoport, 2007) as the algorithm used its own tagging of the sentences collected from the web in order to produce a better final tagging. Unlike most self-training works, our algorithm is not re-trained using the collected data but utilizes it at test time. Moreover, unlike in these works, in this work the data is collected from the web and is used only during unknown words tagging. Interestingly, previous works did not succeed in improving POS tagging performance using self-training (Clark et al., 2003).

3 The Algorithm

Our algorithm utilizes the correlation between the POS of a word and the contexts in which the word appears. When tackling an unknown word, the algorithm searches the web to find contexts similar to the one in which the word appears in the sentence. A new tag assignment is then computed for the unknown word based on the extracted contexts as well as the original ones.

We start with a description of the web-based context searching algorithm. We then describe how we combine the context information collected by our algorithm with the statistics of the MXPOST tagger. While in this paper we implemented this tagger and used it in our experiments, the context information collected by our web-query based algorithm can be integrated into any POS tagger.

3.1 Web-Query Based Context Collection

An unknown word usually appears in a given sentence with other words on its left and on its right. We use three types of contexts. The first includes all of these neighboring words, the second includes the words on the left, and the third includes the words on the right.

For each context type we define a web query using two common features supported by the major search engines: wild-card search, expressed using the '*' character, and exact sentence search, expressed by quoted characters. The retrieved sentences contain the parts enclosed in quotes in the exact same place they appear in the query, while an asterisk can be replaced by any single word.

For a word u we execute the following three queries for each of its test contexts:

1. **Replacement**: "$u_{-2}u_{-1} * u_{+1}u_{+2}$". This retrieves words that appear in the same context as u.

2. **Left-side**: "$*\ *\ u\ u_{+1}\ u_{+2}$". This retrieves alternative left-side contexts for the word u and its original right-side context.

3. **Right-side**: query "$u_{-2}\ u_{-1}\ u\ *\ *$". This retrieves alternative right-side contexts for u and its original left-side context.

Query Type	Query	Matches (Counts)
Replacement	"irradiation and * treatment of"	heat (15) chemical (7) the (6) radiation (1) pressure (1)
Left-side	"* * H2O2 treatment of"	by an (9) indicated that (5) enhanced by (4) familiar with (3) observed after (3)
Right-side	"irradiation and H2O2 * *"	in comparison (3) on Fe (1) treatment by (1) cause an (1) does not (1)

Table 1: Top 5 matches of each query type for the word 'H2O2' in the GENIA sentence: "UV irradiation and H2O2 treatment of T lymphocytes induce protein tyrosine phosphorylation and Ca2+ signals similar to those observed following biological stimulation.". For each query the matched words (matches) are ranked by the number of times they occur in the query results (counts).

An example is given in Table 1, presenting the top 5 matches of every query type for the word 'H2O2', which does not appear in the English WSJ corpus, in a sentence taken from the English Genia corpus. Since matching words can appear

multiple times in the results, the algorithm maintains for each match a counter denoting the number of times it appeared in the results, and sorts the results according to this number.

Seeing the table, readers might think of the following algorithm: take the leading match in the Replacement query, and tag the unknown word using its most frequent tag (assuming it is a known word). We have experimented with this method, and it turned out that its results are worse than those given by MXPOST, which we use as a baseline.

The web queries are executed by Yahoo! BOSS[2], and the resulting XML containing up to a 1000 results (a limit set by BOSS) is processed for matches. A list of matches is extracted from the *abstract* and *title* nodes of the web results along with counts of the number of times they appear. The matches are filtered to include only known words (words that appear in the training data of the POS tagger more than a threshold) and to exclude the original word or context.

Our algorithm uses a positive integer parameter N_{web}: only the N_{web} top-scoring unique results of each query type are used for tagging. If a left-side or right-side query returns less than N_{web} results, the algorithm performs a 'reduced' query: "$* \ * \ u \ u_{+1}$" for left-side and "$u_{-1} \ u \ * \ *$" for the right side. These queries should produce more results than the original ones due to the reduced context. If these reduced queries do not produce N_{web} results, the web query algorithm is not used to assist the tagger for the unknown word u at hand. If a replacement query does not produce at least N_{web} unique results, only the left-side and right-side queries are used.

For Chinese queries, search engines do their own word segmentation so the semantics of the '*' operator is supposedly the same as for English and German. However, the answer returned by the search engine does not provide this segmentation. To obtain the words filling the '*' slots in our queries, we take all possible segmentations in which the two words appears in our training data.

The queries we use in our algorithm are not the only possible ones. For example, a possible query we do not use for the word u is "$**u_{-1}uu_{+1}u_{+2}$". The aforementioned set of queries gave the best results in our English, German and Chinese development data and is therefore the one we used.

[2]http://developer.yahoo.com/search/boss/

3.2 Final Tagging

The MXPOST Tagger. We integrated our algorithm into the maximum entropy tagger of (Ratnaparkhi, 1996). The tagger uses a set h of contexts ('history') for each word w_i (the index i is used to allow an easy notation of the previous and next words, whose lexemes and POS tags are used as features). For each such word, the tagger computes the following conditional probability for the tag t_r:

$$p(t_r|h) = \frac{p(h,t_r)}{\sum_{t'_r \in T} p(h,t'_r)} \quad (1)$$

where T is the tag set, and the denominator is simply $p(h)$. The joint probability of a history h and a tag t is defined by:

$$p(h,t) = Z \prod_{j=1}^{k} \alpha_j^{f_j(h,t)} \quad (2)$$

where $\alpha_1, \ldots, \alpha_k$ are the model parameters, $f_1, \ldots, f_k$ are the model's binary features (indicator functions), and Z is a normalization term for ensuring that $p(h,t)$ is a probability.

In the training phase the algorithm performs maximum likelihood estimation for the α parameters. These parameters are then used when the model tags a new sentence (the test phase). For words that appear 5 times or less in the training data, the tagger extracts special features based on the morphological properties of the word.

Combining Models. In general, we use the same equation as MXPOST to compute joint probabilities, and our training phase is identical to its training phase. What we change are two things. First, we add new contexts to the 'history' of a word when it is considered as unknown (so Equation (2) is computed using different histories). Second, we use a different equation for computing the conditional probability (below).

When the algorithm encounters an unknown word w_i in the context h during tagging, it performs the web queries defined in Section 3.1. For

each of the N_{web} top resulting matches for each query, $\{h'_n | n \in [1, N_{web}]\}$, the algorithm creates its corresponding history representation h_n. Converting h'_n to h_n is required since in MXPOST a history consists of an ordered set of words together with their POS tags, while h'_n is an ordered set of words without POS tags. Consequently, we define h_n to consist of the same ordered set of words as h'_n, and we tag each word using its most frequent POS tag in the training corpus. If w_{i-1} or w_{i-2} are unknown words, we do not tag them, letting MXPOST use its back-off technique for such a case (which is simply to compute the features that it can and ignore those it cannot).

For each possible tag $t \in T$, its final assignment probability to w_i is computed as an average between its probability given the various contexts:

$$\tilde{p}(t_r|h) = \frac{p_{org}(t_r|h) + \sum_{n=1}^{QN_{web}} p_n(t_r|h_n)}{QN_{web} + 1} \quad (3)$$

where Q is the number of query types used (1, 2 or 3, see Section 3.1).

During inference, we use the two search space constraints applied by the original MXPOST. First, we apply a beam search procedure that considers the 10 most probable different tag sequences of the tagged sentence at any point in the tagging process. Second, known words are constrained to be annotated only by tags with which they appear in the training corpus.

4 Experimental Setup

Languages and Datasets. We experimented with three languages, English, German and Chinese, in various combinations of training and testing domains (see Table 2). For English we used the Penn Treebank WSJ corpus (WSJ) (Marcus et al., 1993) from the economics newspapers domain, the GENIA corpus version 3.02p (GENIA) (Kim et al., 2003) from the biological domain and the British National Corpus version 3 (BNC) (Burnard, 2000) consisting of various genres. For German we used two different corpora from the newspapers domain: NEGRA (Brants, 1997) and TIGER (Brants et al., 2002). For Chinese we used the Penn Chinese Treebank corpus version 5.0 (CTB) (Xue et al., 2002).

All corpora except of WSJ were split using random sampling. For the NEGRA and TIGER corpora we used the Stuttgart-Tuebingen Tagset (STTS).

According to the annotation policy of the GENIA corpus, only the names of journals, authors, research institutes, and initials of patients are annotated by the 'NNP' (Proper Name) tag. Other proper names such as general people names, technical terms (e.g. 'Epstein-Barr virus') genes, proteins, etc. are tagged by other noun tags ('NN' or 'NNS'). This is in contrast to the WSJ corpus, in which every proper name is tagged by the 'NNP' tag. We therefore omitted cases where 'NNP' is replaced by another noun tag from the accuracy computation of the GENIA domain adaptation scenario (see analysis in (Lease and Charniak, 2005)).

In all experimental setups except of WSJ-BNC the training and test corpora are tagged with the same POS tag set. In order to evaluate the WSJ-BNC setup, we converted the BNC tagset to the Penn Treebank tagset using the comparison table provided in (Manning and Schuetze, 1999) (pages 141–142).

Baseline. As a baseline we implemented the MXPOST tagger. An executable code for MXPOST written by its author is available on the internet, but we needed to re-implement it in order to integrate our technique. We made sure that our implementation does not degrade results by running it on our WSJ scenario (see Table 2), which is very close to the scenario reported in (Ratnaparkhi, 1996). The accuracy of our implementation is 97.04%, a bit better than the numbers reported in (Ratnaparkhi, 1996) for a WSJ scenario using different sections.

Parameter Tuning. We ran experiments with three values of the unknown word threshold T: 0 (only words that do not appear in the training data are considered unknown), 5 and 8. That is, the algorithm performs the web context queries and utilizes the tag probabilities of equation 3 for words that appear up to 0 ,5 or 8 times in the training data.

Our algorithm has one free parameter N_{web}, the number of query results for each context type used

Language	Expe. name	Training	Development	Test
English	WSJ	sections 2-21 (WSJ)	section 22 (WSJ)	section 23 (WSJ) (2.4%,6.7%,8.4%)
English	WSJ-BNC	sections 2-21 (WSJ)	2000 BNC sentence	2000 BNC sentences (8.4%,14.9%,17%)
English	WSJ-GENIA	WSJ sections 2-21	2000 GENIA sentences	2000 GENIA sentences (22.7%,30.65%,32.9%)
German	NEGRA	15689 NEGRA sentences	1746 NEGRA sentences	2096 NEGRA sentences (11.1%,24.7%,28.7%)
German	NEGRA-TIGER	15689 NEGRA sentences	2000 TIGER sentences	2000 TIGER sentences (16%,27.3%,30.6%)
German	TIGER-NEGRA	15689 TIGER sentences	1746 NEGRA sentences	2096 NEGRA sentence (16.2%,27.9%,31.6%)
Chinese	CTB	14903 CTB sentences	1924 CTB sentences	1945 CTB senteces (7.4%,15.7%,18.1%)

Table 2: Details of the experimental setups. In the 'Test' column the numbers in parentheses are the fraction of the test corpus words that are considered unknown, when the unknown word threshold is set to 0, 5 and 8 respectively.

	$T=0$			$T=5$			$T=8$		
	WSJ	WSJ-BNC	WSJ-GENIA	WSJ	WSJ-BNC	WSJ-GENIA	WSJ	WSJ-BNC	WSJ-GENIA
Baseline	83.56	61.22	80.05	88.79	68.71	80.12	89.45	70.25	80.8
Unlimited (-)	**84.85**	63.51	82.50	**89.86**	71.12	82.51	**90.47**	72.77	83.16
Top 5 (-)	84.25	64.24	82.75	89.73	71.21	82.78	90.36	72.74	83.46
Top 10 (-)	84.42	64.10	**83.17**	89.70	71.36	**83.00**	90.29	72.87	**83.70**
Top 10 (+)	84.67	**64.47**	82.60	89.83	**72.12**	82.54	90.29	**73.53**	83.22
best imp.	1.19 7.23%	3.25 8.38%	3.12 15.63%	1.07 9.54%	3.41 10.89%	2.88 14.48%	1.02 9.66%	3.28 11.02%	2.9 15.1%

	$T=0$			$T=5$			$T=8$		
	NEGRA	NEGRA-TIGER	TIGER-NEGRA	NEGRA	NEGRA-TIGER	TIGER-NEGRA	NEGRA	NEGRA-TIGER	TIGER-NEGRA
Baseline	90.26	85.71	87.18	91.06	87.88	87.86	91.45	88.22	88.18
Unlimited (-)	91.22	86.60	89.49	91.66	88.22	**89.84**	92.25	89.08	90.23
Top 5 (-)	**91.41**	86.68	89.32	**91.95**	**89.01**	89.72	**92.38**	89.33	90.26
Top 10 (-)	91.06	86.83	**89.50**	91.25	88.36	**89.84**	92.33	**89.38**	**90.26**
Top 10 (+)	90.58	**86.86**	89.43	91.25	88.36	**89.84**	91.53	88.35	89.71
best imp.	1.15 11.8%	1.15 8.04%	2.32 18.09%	0.89 9.95%	1.13 9.32%	1.98 16.3%	0.93 10.87%	1.16 9.84%	2.08 17.59%

	CTB		
	$T=0$	$T=5$	$T=8$
Baseline	74.99	78.03	79.81
Unlimited (-)	77.01	80.46	81.94
Top 5 (-)	**77.58**	**80.75**	82.19
Top 10 (-)	77.43	80.68	82.45
Top 10 (+)	77.43	80.68	**82.35**
best imp.	2.59 10.35%	2.72 12.28%	2.74 13.57%

Table 3: Accuracy of unknown word tagging in the English (top table), German (middle table) and Chinese (bottom table) experiments. Results are presented for three values of the unknown word threshold parameter T: 0, 5 and 8. For all setups our models improves over the MXPOST baseline of (Ratnaparkhi, 1996). The bottom line of each table ('best imp.') presents the improvement (top number) and error reduction (bottom number) of the best performing model over the baseline. The best improvement is in domain adaptation scenarios.

in the probability computation of equation 3. For each setup (Table 2) we ran several combinations of query types and values of N_{web}. We report results for the four leading combinations:

- $N_{web} = 5$, left-side and right-side queries (Top 5 (-)).
- $N_{web} = 10$, left-side and right-side queries (Top 10 (-)).
- $N_{web} = 10$, replacement, left-side and right-side queries (Top 10 (+)).
- $N_{web} =$ Unlimited (in practice, this means 1000, the maximum number of results provided by Yahoo! Boss), left-side and right-side queries (Unlimited (-)).

The order of the models with respect to their performance was identical for the development and test data. That is, the best parameter/queries combination for each scenario can be selected using the development data. We experimented with other parameter/queries combinations and additional query types but got worse results.

5 Results

The results of the experiments are shown in Table 3. Our algorithm improves the accuracy of the MXPOST tagger for all three languages and for all values of the unknown word parameter.

Our experimental scenarios consist of three in-domain setups in which the model is trained and tested on the same corpus (the WSJ, NEGRA and CTB experiments), and four domain adaptation setups: WSJ-GENIA, WSJ-BNC, TIGER-NEGRA and NEGRA-TIGER.

Table 3 shows that our model is relatively more effective in the domain adaptation scenarios. While in the in-domain setups the error reduction values are 7.23% – 9.66% (English), 9.95% – 11.8% (German) and 10.35% – 13.57% (Chinese), in the domain adaptation scenarios they are 8.38% – 11.02% (WSJ-BNC), 14.48% – 15.63% (WSJ-GENIA), 8.04% – 9.84% (NEGRA-TIGER) and 16.3% – 18.09% (TIGER-NEGRA).

Run Time. As opposed to previous approaches to unknown word tagging (Blitzer et al., 2006; Daume III, 2007), our algorithm does not contain a step in which the base tagger is re-trained with a corpus collected from the target domain. Instead, when an unknown word is tackled at test time, a set of web queries is run. This is an advantage for flexible multi-domain POS tagging because preprocessing times are minimized, but might cause an issue of overhead per test word.

To show that the run time overhead created by our algorithm is small, we measured its time performance (using an Intel Xeon 3.06GHz, 3GB RAM computer). The average time it took the best configuration of our algorithm to process an unknown word and the resulting total addition to the run time of the base tagger are given in Table 4. The average time added to an unknown word tagging is less than half a second for English, even less for German, and less than a second for Chinese. This is acceptable for interactive applications that need to examine a given sentence without being provided with any knowledge about its domain.

Error Analysis. In what follows we try to analyze the cases in which our algorithm is most effective and the cases where further work is still required. Due to space limitations we focus only on the (Top 10 (+), $T = 5$) parameters setting, and report the patterns for one English setup. The corresponding patterns of the other parameter settings, languages and setups are similar.

We report the errors of the base tagger that our algorithm most usually fixes and the errors that our algorithm fails to fix. We describe the base tagger errors of the type 'POS tag 'a' is replaced with POS tag 'b' (denoted by: a -> b)' using the following data: (1) total number of unknown words whose correct tag is 'a' that were assigned 'b' by the base tagger; (2) the percentage of unknown words whose correct tag is 'a' that were assigned 'b' by the base tagger; (3) the percentage of unknown words whose correct tag is 'a' that were assigned 'b' by our algorithm; (4) the percentage of mistakes of type (1) that were corrected by our algorithm.

In the English WSJ-BNC setup, the base tagger mistakes that our algorithm handles well (according to the percentage of corrected mistakes) are: (1) NNS -> VBZ (23, 3.73%, 0.8%, 65.2%); (2) CD -> JJ (19 ,13.2% ,9.7% ,37.5%); (3) NN ->

	WSJ	WSJ-BNC	WSJ-GENIA	NEGRA	NEGRA-TIGER	TIGER-NEGRA	CTB
Total addition	00:28:26	00:31:53	1:37:32	00:57:03	00:19:10	00:36:54	2:29:13
Avg. time per word	0.42	0.32	0.33	0.36	0.11	0.21	0.95

Table 4: The processing time added by the web based algorithm to the base tagger. For each setup results are presented for the best performing model and for the unknown word threshold of 8. Results for the other models and threshold parameters are very similar. The top line presents the total time added in the tagging of the full test data (hours:minutes:seconds). The bottom line presents the average processing time of an unknown word by the web based algorithm (in seconds).

JJ (97, 6.17%, 5.3%, 27.8%); (4) JJ -> NN (69, 9.73%, 7.76%, 33.3%). The errors that were not handled well by our algorithm are: (1) IN -> JJ (70, 46.36% , 41%, 8.57%); (2) VBP -> NN (25, 19.5%, 21.9% , 0%).

In this setup, 'CD' is a cardinal number, 'IN' is a preposition, 'JJ' is an adjective, 'NN' is a noun (singular or mass), 'NNS' is a plural noun, 'VBP' is a verb in non-third person singular present tense and 'VBZ' is a verb in third person, singular present tense.

We can see that no single factor is responsible for the improvement over the baseline. Rather, it is due to correcting many errors of different types. The same general behavior is exhibited in the other setups for all languages.

Multiple Unknown Words. Our method is capable of handling sentences containing several unknown words. Query results in which '*' is replaced by an unknown word are filtered. For queries in which an unknown word appears as part of the query (when it is one of the two right or left non-'*' words), we let MXPOST invoke its own unknown word heuristics if needed[3].

In fact, the relative improvement of our algorithm over the baseline is *better* for adjacent unknown words than for single words. For example, consider a sequence of consecutive unknown words as correctly tagged if all of its words are assigned their correct tag. In the WSJ-GENIA scenario (Top 10 (+), $T = 5$), the error reduction for sequences of length 1 (unknown words surrounded by known words, 8767 sequences) is 8.26%, while for 2-words (2620 sequences) and 3-words (614 sequences) it is 11.26% and 19.11% respectively. Similarly, for TIGER-NEGRA (same parameters setting) the error reduction is 6.85%, 8.07% and 18.18% for sequences of length 1 (4819) ,2 (1126) and 3 (223) respectively.

[3]They are needed only if the word is on the left of the word to be tagged.

6 Conclusions and Future Work

We presented a web-based algorithm for POS tagging of unknown words. When an unknown word is tackled at test time, our algorithm collects web contexts of this word that are then used to improve the tag probability computations of the POS tagger.

In our experiments we used our algorithm to enhance the unknown word tagging quality of the MXPOST tagger (Ratnaparkhi, 1996), a leading state-of-the-art tagger, which we implemented for this purpose. We showed significant improvement (error reduction of up to 18.09%) for three languages (English, German and Chinese) in seven experimental setups. Our algorithm is especially effective in domain-adaptation scenarios where the training and test data are from different domains.

Our algorithm is fast (requires less than a second for processing an unknown word) and can handle test sentences coming from any desired unknown domain without the costs involved in collecting domain-specific corpora and retraining the tagger. These properties makes it particularly appropriate for applications that work on the web, which is highly heterogeneous.

In future work we intend to integrate our algorithm with additional POS taggers, experiment with additional corpora and domains, and improve our context extraction mechanism so that our algorithm will be able to fix more error types.

References

Blitzer, John, Ryan McDonald, and Fernando Pereira, 2006. Domain adaptation with structural correspon-

dence learning. *EMNLP '06*.

Brants, Sabine, Stefanie Dipper, Silvia Hansen, Wolfgang Lezius and George Smith, 2002. The TIGER Treebank. *Proceedings of the Workshop on Treebanks and Linguistic Theories*.

Brants, Thorsten, 1997. The NEGRA Export Format. *CLAUS Report, Saarland University*.

Brants, Thorsten, 2000. Tnt: a statistical part-of-speech tagger. In *The Sixth Conference on Applied Natural Language Processing*.

Burnard, Lou, 2000. *The British National Corpus User Reference Guide*. Technical Report, Oxford University.

Chen, Qing, Mu Li, and Ming Zhou. 2007. Improving query spelling correction using web search results. In *EMNLP-CoNLL '07*.

Chklovski, Timothy and Patrick Pantel. 2004. Verbocean: Mining the web for fine-grained semantic verb relations. *EMNLP '04*.

Clark, Stephen, James Curran and Miles Osborne. 2003. Bootstrapping POS-taggers using unlabeled data. *CoNLL '03*.

Collins, Michael. 2002. Discriminative training methods for hidden markov models: theory and experiments with perceptron algorithms. *EMNLP '02*.

Daume III, Hal. 2007. Frustratingly easy domain adaptation. *ACL '07*.

Davidov, Dmitry, Ari Rappoport, and Moshe Koppel. 2007. Fully unsupervised discovery of concept-specific relationships by web mining. *ACL '07*.

Huihsin, Tseng, Daniel Jurafsky, and Christopher Manning. 2005. Morphological features help pos tagging of unknown words across language varieties. *The Fourth SIGHAN Workshop on Chinese Language Processing*.

Jin–Dong Kim, Tomoko Ohta, Yuka Teteisi and Jun'ichi Tsujii, 2003. GENIA corpus – a semantically annotated corpus for bio-textmining. *Bioinformatics*, 19:i180–i182, Oxford University Press, 2003.

Lafferty, John D., Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *The Eighteenth International Conference on Machine Learning*.

Keller, Frank, Mirella Lapata, and Olga Ourioupina. 2002. Using the Web to Overcome Data Sparseness. *EMNLP '02*.

Keller, Frank, Mirella Lapata. 2003. . *Computational Linguistics*, 29(3):459–484.

Lease, Matthew and Eugene Charniak. 2005. Parsing Biomedical Literature. *Proceedings of the Second International Joint Conference on Natural Language Processing.*.

Manning Chris and Hinrich Schuetze. 1999. Foundations of Statistical Natural Language Processing. *MIT Press.*.

Marcus, Mitchell P., Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of english: The penn treebank. *Computational Linguistics*, 19(2):313–330.

McClosky, David, Eugene Charniak, and Mark Johnson, 2006a. Effective self-training for parsing. *HLT-NAACL '06*.

Nakagawa, Tetsuji and Yuji Matsumoto. 2006. Guessing parts-of-speech of unknown words using global information. *ACL-COLING '06* .

PennBioIE. 2005. Mining the Bibliome Project.. *http://bioie.ldc.upenn.edu*.

Qiu, Likun, Changjian Hu and Kai Zhao. 2008. A method for automatic POS guessing of Chinese unknown words. *COLING '08.*

Ratnaparkhi, Adwait. 1996. A maximum entropy model for part-of-speech tagging. *EMNLP '96*.

Reichart, Roi and Ari Rappoport. 2007. Self-Training for Enhancement and Domain Adaptation of Statistical Parsers Trained on Small Datasets. *ACL '07*.

Reynolds, Sheila M. and Jeff A. Bilmes. 2005. Part-of-speech tagging using virtual evidence and negative training. *EMNLP '06*.

Szpektor, Idan, Hristo Tanev, Ido Dagan, and Bonaventura Coppola. 2004. Scaling web-based acquisition of entailment relations. *EMNLP '04*.

Toutanova, Kristina and Christopher D. Manning. 2000. Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. *EMNLP '00*.

Toutanova, Kristina, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. *NAACL '03*.

Vadas, David and James R. Curran. 2005. Tagging unknown words with raw text features. *Australasian Language Technology Workshop 2005*.

Nianwen Xue, Fu-Dong Chiou and Martha Palmer, 2002. Building a large–scale annotated Chinese corpus. *ACL '02*.

Urdu and Hindi: Translation and sharing of linguistic resources

Karthik Visweswariah, Vijil Chenthamarakshan, Nandakishore Kambhatla
IBM Research India
{v-karthik,vijil.e.c,kambhatla}@in.ibm.com

Abstract

Hindi and Urdu share a common phonology, morphology and grammar but are written in different scripts. In addition, the vocabularies have also diverged significantly especially in the written form. In this paper we show that we can get reasonable quality translations (we estimated the Translation Error rate at 18%) between the two languages even in absence of a parallel corpus. Linguistic resources such as treebanks, part of speech tagged data and parallel corpora with English are limited for both these languages. We use the translation system to share linguistic resources between the two languages. We demonstrate improvements on three tasks and show: statistical machine translation from Urdu to English is improved (0.8 in BLEU score) by using a Hindi-English parallel corpus, Hindi part of speech tagging is improved (upto 6% absolute) by using an Urdu part of speech corpus and a Hindi-English word aligner is improved by using a manually word aligned Urdu-English corpus (upto 9% absolute in F-Measure).

1 Introduction

Hindi and Urdu are official languages of India and Urdu is also the national language of Pakistan. Hindi is spoken by around 853 million people and Urdu by around 164 million people (Malik et al., 2008). Although native speakers of Hindi can comprehend most of spoken Urdu and vice versa, these languages have diverged a bit since independence of India and Pakistan – with Hindi deriving a lot of words from Sanskrit and Urdu from Persian. One clear difference between Hindi and Urdu is the script: Hindi is written in a left-to-right Devanagari script while Urdu is written in Nastaliq calligraphy style of the right-to-left Perso-Arabic script. Hence, despite the similarities, it is impossible for an Urdu speaker to read Hindi text and vice versa. The first problem we address is the translation between Hindi and Urdu in the absence of a Hindi-Urdu parallel corpus.

Though these languages together are spoken by around a billion people they are not very rich in linguistic resources. A treebank for Hindi is still under development[1] and part of speech taggers for Hindi and Urdu are trained on very small amounts of data. For translation between Hindi/Urdu and English there are no large corpora, the available corpora are an order of magnitude smaller than those available for European languages or Arabic-English. Given the lack of linguistic resources in each of the languages and the similarities between these languages, we explore whether each language can benefit from resources available in the other language.

1.1 Urdu-Hindi script conversion/translation

Sharing resources between Hindi and Urdu requires us to be able to convert from one written form to the other. Given that the languages share a good fraction of their spoken vocabularies, the obvious approach to convert between the two scripts would be to transliterate between them. While this approach has recently been attempted (Malik et al., 2009), (Malik et al., 2008) there are two main problems with this approach.

Challenges in Hindi-Urdu transliteration: Urdu uses diacritical marks that were taken from the Arabic script which serve various purposes. Urdu has short and long vowels. Short vowels are indicated by placing a diacritic with the con-

[1] https://verbs.colorado.edu/hindi_wiki/index.php/Hindi_Treebank_Data

Coling 2008: Poster Volume, pages 1283–1291,
Beijing, August 2010

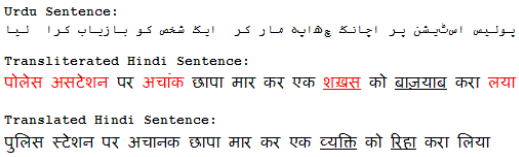


Figure 1: An Urdu sentence transliterated and translated to Hindi

sonant that precedes it in the syllable. The diacritical marks are also used for gemination (doubling of a consonant), which in Hindi is handled using a conjunct form where the consonant is essentially repeated twice. Yet another function of diacritical marks is to mark the absence of a vowel following a base consonant. Though diacritical marks are critical for correct pronunciation and sometimes even for disambiguation of certain words, they are sparingly used in written material intended for native speakers of the language. Missing diacritical marks create substantial difficulties for transliteration systems. Another difficulty is created by the fact that Urdu words cannot have a short vowel at the end of a word, whereas the corresponding Hindi word can sometimes have a short vowel. This cannot be resolved deterministically and results ambiguity in transliteration from Urdu to Hindi. A third issue is the presence of certain sounds (and their corresponding letters) that have no equivalent in Urdu. These letters are approximated in Urdu with phonetic equivalents. Transliteration from Urdu to Hindi suffers in the presence of words with these letters. Recent work on Urdu-Hindi transliteration (Malik et al., 2009) report transliteration word error rates of 16.4% and 23.1% for Urdu sentences with and without diacritical marks respectively. This problem is illustrated in Figure 1. The figure shows an Urdu sentence that is transliterated to Hindi using the Hindi Urdu Machine Transliteration (HUMT) system [2] and translated using our Statistical Machine Translation System. The words which are in red are transliteration errors (mainly because of missing diacritical marks).

Difference in Word Frequency Distributions: Even if we could transliterate perfectly between Urdu and Hindi it might not be desirable to do so from the point of view of human understanding or for machine consumption. This is because word frequencies of shared words would be different in Hindi and Urdu. At the extreme, there are several Urdu words that a fluent Hindi speaker would not understand and vice versa. More commonly, native speakers of Hindi and Urdu would use different words to refer to the same concept, even though both these words are technically correct in either of these languages. In initial experiments to quantify this issue on our corpus, which is mainly from the news domain, we estimated that around 28% of the word tokens in Urdu would not be natural in Hindi. This estimate assumes perfect transliteration, and we estimated the total error rate including transliteration at around 55% for the publicly available HUMT system. In Figure 1, the words that have been underlined have been replaced using a different word by our SMT system, even though the original word might be technically correct. Our preliminary experiments exploring this issue convinced us that to be able to convert from Urdu into natural Hindi (and vice versa) we would need to go beyond transliteration to translation to deal with the divergence of the vocabularies in the written forms of the two languages.

Importance of Context We would like to point out that in addition to word for word fidelity, there are more subtle issues in translating from Urdu-Hindi. One issue is that words in Hindi are drawn from different source languages, and with word to word translations, we might end up with phrases that are unnatural. For example, consider different ways of writing the English phrase *National* and *News* in Hindi. The word *National* in Hindi could possibly be written as *rashtriya*, *kaumi* or *national* which have origins in Sanskrit, Persian/Arabic and English respectively. Similarly the word *News* could be written as *samachar*, *khabaren* or *news* (once again with origins in Sanskrit, Persian/Arabic and English). The natural ways for writing the phrase *national news* are: *rashtriya samachar*, *kaumi khabaren* or *national news*, any of the other six combinations would be quite rare.

Another issue is that corresponding words in Hindi and Urdu might have different genders. An

[2] http://www.puran.info/HUMT/HUMT.aspx

example from (Sinha, 2009) are the words *vajah* (Urdu, feminine) and *karan* (Hindi, masculine), which would mean that the phrase *because of him* would be written as *us ke karan* in Hindi and as *us ki vajah se* in Urdu. We note that the *ke* in Hindi and *ki* in Urdu are different because of the difference in genders of the word following them. This suggests we would need to go beyond word for word translation and would need to use a higher order n-gram language model to translate with fidelity between Hindi and English.

We have established the need for going beyond transliteration, but a key challenge is to achieve good translation accuracy in the absence of a Hindi-Urdu parallel corpus. In Section 3 we describe a multi-pronged approach to translate between Hindi and Urdu in the absence of a parallel corpus that exploits the similarities between the languages.

1.2 Applications: sharing linguistic resources

We next outline the three tasks for which we consider sharing resources between Hindi and Urdu which serve as a test of the efficacy of our systems.

Statistical machine translation

In recent years, there is a lot of interest in Statistical Machine Translation (SMT) Systems (Brown et al., 1993). Modern SMT systems (Koehn et al., 2003; Ittycheriah and Roukos, 2007) learn translation models based on large amounts of parallel data. The quality of an SMT system is dependent on the amount of parallel data on which the system is trained. Unfortunately, for the pairs Urdu-English and Hindi-English, parallel data are not available in large quantities, thereby limiting the quality of these SMT systems. In this paper we show that we can improve the accuracy of an Urdu→English SMT system by using a Hindi-English parallel corpus.

Part of Speech tagging

Part of Speech (POS) tagging involves marking the part of speech of a word based on its definition and surrounding context in a sentence. Sequential modeling techniques like Hidden Markov Models (Rabiner, 1990) and Conditional Random Fields (Lafferty et al., 2001) are commonly used to build Part of Speech taggers. These models are typically trained using a manually tagged part of speech corpus. Manual tagging of data requires lot of human effort and hence large corpora are not readily available for many languages. We improve a Hindi POS tagger by using a manually tagged Urdu POS corpus.

Supervised bitext alignment

Machine generated word alignments between pairs of languages have many applications: building statistical machine translation systems, building dictionaries, projection of syntactic information to resource poor languages (Yarowsky and Ngai, 2001). Most of the early work on generating word alignments has been unsupervised, e.g. IBM Models 1-5 (Brown et al., 1993), recent improvements on the IBM Models (Moore, 2004), and the HMM algorithm described in (Vogel et al., 1996). Recently, significant improvements in performance of aligners have been achieved by the use of human annotated word alignments (Ittycheriah and Roukos, 2007; Lacoste-Julien et al., 2006). We describe a method to transfer manual word alignments from Urdu-English to Hindi-English to improve Hindi-English word alignments.

1.3 Contributions

Our main contributions are summarized below: We present a hybrid technique to translate between Hindi and Urdu in the *absence* of a Hindi-Urdu parallel corpus that significantly improves upon past efforts to convert between Hindi and Urdu via transliteration. We validate the efficacy of the translation systems we present, by using it to share linguistic resources between Hindi and Urdu for three important tasks:

1. We improve a part of speech tagger for Hindi using an Urdu part of speech corpus.

2. We use manual Urdu-English word alignments to improve the task of Hindi-English bitext alignments.

3. We use a Hindi-English parallel corpus to improve translation from Urdu to English.

2 Related work

Converting between the scripts of Hindi and Urdu is non-trivial and has been a recent focus (Malik et al., 2008; Malik et al., 2009). (Malik et al., 2008) uses hand designed rules encoded using finite state transducers to transliterate between Hindi and Urdu. As reported in (Malik et al., 2009) these hand designed rules achieve accuracies of only about 50% in the absence of diacritical marks. (Malik et al., 2009) improves Urdu→Urdu transliteration performance to 79% by post processing the output of the transducer with a statistical language model. In contrast to (Malik et al., 2009) we use a statistical model for character transliteration. As discussed in Section 1.1, due to the divergence of vocabularies in written Hindi and Urdu, transliteration is not sufficient to convert from written Urdu to written Hindi. We also use a more flexible model that allows for more natural translations by allowing Urdu words to translate into Hindi words that do not sound the same.

(Sinha, 2009) builds an English-Urdu machine translation system using an English-Hindi machine translation system and a Hindi-Urdu word mapping table, suitably adjusted for part of speech and gender. Their system is not statistical, and is largely based on manual creation of a large database of Hindi-Urdu correspondences. Additionally, as mentioned in the conclusion, their system cannot be used for direct translation from Hindi to Urdu, since a grammatical analysis of the English provides information necessary for the Hindi to Urdu mapping. In contrast to this work, our techniques are largely statistical, require minimal manual effort and can directly translate between Hindi and Urdu without the associated English.

3 Approach to translating between Hindi and Urdu

As discussed in Section 1, transliteration between Hindi and Urdu is not a straightforward task and current efforts result in fairly high error rates. We would like to combine the approaches of transliteration and translation since our goal is to use the translation for sharing linguistic resources rather than for direct consumption.

We use a fairly standard phrase based translation system to translate between Hindi and Urdu. The key challenge that we overcome is being able to develop such a system with acceptable accuracy in the absence of Hindi-Urdu resources (we have neither a parallel corpus nor a dictionary with sufficient coverage). In spite of the absence of resources, translation between this language pair is made feasible by the fact that word order is largely maintained and translation can be done maintaining a word to word correspondence. There are some exceptions to the monotonicity in the two languages. Consider the English phrase *Government of Sindh* which in Urdu would be *hukumat e sindh* in the same word order as in English, while in Hindi it would be *sindhi sarkar* with the word order flipped (with respect to English and Urdu). This example also shows that sometimes we do not have a word for word translation between Hindi and Urdu, the word *sindhi* in Hindi corresponding to the Urdu words *e sindh*. In spite of these exceptions, Hindi-Urdu translation can largely be done with the monotonicity assumption and with the assumption of word to word correspondences. Thus the central issue in translating between Hindi and Urdu is the creation of a word to word conditional probability table. We explain our technique assuming we are translating from Urdu to Hindi. We take a hybrid approach to creating this table, using three different approaches.

The first approach is the pivot language approach (Wu and Wang, 2007), with English as a pivot language. We get probabilities of a Urdu word u being generated by a Hindi word h, considering intermediate English phrases e as:

$$P_p(u|h) = \sum_e P(u|e)P(e|h)$$

The translation probabilities $P(u|e)$ and $P(e|h)$ are obtained using an Urdu-English and an English-Hindi parallel corpus respectively.

This approach works reasonably well, but suffers from a couple of drawbacks. There are several common Hindi and Urdu words for which the translation is unsatisfactory. This is because the alignments for these words are not precise, they often do not align to any English word, or align to

an English words in combination with other Hindi words. A common example of this is with verbs, consider for example the English sentence
He works
which would translate into Hindi/Urdu as:
vah kaam karta hai
with word alignments *He* ↔ *vah*, *works* ↔ *kaam karta hai* . Automatic aligners often make mistakes on these multi-word alignments, and this create problems for words like *karta* and *hai* which often do not have direct equivalents in English. To deal with this issue we manually build a small phrase table for the most frequent Hindi and Urdu words by a consulting an online Hindi-Urdu-English dictionary (Platts, 1884). We also manually handle the frequent examples we observed of cases where we need to handle differences in tokenization between Hindi and Urdu (e.g *keliye* written as one word in Urdu and as *ke liye* in Hindi).

The other issue with the pivot language approach is that for word pairs which are rare in one of the languages, $\sum_e P(u|e)P(e|h)$ can easily work out to zero. This is exacerbated by alignment errors for rarer words. Thus, to strengthen our phrase table especially for infrequent words, we use a transliteration approach to build a phrase table. Note that for rare words like names of people and places, the words in Hindi and Urdu are transliterations of each other.

In light of the issues in transliterating between Hindi and Urdu (Malik et al., 2008; Malik et al., 2009) we take a statistical approach (Abdul-Jaleel and Larkey, 2003) to building a transliteration based phrase table.

We assume a generative model for producing Urdu words from Hindi words based on a character transliteration probability table P_c. The probability $P_t(u|h)$ of generating a Urdu word u from a Hindi word h is given by:

$$P_t(u|h) = \sum_{\mathbf{a}} \prod_i P_c(u_i|h_{a(i)})P(a_i|a_{i-1}),$$

where $\mathbf{a}$ represents the alignment between the Hindi and Urdu characters, $a(i)$ is the the index of the Hindi character that the i^{th} Urdu character is aligned to, $P_c(u_c|h_c)$ is the probability of an Urdu character u_c being generated by a Hindi character h_c and $P(a_i|a_{i-1})$ represents a distortion probability. Since transliteration is monotonic and we want to encourage small jumps we set: $P(a_i|a_{i-1}) = c\eta^{(a_i - a_{i-1})}$ for $a_i > a_{i-1}$ and 0 otherwise. To obtain P_c we use the EM algorithm and we can reuse standard machinery that is used to obtain HMM word alignments in Statistical Machine Translation (with the constraint of Monotone alignments). To calculate a transliteration based phrase table, for each Hindi word h we search over a large vocabulary of Urdu words and retain words u for which $P_t(u|h)$ is sufficiently high as possible transliterations of h. We set the probabilities in the transliteration based phrase table to be proportional to $P_t(u|h)$. Finding this table requires calculating $P_t(u|h)$ for every pair of words in the Urdu and Hindi vocabulary, we use the Forward-Backward algorithm for efficiency and parallelize the calculations over several machines.

The only remaining issue is how we get training data to train our transliteration model. To obtain such training data we use a table of consonant character conversions between Hindi and Urdu as given in (Malik et al., 2008). We look for words in our pivot language based translation table, where there are at least three consonants and at least 50% of the consonants are shared. We observed that this yields pairs of words that are transliterations of one another with high precision. These word pairs are used as training data to build our character transliteration model P_c.

Final word translation table is obtained by combining our three approaches as follows: If the word is present in our dictionary, we use the translation given in the dictionary and exclude all others, if not we linearly interpolate between the probability table we get based on using English as a pivot language and probability table we get based on transliteration.

4 Experimental results

In this section we report on experiments to evaluate the quality of our translation method described in Section 3 and report on the application of Hindi↔Urdu translation to the sharing of linguistic resources between the two languages.

Algorithm 1 Create Urdu-Hindi Phrase Table

for all u such that u is very frequent Urdu word **do**
  $h \leftarrow$ Hindi word for u from dictionary
  $P_d(u|h) \leftarrow 1$
end for
$U \leftarrow$ Urdu vocabulary
$H \leftarrow$ Hindi vocabulary vocabulary
for all $u \in U, h \in H$ **do**
  $P_p(u|h) \leftarrow \sum_e P(u|e)P(e|h)$ {Create an Urdu-Hindi translation table using English as the pivot}
end for
for all $u \in U, h \in H$ such that $P_p(u|h) > \delta$ and $ConsonantOverlap(u, h) > \Delta$ **do**
  Add (u, h) to training set T
end for
$P_c \leftarrow$
$\arg\max_Q \prod_{(u,h)\in T} \sum_{\mathbf{a}} \prod_i Q(u_i|h_{a_i})P(a_i|a_{i-1})$
{Maximize using EM}
for all $u \in U, h \in H$ **do**
  $P_t(u|h) \leftarrow c \sum_{\mathbf{a}} \prod_i P_c(u_i|h_{a(i)})P(a_i|a_{i-1})$
  {Use Forward-Backward Algorithm}
end for
for all $u \in U, h \in H$ **do**
  if $P_d(u|h) \leftarrow 1$ **then**
    $P_{final}(u|h) \leftarrow 1$
  else
    $P_{final}(u|h) \leftarrow \lambda_p P_p(u|h) + \lambda_t P_t(u|h)$
  end if
end for

4.1 Evaluation of Hindi-Urdu translation

We built a Hindi-Urdu transliteration system as explained in Section 3. For building a pivot language based translation table we used 70k sentences from the NIST MT-08 corpus training corpus for Urdu-English. For Hindi-English we used an internal corpus of 230k sentences. We built our statistical transliteration model on roughly 3k word pairs that we obtained as described in Section 3. For Urdu→Hindi translation, we used a five gram language model built from a crawl of archives from Hindi news web sites (the corpus size was about 60 million words). For Hindi→Urdu translation we use the MT-08 Urdu corpus (about 1.5 million words) to build a trigram LM.

We evaluated the translation system in translating from Urdu to Hindi. We asked an annotator to evaluate 100 sentences (2700 words), by marking an error on a word if it was a wrong translation or unnatural in Hindi. We compared our translation system against the Hindi Urdu Machine Transliteration (HUMT) system[3]. We found an error rate of 18% for our system as against 46% for the HUMT system.

4.2 Word alignments

In this section we describe experiments at improving a Hindi-English word aligner using hand alignments for an Urdu-English corpus. For the Urdu-English corpus we use a manually word aligned corpus of roughly 10k sentences, while for the Hindi-English corpus we had roughly 3k sentences out of which we set aside 300 sentences (5300 words) for a test set. In addition to these (relatively) small supervised corpora we also use a sentence parallel Hindi-English corpus (without manual word alignments) of roughly 250k sentences.

For word alignments we use the Maximum Entropy aligner described in (Ittycheriah and Roukos, 2005) that is trained using hand aligned training data. We first translate the Urdu sentences in the Urdu-English word aligned corpus to Hindi, and then transfer the alignments by simply replacing the alignment links to a Urdu word by links to the corresponding decoded Hindi word. The above procedure covers bulk of the cases since Urdu-Hindi translation is largely a word to word translation. The special case of a phrase of multiple Urdu words decoded to multiple Hindi words is handled as follows: we align each of the words in the Hindi phrase to the union of the sets of English words that each word in the Urdu phrase aligns to. Once we convert the Urdu-English manual alignments to an additional corpus we build two Hindi-English alignment models, one on the original corpus, the other on the (Urdu→Hindi)-English corpus. The MaxEnt aligner (Ittycheriah and Roukos, 2005) models the probability of a

[3] http://www.puran.info/HUMT/HUMT.aspx

nTrain	Hindi data	+ Urdu
5	60.8	69.8
50	64.1	70.5
800	71.4	73.0
2800	75.1	75.7

Table 1: *Word alignment F-Measure as a function of the number of manually aligned Hindi-English sentences used for training. The third column shows improvements obtained by adding 10k Urdu-English word alignments sentences.*

num. words	$f(w_i, t_i), g(t_{i-1}, t_i)$	+ $h(t_i^u, t_i)$
5k	76.5	82.5
10k	81.7	84.7
20k	84.5	86.7
47k	90.6	91.0

Table 2: *POS tagging accuracy as a function of the amount of Hindi POS tagged data used to build the model. The third column indicates the use of the Urdu data via a feature type.*

particular set of links in the alignment L given the source sentence S and the target sentence T as: $P(L|S,T) = \prod_{i=1}^{M} p(l_i|t_1^M, s_1^K, l_1^{i-1})$. Let us denote by P_h and P_u the alignment models trained on the Hindi-English and the (Urdu→Hindi)-English corpora respectively. We combine these models log-linearly to obtain our final model for alignment:

$$P(L|S,T) = P_h^{\alpha}(L|S,T)P_u^{1-\alpha}(L|S,T).$$

To find the most likely alignment we use the same algorithm as in (Ittycheriah and Roukos, 2005) since the structure of the model is unchanged.

We report on the performance (Table 1) of a baseline Hindi-English word aligner built with varying amounts of Hindi-English manually word aligned training data compared against an aligner that combines in a model trained on the 10k (Urdu→Hindi)-English sentences. We observe large gains with small amounts of labelled Hindi-English alignment data, and even when we have 2800 sentences of Hindi-English data we see a gain in performance adding in the Urdu data. We note that the MaxEnt aligner we use (Ittycheriah and Roukos, 2005) defaults to (roughly) doing an HMM alignment using a word translation matrix obtained via unsupervised training. Thus the aligners reported on in Table 1 use a large amount of unsupervised data in addition to the small amounts of labelled data mentioned in the Table.

4.3 POS tagging

Unlike English for which there is an abundance of POS training data for Hindi and Urdu data is quite limited. For our experiments, we use the CRULP corpus (Hussain, 2008) for Urdu and a corpus from IITB (Dalal et al., 2007) for Hindi. The CRULP POS corpus has 150k words and uses a tagset of size 46 to tag the corpus. The IITB corpus has 50k words and uses a tagset of size 26. We set a side a test set of size 5k words from the IITB corpus. For part of speech tagging we use CRFs (Lafferty et al., 2001) with two types of features, $f(t_i, w_i)$ and $g(t_i, t_{i-1})$. With the small amounts of training data we have, adding additional feature templates degraded the performance.

In our POS tagging experiments we consider using the Urdu corpus to help POS tagging in Hindi. We first translate all of the CRULP Urdu data to Hindi. We cannot simply add in this data to the training data because of differences in the tagsets used in the data sets for the two languages. In order to make use of the additional Urdu POS tagged data (translated to Hindi), we build a separate POS tagger on this data, and use predictions from this model as a feature in training the Hindi POS tagger. We use these predictions via a feature template $h(t_i, t_i^u)$ where t_i^u denotes the tag assigned to the ith word by the POS tagger built from the CRULP Urdu data set translated into Hindi.

We present results in Table 2 with varying amounts of Hindi data used for training, in each case we present results with and without use of the Urdu resources. We see a small gain even when we use all of the available Hindi training data and as expected we see larger gains when smaller amounts of Hindi data are used.

We analyzed the type of errors and the error reduction when using the Urdu data for the case where we used only 5k words of Hindi data.

We find that the two frequent error types that were greatly reduced were noun being tagged as main verb (reduction of 65% relative) and main verb tagged as auxiliary verb (reduction of 71%). Reduction in confusion between nouns and main verbs is expected since these are open word classes that can most benefit from additional data. This also causes the reduction in errors of tagging main verbs as auxiliary verbs, since in Hindi, verbs are multi word groups with a main verb followed by one or more auxiliary verbs. Reduction of error rate in most of the other error types were close to the overall error rate reduction.

4.4 Sharing parallel corpora for machine translation

We experimented with using our internal Hindi-English parallel corpus (230k) sentences to obtain better translation for Urdu-English. The Urdu-English corpus we use is the NIST MT-08 training data set (70k sentences). We use the Direct Translation Model 2 (DTM) described in (Ittycheriah and Roukos, 2007) for all our translation experiments.

We build our baseline Urdu→English system using the NIST MT-08 training data. In training our DTM model we use HMM alignments, alignments with the MaxEnt aligner, and hand alignments for 10k sentences (the hand alignments were used to train the MaxEnt aligner).

We translated the Hindi in our Hindi-English corpus to Urdu, creating an additional Urdu-English corpus. We then use a MaxEnt aligner to align the Urdu-English words in this corpus. Since we expect this corpus to be relatively noisy due to incorrect translation from Urdu to Hindi we do not include this corpus while generating HMM alignments. We add the synthetic Urdu-English data with MaxEnt alignments to our baseline data and train a DTM model. Results comparing to the baseline are given Table 3, which shows an improvement of 0.8 in BLEU score over the baseline system by using data from the Hindi-English corpus.

This improvement is not due to unknown words being covered (the vocabulary covered is the same). Also note that in the bridge language approach we cannot get alternative translations for single words that were not already present in the Urdu-English phrase table. Thus, we believe that the improvement is due to longer phrases being seen more often in training. An example improved translation is shown below:

Corpus	MT08 Eval
Urdu	23.1
+Hindi	23.9

Table 3: *Improvement in Urdu-English machine translation using Hindi-English data* .

Ref: *just as long as its there they feel safe*

Baseline: *as long as this they just think there are safe*

Improved: *just as long as they are there they feel safe*

5 Conclusions

In this paper, we showed that we can translate between Hindi and English *without* a parallel corpus and improve upon previous efforts at transliterating between the two languages. We also showed that Hindi-Urdu translation can be useful to the sharing of linguistic resources between the two languages. We believe this approach to sharing linguistic resources will be of immense value especially with resources like treebanks which require a large effort to develop.

Acknowledgments

We thank Salim Roukos and Abe Ittycheriah for discussions that helped guide our efforts.

References

[AbdulJaleel and Larkey2003] AbdulJaleel, Nasreen and Leah S. Larkey. 2003. Statistical transliteration for english-arabic cross language information retrieval. In *CIKM*.

[Brown et al.1993] Brown, Peter F., Vincent J.Della Pietra, Stephen A. Della Pietra, and Robert. L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19:263–311.

[Dalal et al.2007] Dalal, Aniket, Kumara Nagaraj, Uma Sawant, Sandeep Shelke, and Pushpak Bhattacharyya. 2007. Building feature rich pos tagger for morphologically rich languages. In *Proceedings of the Fifth International Conference on Natural Language Processing*, Hyderabad, India, January.

[Hussain2008] Hussain, Sarmad. 2008. Resources for urdu language processing. In *Proceedings of the 6th workshop on Asian Language Resources*.

[Ittycheriah and Roukos2005] Ittycheriah, Abraham and Salim Roukos. 2005. A maximum entropy word aligner for arabic-english machine translation. In *HLT/EMNLP*.

[Ittycheriah and Roukos2007] Ittycheriah, Abraham and Salim Roukos. 2007. Direct translation model 2. In Sidner, Candace L., Tanja Schultz, Matthew Stone, and ChengXiang Zhai, editors, *HLT-NAACL*, pages 57–64. The Association for Computational Linguistics.

[Koehn et al.2003] Koehn, Philipp, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *NAACL '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 48–54, Morristown, NJ, USA. Association for Computational Linguistics.

[Lacoste-Julien et al.2006] Lacoste-Julien, Simon, Benjamin Taskar, Dan Klein, and Michael I. Jordan. 2006. Word alignment via quadratic assignment. In *HLT-NAACL*.

[Lafferty et al.2001] Lafferty, J., A. McCallum, , and F. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *International Conference on Machine Learning*.

[Malik et al.2008] Malik, M. G. Abbas, Christian Boitet, and Pushpak Bhattacharyya. 2008. Hindi urdu machine transliteration using finite-state transducers. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 537–544, Manchester, UK, August. Coling 2008 Organizing Committee.

[Malik et al.2009] Malik, Abbas, Laurent Besacier, Christian Boitet, and Pushpak Bhattacharyya. 2009. A hybrid model for urdu hindi transliteration. In *Proceedings of the 2009 Named Entities Workshop: Shared Task on Transliteration (NEWS 2009)*, pages 177–185, Suntec, Singapore, August. Association for Computational Linguistics.

[Moore2004] Moore, Robert C. 2004. Improving ibm word alignment model 1. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04), Main Volume*, pages 518–525, Barcelona, Spain, July.

[Platts1884] Platts, John T. 1884. *A dictionary of Urdu, classical Hindi and English*. W. H. Allen and Co.

[Rabiner1990] Rabiner, Lawrence R. 1990. A tutorial on hidden markov models and selected applications in speech recognition. pages 267–296.

[Sinha2009] Sinha, R. Mahesh K. 2009. Developing english-urdu machine translation via hindi. In *Third Workshop on Computational Approaches to Arabic-Script-based Languages*.

[Vogel et al.1996] Vogel, Stephan, Hermann Ney, and Christoph Tillmann. 1996. Hmm-based word alignment in statistical translation. In *Proceedings of the 16th conference on Computational linguistics*, pages 836–841, Morristown, NJ, USA. Association for Computational Linguistics.

[Wu and Wang2007] Wu, Hua and Haifeng Wang. 2007. Pivot language approach for phrase-based statistical machine translation. In *ACL*.

[Yarowsky and Ngai2001] Yarowsky, David and Grace Ngai. 2001. Inducing multilingual pos taggers and np bracketers via robust projection across aligned corpora. In *NAACL*.

Phrase Structure Parsing with Dependency Structure

Zhiguo Wang and **Chengqing Zong**
National Laboratory of Pattern Recognition
Institute of Automation, Chinese Academy of Sciences
{zgwang, cqzong}@nlpr.ia.ac.cn

Abstract

In this paper we present a novel phrase structure parsing approach with the help of dependency structure. Different with existing phrase parsers, in our approach the inference procedure is guided by dependency structure, which makes the parsing procedure flexibly. The experimental results show our approach is much more accurate. With the help of golden dependency trees, F1 score of our parser achieves 96.08% on Penn English Treebank and 90.61% on Penn Chinese Treebank. With the help of N-best dependency trees generated by modified MSTParser, F1 score achieves 90.54% for English and 83.93% for Chinese.

1 Introduction

Over the past few years, several high-precision phrase parsers have been presented, and most of them are employing probabilistic context-free grammar (PCFG). As we all know, the basic PCFG has the problems that the independence assumption is too strong and lacks of lexical conditioning (Jurafsky and Martin, 2007). Although researchers have proposed various models and inference algorithms aiming to solve these problems, the performance of existing phrase parsers is still remained to further improve. Most of the existing approaches can be classified into two categories: unlexicalized PCFG based (Johnson, 1998; Klein and Manning, 2003; Levy and Manning, 2003; Matsuzaki et al., 2005; Petrov et al., 2006) and lexicalized PCFG based (Collins, 1999a; Charniak, 1997; Bikel, 2004; Charniak and Johnson, 2005).

Unlexicalized PCFG based approach attempts to weaken the independence assumption by annotating non-terminal symbols with labels of ancestor, siblings and even the latent annotations encoded by local information. In lexicalized PCFG based approach, researchers believe that the forms of a constituent and its sub-constituents are determined more by the constituent's head than any other of its lexical items (Charniak, 1997), so they annotate non-terminal symbols with the head words information.

Both of the two PCFG based approaches have improved the basic PCFG based parsers significantly. However, neither of them has been guided by enough linguistic priori knowledge. Their parsing procedures are too mechanical. Because of this, the efficiency is always worse, and much more artificial ambiguities, which are different from linguistic ambiguities (Krotov et al., 1998; Johnson, 1998), are generated. We believe parsing procedure guided by more linguistic priori knowledge will help to overcome the drawbacks in some extent. From our intuition, dependency structure, another type of syntactic structure with much linguistic knowledge, will be a good candidate to guide phrase parsing procedure.

In this paper we present a novel approach to using dependency structure to guide phrase parsing. This novel approach has its virtues from multiple angles. First, dependency structure offers a good compromise between the conflicting demands of analysis depth, which makes it much easier to get through hand annotating than phrase structure (Nivre, 2004). So, when we want to build a phrase structure corpus, we can build a dependency structure corpus first, and get the corresponding phrase structure automatically with the help of dependency structure. Second, many parsing algorithms with linear-time complexity used in dependency parsers can still achieve the state-of-the-art results (Johansson, 2007), but almost all phrase parsers with high-precision have no efficient algorithms superior to cubic-time complexity. So, in order to get an efficient

Coling 2008: Poster Volume, pages 1292–1300,
Beijing, August 2010

parser, we can first get a dependency structure through linear-time algorithm, and then obtain the phrase structure with the help of dependency structure more efficiently. Third, the lexicalized PCFG based parsers which just bring the head words into account have got a highly improved performance. It gives us reasons to believe dependency structure which takes the relationship of all the words will bring phrase parser a great help.

Remainder of this paper is organized as follows: Section 2 introduces the related work. Section 3 gives a consistency between dependency structure and phrase structure, and presents an approach to parsing phrase structure with dependency structure. In Section 4, we discuss the experiments and analysis. Finally, we conclude this paper and point out some future work in Section 5.

2 Related work

Unlexicalized PCFG based parsers (Johnson, 1998; Klein and Manning, 2003; Levy and Manning, 2003; Matsuzaki et al., 2005; Petrov et al., 2006) are the most successful parsing tools. They regard parsing as a pure machine learning question. However, they haven't taken any extra linguistic priori knowledge directly into account. Lexicalized PCFG based parsers (Collins, 1999a; Charniak, 1997; Bikel, 2004; Charniak and Johnson, 2005) just bring a little linguistic priori knowledge (head word information) into learning phase. In inference phase, both of the unlexicalized PCFG based approach and lexicalized PCFG based approach are using the pure searching algorithms, which try to parse a sentence monotonously, either from left to right or from right to left. From these states, we can find that manners of current parsers are too mechanical. Because of this, the efficiency of phrase parsers is always worse, and much more artificial ambiguities are generated.

There have been some work (Collins et al., 1999b; Xia and Palmer, 2001) about converting dependency structures to phrase structures. Collins et al. (1999b) proposed an algorithm to convert the Czech dependency Treebank into a phrase structure Treebank and do dependency parsing through Collins (1999a)'s model. Results showed the accuracy of dependency parsing for Czech was improved largely. Xia and Palmer (2001) proposed a more generalized algorithm according to X-bar theory and Collins et al. (1999b), and they did some experiments on Penn Treebank. The results showed their algorithm produced phrase structures that were very close to the ones in Penn Treebank. However, we have to point out that they only computed the unlabeled performance but lost all the exact syntactic symbols. Different from tree-transformed PCFG based approach and lexicalized PCFG based approach, both of Collins et al. (1999b) and Xia and Palmer (2001) attempted to build some heuristic rules through linguistic theory, but didn't try to learn anything from Treebank.

Li and Zong (2005) presented a hierarchical parsing algorithm for long complex Chinese sentences with the help of punctuations. They first divided a long sentence into short ones according to punctuation marks, then parsed the short ones into sub-trees individually, and at last combined all the sub-trees into a whole tree. Experimental results showed the parsing time was reduced largely, and performance was improved too. Although the procedure of their parser is more close to human beings' manner, it appears a little shallow just using the punctuation marks.

In this paper our motivations are to bring more linguistic priori knowledge into phrase parsing procedure with the help of dependency structure, and make the parsing procedure flexibly.

Matsuzaki et al. (2005) defined a generative model called PCFG with latent annotations (PCFG-LA). Using EM-algorithm each non-terminal symbols was annotated with a latent variable, and a fine-grained model can be got. In order to get a more compact PCFG-LA model, Petrov et al. (2006) presented a split-and-merge method which can get PCFG-LA model hierarchically, and their final result outperformed state-of-the-art phrase parsers. To make the parsing process of hierarchical PCFG-LA model more efficient, Petrov and Klein (2007) presented a coarse-to-fine inference algorithm. In Section 4 of this paper, we try to combine the hierarchical PCFG-LA model in learning phase and coarse-to-fine method in inference phase into our parser in order to get an accurate and efficient parser.

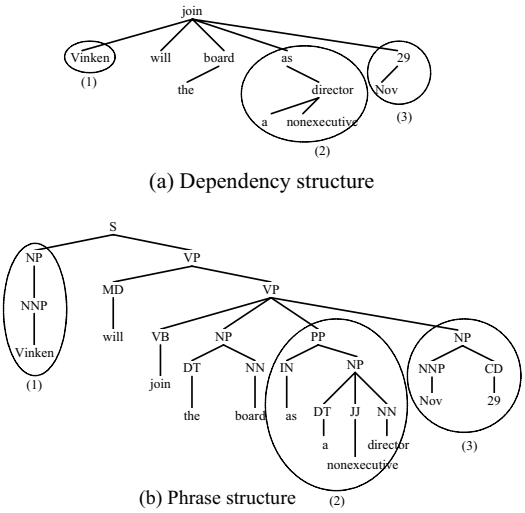


Figure 1. The consistency between phrase structure and dependency structure

3 Our framework

In this section, we first compare phrase structure with dependency structure of the same sentence, and get a consistent relationship among them. Then, based on this relationship, we present an inference framework to make the parsing procedure flexible and more efficient.

3.1 Analysis on consistency between phrase structure and dependency structure

Phrase structure and dependency structure are two different ways to represent syntactic structures of sentences. Phrase structure represents sentences by nesting of multi-word constituents, while dependency structure represents sentences as trees, whose nodes are words and edges represent the relations among words.

As we know, there are two kinds of dependency structures, projective structure and non-projective structure. For free-word order languages, non-projectivity is a common phenomenon, e.g. Czech. For languages like English and Chinese, the dependency structures are often projective trees. In this paper, we only consider English parsing based on Penn Treebank (PTB) and Chinese parsing based on Penn Chinese Treebank (PCTB), so we just research the consistency between phrase structure and projective dependency structure through PTB/PCTB.

Information carried by the two structures isn't equal. Phrase structure is more flexible, carries more information, and even contains all the information of dependency structure. So the task to convert a phrase structure to dependency structure is more straight, e.g. Nivre and Scholz (2004), Johansson and Nugues (2007). However, the reverse procedure is much more difficult, because dependency structure lacks the syntactic symbols, which are indispensable in phrase structure.

Although the two structures are completely different, they have consistency in some deep level. In this paper we analyze the consistency from a practical perspective in order to do phrase parsing with the help of dependency structure. Having investigated the two kinds of trees with dependency structure and phrase structure, we find a consistency[1] that each sub-tree in dependency structure must correspond to a sub-tree in phrase structure who dominates all the words appearing in dependency sub-tree. Figure 1 shows this relationship more intuitively. The dependency sub-tree surrounded by circle (1) in Figure 1(a) is a one-layer sub-tree, and has a corresponding phrase sub-tree surrounded by circle (1) in Figure 1(b). Both of the two sub-trees dominate the same word "Vinken". This consistency is also satisfied in other cases, e.g. two-layer sub-tree surrounded by circle (3) and three-layer sub-tree surrounded by circle (2) in Figure 1(a). These dependency sub-trees respectively have their corresponding phrase sub-trees dominating the same words in Figure 1(b).

This consistency brings us inspiration to make use of dependency structure for phrase parsing. In other words, in our method when a phrase sub-tree is generated from a dependency sub-tree, it must dominate all the same words with ones in the corresponding dependency sub-tree.

3.2 Inference framework

[1] Be aware that the consistency is irreversible and not every phrase sub-tree has its corresponding dependency sub-tree.

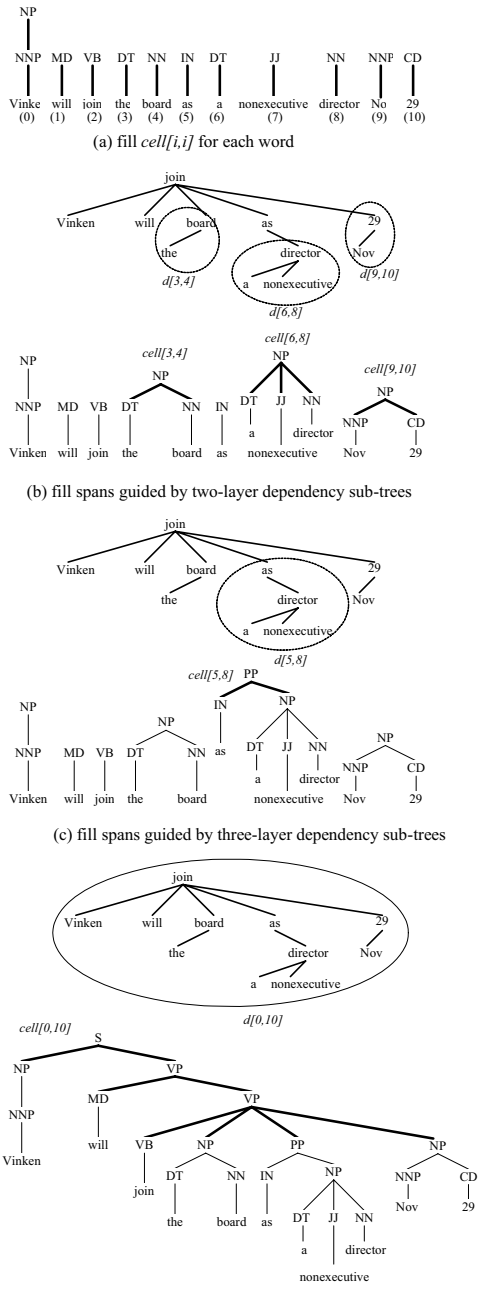


Figure 2. Parsing procedure of our inference framework guided by dependency structure

As we mentioned in Section 2, most of current inference algorithms are monotonous, which generate much more artificial ambiguities. For example, in Figure 1, if a sub-tree only dominating "board" and "as" is built (actually it is not occurred in golden tree) an artificial ambiguity is generated, and it thus will further bring a worse effect to other parts. The final precision will certainly descend. However, if we are given a corresponding dependency structure, those errors will be avoided. The consistency analyzed above tells us that there isn't a sub-tree dominating only "board" and "as" in dependency tree, so the two words can't build a sub-tree independently in phrase parsing. According to this strategy, we design an inference framework for phrase parsing.

Our inference framework parses a sentence flexibly with a traditional inference algorithm. The following terms will help to explain our work. A key data structure is *cell[i,j]*, which is used to store phrase sub-trees spanning words from positions i to j of the input sentence. *d[i,j]* is a dependency sub-tree spanning words from positions i to j. *cells[i,j]* is an array to store all the *cell*s which can be combined to build *cell[i,j]*. The pseudo-code of our inference framework is shown in Algorithm1. The line indicated by (1) and (2) gives us freedom to select any kinds of inference algorithms and matching parsing models.

Algorithm 1
InferenceFramework(sentence *S*, dependency tree *D*)

```
▷ initialize a List for the input sentence
for each word w_i in S do
      fill cell[i, i] and add it to a list L

▷ parse the cells[] hierarchically
for each d[s, t] of D in topological order do
      fill cells[s, t] with spans in L
      fill cell[s, t] with cells[s, t] through
                traditional inference algorithm   (1)
      add cell[s, t] to L

▷ extract the best tree
estimate all trees in cell[0, n]
                through parsing model             (2)
return the best phrase tree
```

Now, let's illustrate the flexible parsing procedure step by step through an example. Please see Figure 2. For simplicity, we just draw sub-trees of the final best tree, and ignore all the others. Figure 2(a) shows the procedure of filling *cell[i,i]* for each word. In Figure 2(b), there are three two-layer dependency sub-trees *d[3,4]*, *d[6,8]* and *d[9,10]*. So we try to generate phrase sub-trees for *cell[3,4]*, *cell[6,8]* and *cell[9,10]*, which have been annotated with bold edges. For example, we use sub-trees contained in *cell[6,6]*, *cell[7,7]* and *cell[8,8]* to

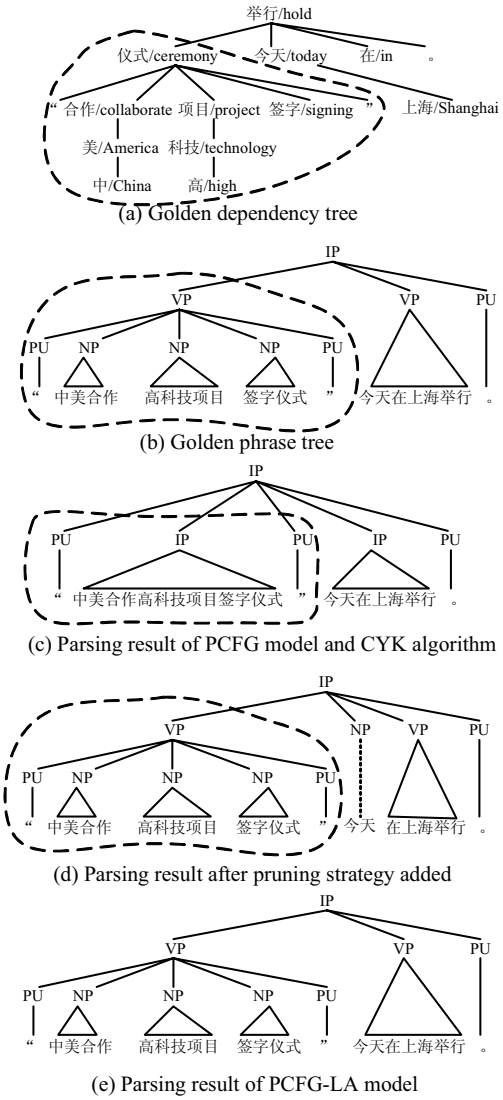


Figure 4. An example showing experimental results

build new sub-trees for *cell[6,8]*. Figure 2(c) and Figure 2(d) show the same procedure for parsing with the help of three-layer dependency sub-trees and four-layer dependency sub-trees individually. The generated phrase sub-trees are all annotated with bold edges in the figure. Obviously, the biggest dependency sub-tree is the whole dependency tree of sentence. In this example, when the four-layer dependency sub-tree is processed, the whole phrase trees are built. Usually, more than one phrase trees with the similar skeletons are generated. So we use a model to evaluate candidate results, and get out the one with the highest score as the final result.

Benefiting from the dependency structure, we can parse a sentence flexibly. Comparing with previous work on converting dependency structure to phrase structure (Collins et al., 1999b; Xia and Palmer, 2001), we make use of Treebank knowledge more sufficient with the help of traditional parsing technology. The search space has been pruned tremendously. As we know, the traditional parsing approach often tries to search all the n*(n+1)/2 cells for input sentence which has n words, but our parsing framework search cells intelligently with the help of corresponding dependency structure. Let's get a view of this through the sentence shown in Figure 2. From the whole parsing procedure shown in Figure 2, our framework just tries to fill 16 cells, which are *cell[i,i]* for each word, *cell[3,4]*, *cell[6,8]*, *cell[9,10]*, *cell[5,8]* and *cell[0,10]* hierarchically, but traditional parsing approach would try to fill all 66 cells. So 75.76% searching space has been pruned.

	English	Chinese
Train Set	Sections 2-21	Art. 1-270, 400-1151
Dev Set	Section 22	Articles 301-325
Test Set	Section 23	Articles 271-300

Table 1. Experimental settings

4 Experiments and results

In order to evaluate the effectiveness of our approach, we have done some experiments both for English parsing and Chinese parsing.

4.1 Preparation

To make comparison with previous work fairly, our experiments are based on general Treebank according to standard settings. We choose Penn English Treebank for English parsing experiments and Penn Chinese Treebank for Chinese. Table 1 shows the standard settings we take.

Because the two Treebanks are in type of phrase structure, we should get dependency structures corresponding with them. There are two ways to accomplish this work. First, use converting tools to get dependency trees directly through converting the original Treebanks, and the generated trees are always considered as golden trees during dependency parsing. Second, use a dependency parser with state-of-the-art

performance to parse all the sentences automatically. In this paper, we design two groups of experiments, as following:

(1) Phrase parsing with the help of golden dependency trees.
(2) Phrase parsing with the help of N-best dependency trees generated automatically.

4.2 Phrase parsing with golden dependency trees

In order to verify how much dependency structure can help phrase parsing and get an upper bound of our approach, we do phrase parsing with the help of golden dependency trees in this subsection.

Based on the parsing framework shown in Figure 3, we only use the basic PCFG in learning phase and our inference framework with basic CYK algorithm in inference phase. The parsing results are shown with the mark (1) in Table 2 for English and Table 3 for Chinese respectively.

Having investigated the generated trees with golden trees, we find the consistency of dependency structure and phrase structure is broken by some trees. Let's get a view of this through an example from Penn Chinese Treebank. In Figure 4(a), the dependency sub-tree surrounded by circle tells us that there must be a phrase sub-tree which dominate the word sequence of "中 美 合作 高 科技 项目 签字 仪式" (the signing ceremony of collaborating in high technology between America and China), and the golden phrase tree shown in Figure 4(b) has a corresponding sub-tree surrounded by circle indeed. However, the parsing tree generated by our approach shown in Figure 4(c) doesn't conform. There are three sub-trees dominating the word sequence mutually, but they don't construct a whole one. In our opinion, the contradiction derived from binarizing process of CYK[2]. The binary trees generated by our algorithm have consisted with the consistency originally, but after debinarizing process the consistency is broken.

Trying to check our opinion, we add a pruning strategy to the original inference algorithm to prune all the medial nodes which may break the consistency during parsing procedure. With the help of pruning strategy, the performances of English and Chinese are all improved further. Corresponding figures are shown in Table 2 and Table 3 with the mark (2). The parsing result of above example is shown in Figure 4(d) and the error appearing in Figure 4(c) is corrected naturally after the pruning strategy added.

Comparing with previous work which have done much work in learning phase, our algorithm achieves such amazing results only using basic PCFG model. From this aspect, our inference framework is much more effective. However, there are still some errors our approach can't deal with. For example, in Figure 4(d) the sub-tree rooted at NP and dominating word sequence of "今天 在 上海 举行" (hold in Shanghai today) is separated by two sub-trees. The reason is that the model (basic PCFG) we use in learning phase is too coarse to disambiguate sufficiently. So we don't pin all hopes in inference phase. We also modify the model in learning phase. PCFG-LA is one of the most successful models in phrase parsing, so we choose PCFG-LA as the model in learning phase. After this modification, performance of our approach has been improved delightedly. F1 score is 96.08% for English and 90.06% for Chinese. The line marked with (3) in Table 2 and Table 3 shows more details.

4.3 Phrase parsing with N-best dependency trees generated automatically

The experimental results shown in subsection 4.2 bring us confidence that do phrase parsing with the help of dependency structure is a highly effective approach. However, we don't usually have golden dependency structures, and a more acceptable way is using a dependency parser to generate dependency trees automatically. In this subsection we explore feasibility and effectiveness of phrase parsing with the help of dependency trees generated automatically. As we all know, even state-of-the-art dependency parser cannot generate totally correct result. So in order to make our system more robust we use N-best dependency structures to guide phrase parsing procedure.

[2] The premise of using CYK is that all the rules must have CNF form. So we usually bring some medial nodes to binarize rules gathered from Treebank.

	length<=40			all sentences		
	Precision	Recall	F1	Precision	Recall	F1
(1) Using PCFG and CYK	90.28	88.41	89.34	90.11	88.32	89.21
(2) Using pruning strategy	90.69	89.53	90.11	90.51	89.45	89.97
(3) Using PCFG-LA	96.28	95.97	96.13	96.25	95.91	96.08

Table 2. Parsing performance (%) for English with the help of golden dependency tree.

	length<=40			all sentences		
	Precision	Recall	F1	Precision	Recall	F1
(1) Using PCFG and CYK	86.89	78.25	82.34	85.56	77.43	81.29
(2) Using pruning strategy	87.65	82.33	84.91	86.39	81.45	83.85
(3) Using PCFG-LA	91.51	91.26	91.38	90.43	90.79	90.61

Table 3. Parsing performance (%) for Chinese with the help of golden dependency tree.

We choose MSTParser[3] which is the most famous dependency parser and modify it to generate N-best dependency trees. The oracle unlabeled accuracy of N-best dependency trees generated from 1-order model is shown in Table 4. To show the effectiveness of our approach, we choose Berkeleyparser[4] as the baseline parser, take the same configuration and combine it into our general parsing framework shown in Figure 3.

Considering the number of dependency structures (N-best) will affect the final result, we make use of the development set shown in Table1 to turning parameters. We parse the development set many times with different number of dependency structures. The F1 scores are shown in Figure 5 for English and Figure 6[5] for Chinese. From Figure 5 and Figure 6, we can find when we use 10-best dependency structures the performance is better. So we choose 10-best dependency trees for the test set.

The final performances of test set comparing with previous work are shown in Table 5 and Table 6. We can easily find that our approach has outperformed all the parsers which aren't improved through reranking stage or semi-supervised approach. Although there is still a margin between our parser and reranked parser or semi-supervised parser, we believe that the parsing performance can be improved further if we bring the reranking or semi-supervised approaches into our parsing framework.

4.4 Discussion

The experiment of parsing with golden dependency structure gets an amazing performance. It brings us a new way to build PTB/PCTB style phrase structure corpus. Because dependency structure is much easier to get through hand annotating than phrase structure, we can build a dependency structure corpus first, and then get phrase structure corpus through our approach guided by the dependency structure corpus.

The experiment of parsing with N-best dependency structures generated automatically uplifts the parsing performance to a new height. It brings us a more applied parsing tool for other NLP applications.

From the experiments in Section 4.2, we can find that even using the golden dependency structure we can't get totally correct phrase structure. The reason is that although every dependency sub-tree has its corresponding phrase sub-tree, not every phrase sub-tree has its corresponding dependency sub-tree. So the remainder errors can't be solved only by dependency structure and a better way is to modify the parsing model.

5 Conclusion and Future work

In this paper, we present a novel phrase parsing approach with the help of dependency structure. Based on the consistency between phrase structure and dependency structure, we propose a novel inference framework. Guided by the inference framework, inference algorithms parse sentences hierarchically with the help of dependency structures. Experimental results show that our approach can efficiently get better performance with both golden dependency structure and N-best dependency

[3] http://www.ryanmcd.com/MSTParser/MSTParser.html
[4] http://code.google.com/p/berkeleyparser/
[5] F1 score at n=0 is the result of Berkeley parser running on my machine

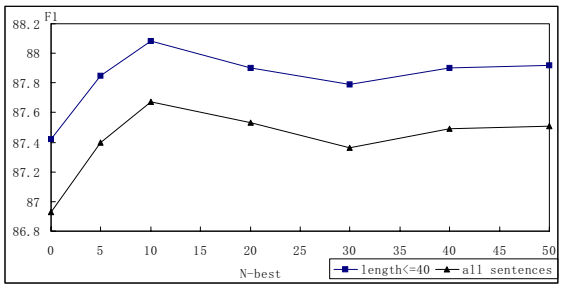


Figure 5. F1 scores (%) of Dev Set for English with the help of N-best dependency trees

	English		Chinese	
	len<=40	all	len<=40	all
5-best	90.62	90.49	87.92	84.93
10-best	91.6	91.48	89.05	85.9
20-best	92.36	92.21	89.86	86.79
30-best	92.74	92.6	90.3	87.28
40-best	92.96	92.83	90.62	87.63
50-best	93.08	92.95	90.79	87.87

Table 4. Oracle unlabeled accuracy (%) of N-best dependency structures generated from MSTParser

	<=40	All
Collins(1999)	88.6	88.2
Charniak and Johnson(2005)	90.1	89.55
Petrov and Klein(2007)	90.6	90.05
This Paper	**91.13**	**90.54**
Reranked		
Charniak and Johnson(2005)	92.0	91.4
Huang(2008)	——	**91.7**
Semi-supervised		
McClosky et al. (2006)	——	**92.1**

Table 5. F1 (%) of Test Set for English of previous work and our approach

	<=40	All
Chiang et al.(2002)	79.93	76.57
Bikel Thesis(2004)	81.2	79.0
Petrov and Klein(2007)	86.3	83.32
This Paper	**86.76**	**83.93**
Semi-supervised		
Huang and Harper(2009)	——	**85.18**

Table 6. F1 (%) of Test Set for Chinese of previous work and our approach

structures generated automatically.

However, there are still some problems remaining to further study. First, in our approach we just use the unlabeled dependency trees. The relationship labels carry some useful information too, and we can make use of them to further improve phrase parsing. Second, phrase structure can also help the process of dependency parsing (McDonald et al., 2006), so

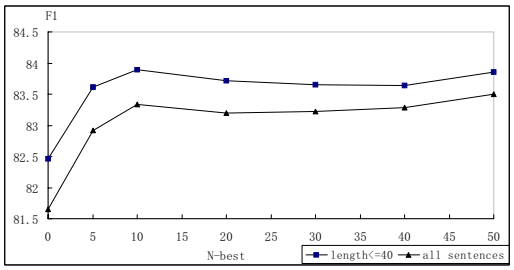


Figure 6. F1 scores (%) of Dev Set for Chinese with the help of N-best dependency trees

we can combine phrase parsing process and dependency parsing process together and make them help each other.

Acknowledgments

The research work has been partially funded by the Natural Science Foundation of China under Grant No. 60975053, 90820303 and 60736014, the National Key Technology R&D Program under Grant No. 2006BAH03B02, the Hi-Tech Research and Development Program ("863" Program) of China under Grant No. 2006AA010108-4, and also supported by the China-Singapore Institute of Digital Media (CSIDM) project under grant No. CSIDM-200804.

References

Alexander Krotov, Mark Hepple, Robert Gaizauskas, and Yorick Wilks. 1998. *Compacting the Penn Treebank grammar*. In ACL-COLING '98, pages 699-703.

Dan Klein and Chris Manning. 2003. *Accurate Unlexicalized Parsing*, In ACL '03, pages 423-430.

Daniel Jurafsky and James H. Martin. 2007. *SPEECH and LANGUAGE PROCESSING*--a draft, at Chapter 14.4.

Daniel M. Bikel. 2004. *On the Parameter Space of Generative Lexicalized Statistical Parsing Models*. Ph.D. thesis, U. of Pennsylvania.

Daniel M. Bikel. 2004. *Intricacies of Collins' Parsing Model*. In Computational Linguistics, 30(4), pages 479-511.

Daniel M. Bikel and David Chiang. 2000. *Two Statistical Parsing Models Applied to the Chinese Treebank*. In the Proceedings of the Second Chinese Language Processing Workshop.

David Chiang and Daniel M. Bikel. 2002. *Recovering Latent Information in Treebanks*. In COLING '02.

David McClosky, Eugene Charniak and Mark Johnson. 2006. *Effective self-training for parsing*. In ACL-06.

Deyi Xiong, Qun Liu and Shouxun Lin. 2005. *Lexicalized Beam Thresholding Parsing with Prior and Boundary Estimates*. the 6th Conference on Intelligent Text Processing and Computational Linguistics (CICLing), Pages 132 – 141.

D.H. Younger. 1967. *Recognition and parsing of context-free-languages in time n^3*. Information and Control, 10(2):189-208.

Eugene Charniak. 1997. *Statistical parsing with a context-free grammar and word statistics*. Proceedings of the Fourteenth National Conference on Artificial Intelligence AAAI Press/MIT Press, Menlo Park.

Eugene Charniak. 2000. *A maximum-entropyinspired parser*. In NAACL '00, pages 132–139.

Eugene Charniak and Mark Johnson. 2005. *Coarse-to-Fine n-Best Parsing and MaxEnt Discriminative Reranking*. In ACL '05.

Fei Xia and Martha Palmer. 2001. *Converting Dependency Structures to Phrase Structures*. The 1st Human Language Technology Conference (HLT-2001).

H. Gaifman. 1965. *Dependency Systems and phrase-Structure Systems*. Information and Control, pages 304-337.

H. Yamada and Y. Matsumoto. 2003. *Statistical dependency analysis with support vector machines*. In Proceedings of IWPT

J. Nivre, M. Scholz. 2004. *Deterministic dependency parsing of English text*. In COLING '04.

Liang Huang. 2008. *Forest reranking: Discriminative parsing with non-local features*. In ACL '08.

Mark Johnson. 1998. *PCFG models of linguistic tree representations*. Computational Linguistics, 24(4):613–632.

Michael Collins. 1999a. *Head-Driven Statistical Models for Natural Language Parsing*. Ph.D. thesis, U. of Pennsylvania.

Michael Collins, Jan Hajic, Lance Ramshaw and Christoph Tillmann. 1999b. *A Statistical Parser for Czech*. In ACL '99.

Richard Johansson and Pierre Nugues. 2007. *Extended Constituent-to-dependency Conversion for English*. In Proceedings of NODALIDA.

Roger Levy, Christopher Manning. 2003. *Is it harder to parse Chinese, or the Chinese Treebank?* In ACL '03.

Ryan McDonald, Koby Grammer and Fernando Pereira. 2006. *Online learning of approximate dependency parsing algorithms*. In EACL '06.

Slav Petrov and Dan Klein. 2007. *Improved Inference for Unlexicalized Parsing*. In HLT-NAACL '07.

Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. 2006. *Learning Accurate, Compact, and Interpretable Tree Annotation*. In COLING-ACL '06.

T. Matsuzaki, Y. Miyao, and J. Tsujii. 2005. *Probabilistic CFG with latent annotations*. In ACL '05, pages 75–82.

T. Kasami. 1965. *An efficient recognition and syntax analysis algorithm for context-free languages*. Technical Report, AFCRL-65-758, Air Force Cambridge Reserch Lab., Bedford, MA

Xavier Carreras, Michael Collins, and Terry Koo. *TAG, Dynamic Programming and the Perceptron for Efficient, Feature-rich Parsing*. In CONLL '08.

Xing Li, Chengqing Zong. 2005. *A Hierarchical Parsing Approach with Punctuation Processing for Long Complex Chinese Sentences*. In IJCNLP '05

Yusuke Miyao, Rune Sætre, Kenji Sagae, Takuya Matsuzaki and Jun'ichi Tsujii. 2008. *Task-oriented Evaluation of Syntactic Parsers and Their Representations*. In ACL '08, pages 46-54.

Zhongqiang Huang, Mary Harper. 2009. *Self-Training PCFG Grammars with Latent Annotations Across Languages*. In EMNLP '09.

Automatic Generation of Semantic Fields for Annotating Web Images

Gang Wang[§♀], Tat Seng Chua[#], Chong-Wah Ngo[⊙], Yong Cheng Wang[♀]
♀Shang Hai Jiao Tong University
School of Computing, National University of Singapore
⊙Dept of Computer Science, City University of HongKong
§ Na Xun Hi-Tech Application Institute
wanggang_sh@hotmail.com,chuats@comp.nus.edu.sg,
cwngo@cs.cityu.edu.hk,ycwang@mail.sjtu.edu.cn

Abstract

The overwhelming amounts of multimedia contents have triggered the need for automatically detecting the semantic concepts within the media contents. With the development of photo sharing websites such as Flickr, we are able to obtain millions of images with user-supplied tags. However, user tags tend to be noisy, ambiguous and incomplete. In order to improve the quality of tags to annotate web images, we propose an approach to build Semantic Fields for annotating the web images. The main idea is that the images are more likely to be relevant to a given concept, if several tags to the image belong to the same Semantic Field as the target concept. Semantic Fields are determined by a set of highly semantically associated terms with high tag co-occurrences in the image corpus and in different corpora and lexica such as WordNet and Wikipedia. We conduct experiments on the NUS-WIDE web image corpus and demonstrate superior performance on image annotation as compared to the state-of-the-art approaches.

1 Introduction

The advancement in computer processor, storage and the growing availability of low-cost multimedia recording devices has led to an explosive growth of multimedia data. In order to effectively utilize such a huge amount of multimedia contents, we need provide tools to facilitate their management and retrieval. One of the most important tools is the automatic media concept detectors, which aim to assign high-level semantic concepts such as "bear" to the multimedia data. More formally, the concept detection for an web image is defined as: given a set of predefined concepts $\vec{C}$: [C_1, C_2 ...C_n], we assign a semantic concept C_i to the image if it appears visually in the image. Traditionally, such concept detectors are built by the classifier approaches. The performance of such detectors depends highly on the quality of training data. However, preparing a set of high quality training data usually needs a large amount of human labors. On the other hand, the social web is changing the way people create and use information. For example, users started to develop novel strategies to annotate the massive amount of multimedia information from the web. In image annotation, Kennedy et al. (2006) explored the trade-offs in acquiring training data by automated web image search as opposed to manual human labeling. Although the performance of systems with training data obtained by manual human labeling is still better than those whose training data is acquired by automated web search, the latter approaches have attracted many researchers' interest due to their potential in reducing human label efforts. However, the tags in the web images are known to be ambiguous and overly personalized (Matusiak 2006).

Figure 1 gives four examples to illustrate the relationships between the visual concept "bear" and the annotation tag "bear". Generally speaking, there are four types of relationships:

Coling 2008: Poster Volume, pages 1301–1309,
Beijing, August 2010

- The relevant tag: The user-tag "bear" properly reflects the content of an image, as shown in Figure 1(a). While "bear" has multiple senses, the visual concept corresponds directly to the most common sense of "bear".
- The ambiguous tag: The user-tag "bear" is ambiguously related to the visual content, as shown in Figure 1(b). In this example, the visual content is related to another sense of "bear": "a surly, uncouth, burly, or shambling person" (Merriam-Webster dictionary, 2010).
- The noisy tag: The user-tag "bear" is a noisy tag, as shown in Figure 1(c). In this example, the visual content is irrelevant to the concept "bear".
- The incomplete tag: The user-tag "bear" doesn't occur in the tag list of Figure 1(d). However, many human annotators believe that the visual concept "bear" exist in the Figure 1(d). Also, in Wikipeida, a panda is defined as a kind of a bear.

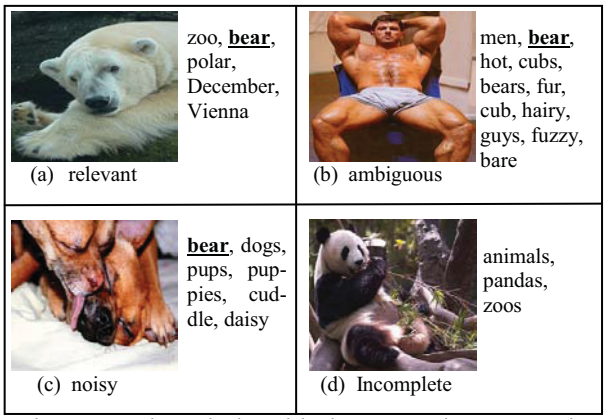


Figure 1: The relationship between the tags and the visual concept "bear" in NUS-Wide corpus.

In this paper, we aim to assign relevant tags to images in order to reduce the effects of ambiguous, noisy and incomplete tags. To distinguish relevant tags from other sense of tags, a common practice is to perform word sense disambiguation (WSD) to predict the right sense of a tag. Nevertheless, performing a WSD on a noisy and sparse set of tags, where the order and position of tags do not matter, is by no means easy. Most existing works on WSD, such as Navigli (2009) are based on clean data and word neighborhood statistics. They cannot be directly applied to address this problem. Although there are some works such as Wang et al. (2003) on capturing the semantics of noisy data, the problem of ambiguous words has not been considered. In addition, some semantic models such as PLSA (Hofmann 1999), LDA (David et al. 2003) have been proposed to capture the semantics. However, one challenge of employing such models is that there are many noisy tags in the web image domain. The reason for noisy tags is that the purpose of tagging is not only for content description, but also for other factors such as getting attention and so on (Ame and Naaman, 2007, Bischoff et al. 2008).

Given a web image with a tag list, we propose an approach to predict the "Semantic Field" of the image. Semantic Field (Jurafsky and Martin 2000) is designed to capture a more integrated relationship among the entire sets of tags. In our work, we consider four different cases of examples, as shown in Figure 1. In 1(a), the concept "bear" will be assigned to the image with relatively high probability, because "zoo", "bear", and "polar" provide clues that "bear" is the major focus of the image. In 1(b), the concept "bear" could possibly be disambiguated as not related to "animal", the most common sense of "bear", by investigating other tags such as "men","guys". In 1(c), the image will not be labeled as "bear", since the surrounding tags such as "dogs", "pups" do not support the existence of "bear" in the image. In 1(d), although the concept "bear" is missing, the image will be still labeled as "bear" since the surrounding tags such as "pandas", "animals", and "zoos" jointly suggest that "bear" appears in the image. The significance of user tags towards a target concept can be modeled from three different sources: the statistics from the web image corpus, Wordnet and Wikipedia. In summary, instead of directly matching the keywords and tags, we consider tags of an image collectively to predict the underlying semantic field. Ideally, the semantic field can highlight the major visual concepts in images so that we can assign the correct semantic labels to the images.

In the rest of this paper, we discuss related work in Section 2, while Section 3 reports the building of Semantic Fields and its application to web image ranking. Section 4 discusses the experimental setup and results. Finally, Section 5 contains our concluding remarks.

2 Related Work

In this section, we report the works on Semantic Field theory, text analysis in multimedia and the existing systems for a web image corpus.

2.1 Semantic Fields

Semantic Fields have been hotly debated in linguistics community (Grandy 1992, Garret 1992). Compared to lexical analysis, it considers the entire sets of words instead of a single word. The FrameNet project (Baker et al. 1998) is an attempt to realize the Semantic Field. However, the problem with FrameNet project is that it needs extensive human efforts to define the thematic roles for each domain and each frame, and hence it is domain specific.

2.2 Text Analysis in Multimedia

In multimedia, one of the important tasks is concept detection, which attempts to find the visual appearance of a concept such as "bear" in an image. However, due to the large variations in the low level visual feature space such as color, texture etc, in many cases, researchers are hardly able to capture the concept by visual information alone. Some researchers attempted to employ natural language analysis to detect the visual concept. Rowe (1994) explored the syntax of images' captions to infer the visual concepts present in images. For example, he found that the primary subject noun phrase usually denotes the most significant information in the media datum or its "focus". He assumed that both visual and text features will describe the same focus of the content. Wang et al. (2008) employed the similar idea to infer visual concepts in news video. They first aligned text information with visual information, and then captured the text focus to infer the visual concept. These works suggest that we can transfer the problem of visual concept detection to that of finding a text focus.

In addition, researchers proposed statistical models to combine text and visual features, such as the translation model (Duygulu et al. 2002, Jin et al. 2005), cross media relevance model (Jeon et al. 2003) and continuous relevance model (Lavrenko and Jeon, 2003). However, no matter what models we used, the annotation accuracy is still quite low, partially because of the existence of noise in tags. Jin et al. (2005) provided a solution to tackle such a noisy tag problem. They first investigated various semantic similarity measures between each keyword pairs in the tag list based on Wordnet. They then regarded non-correlated keywords as noises and discarded them. In this paper, there are three major differences between our work and the above work. First, because tags from Internet are not always included in Wordnet, we employ multi-resources of information to analyze the semantics. Second, we extend the analysis of the word pair relationship to the Semantic Field analysis. Third, since it is not easy to identify the noise in the tag list directly, we only analyze the tags which are highly relevant to the concept with a specific sense.

2.3 The State of the Art Systems

NUS-WIDE (Chua et al. 2009) is a large scale Web image corpus. It provides not only social tags from the web, but also the "gold" labels (or ground truth) for 81 concepts from large human labeling efforts. As far as we know, there are two reported systems that used the whole NUS-WIDE corpus to test their proposed methods. In Chua et al. (2009), the 81 concepts are detected by k nearest neighbor using the visual features of: color moments, color auto-correlogram, color histogram, edge direction histogram, wavelet texture, and a bag of visual words. The mean average precision (MAP) for the 81 concepts reaches 0.1569. Gao et al. (2009) extended the k-NN approach to use both text tags and visual information. For the tag information, they made use of the co-occurrence information to compute the probability of an image belonging to a contain concept. They used the same visual features as in (Chua et al. 2009). In their work, the taxonomy in WordNet is exploited to identify whether a target concept is generic or specific. The co-occurrence tag analysis is employed for generic concepts, while visual analysis is used for specific concepts. The MAP for this approach reaches 0.2887.

3 Building Semantic Fields for Annotating Web Images

In this paper, we attempt to capture text semantics collectively from the tag list of images to annotate their visual contents. Semantic Fields consist of a selected subset of the tag list and

the choice of these tags is based on their relevance to the contents of the targeted image with a specific sense. There are three characteristics in our Semantic Field model. First, the Semantic Field is built by only a subset of tag list. For example, the Semantic Field in Figure 1(a) is {zoo, bear, polar}. It could partially reduce the effect of the noise. Second, because inferring the visual concept of an image is more reliable by joint analysis of tags in the Semantic Field, rather than investigating one tag at a time in the whole tag list, we analyze the whole Semantic Field as a unit. By utilizing the context information in Semantic Field, the problems of ambiguous, noisy and incomplete tags are partially tackled. Third, we perform normalization to estimate the importance of Semantic Field, which is discussed in Section 3.1. If the value is large, it suggests that most of the tags in the image support the Semantic Field; that is, the probability that the target concept is the focus of the image is high, and vice versa. Such a design aims to minimize the effects of noisy and ambiguous tags.

3.1 A Probabilistic Model

We denote C_x as a target concept that appears in the content of an image. We want to determine the set of tags that are related to C_x from the user-supplied tags by building a Semantic Field SF_i for each image. The probability of the appearance of concept $P(C_x \mid SF_i)$ can be computed as:

$$P(C_x \mid SF_i) = \frac{P(SF_i \mid C_x) \times P(C_x)}{P(SF_i)} \qquad (1)$$

For the purpose of collecting and annotating images and simplifying the model, we did not consider the prior knowledge for each image. Thus, the prior probability $P(C_x)$ can be viewed as a constant with respect to a concept C_x. In addition, the range of the normalization factor $P(SF_i)$ is expected to be small, which will not affect the annotation of web images. This assumption is reasonable due to the fact that there are a large number of different tags, and these tags can be combined to form any Semantic Field in an arbitrary manner. The number of combinations is exponential to the number of possible tags available. This is also evident by the observation that most tag lists associated with the images are unique. In other words, two images with the same Semantic Field are seldom found in reality. With these in mind, Equation (1) can be approximated and simplified to:

$$P(C_x \mid SF_i) \propto P(SF_i \mid C_x) \qquad (2)$$

Given a Semantic Field SF_i, it may include n related tags $T_1, T_2, T_3, ..., T_n$. Thus Equation (2) is expanded to:

$$P(SF_i \mid C_x) = P(T_1, T_2, ...T_n \mid C_x) \qquad (3)$$

Two obvious approaches to compute Equation (3) are using the product of the individual terms or chain rule decomposition. However, we consider the individual terms to be interdependent and the chain rule decomposition is not easy to compute. To simplify the model, we employ the normalized linear fusion to expand Equation (3) as follows:

$$\mathrm{P}(T_1, T_2, ...T_n \mid C_x) = \frac{\sum_{i=1}^{n} P(T_i \mid C_x)}{\mathrm{TN}} \qquad (4)$$

The normalization factor is the total number (TN) of tags in the image tag list.

3.2 Using Multiple External Sources

To estimate the probability of a tag T_i given a target concept C_x, i.e., $P(T_i|C_x)$, we consider both the domain knowledge and general knowledge acquired from Internet. For the former, we utilize the co-occurrence statistics of tags in images which can be computed offline from any web image corpus. For the latter, we employ WordNet and Wikipedia for inferring the relatedness between tags and a target concept. Combining different knowledge sources, the probability is estimated as:

$$P(T_i \mid C_x) = P(T_{i\_wd} \mid C_x) \times P(T_{i\_wiki} \mid C_x) \times P(T_{i\_co} \mid C_x) \quad (5)$$

where T_{i_wd}, T_{i_wiki}, T_{i_co} represent the tag occurrences in WordNet, Wikipedia and co-occurrence statistics, respectively.

To compute Equation (5), we query different information sources using the target concept C_x. In WordNet, because the sense of the concept usually refers to the most common sense in our corpus, we choose the most common sense (noun) as the target. Using Figure 2 as an example, the concept "bear" is defined in WordNet as “massive plantigrade carnivorous or omnivorous mammals with long shaggy coats and strong claws”. In Wikipedia, with Figure 3 as an example, the related page is downloaded to

describe the concept "bear". For the co-occurrence statistics of the tag lists, we estimate their values from co-occurrence information from the image corpus. With the above knowledge, we compute the conditional probability of a tag being related to C_x as:

$$P(T_j \mid C_x) = \frac{\#(T_j, C_x)}{\#(C_x)} \qquad (6)$$

where j = {wd, wiki, co}, $\#(T_j, C_x)$ indicates the number of times the tag and the concept co-occur in an information source, and $\#(C_x)$ denotes the number of times the concept C_x appear in the information source. In addition, we employ an add-one smoothing approach [Jurafsky and Martin 2000] to further process the results.

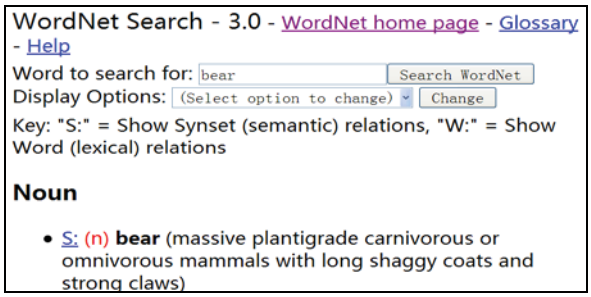


Figure 2: The information in WordNet

Figure 3: The information in Wikipedia.

Given a concept with a special sense, for all the tags in the corpus, we can obtain the conditional probabilities of each tag T_i based on Equation (5). We rank the tags according to $P(T_i \mid C_x)$. To reduce computations, we select the top N (N=200) tags as the highly related tags to a given concept and place them in a dictionary.

3.3 Building Semantic Field for Image Annotation

We now build the Semantic Fields to rank the images with respect to concept C_x. The detailed algorithm is shown in Figure 4.

Input:
1) Given a target concept, we rank all the tags in the corpus based on Equation (5).
2) Given a web image, we have a list of annotation tags ($l_1, l_2, ...l_{n1}$).

Step 1: Generate a dictionary (D) based on top N tags

Step 2: For (i=1; i<n1; i++)
If ($l_i \in D$) then put l_i into the Semantic Field for the image.

Step 3: Annotate the images and compute the probability of the occurrence of the concept via Equation (4)

Figure 4: The algorithm for building the Semantic Fields and annotating the images.

The algorithm comprises three steps:

1. bear	2. bears	3. polar	4. species
5. panda	6. cubs	7. giant	8. grizzly
9.teddy	10. pandas	…	…

Table 1: The top 10 tags for concept "bear" in most common sense.

First, given a target concept with a specific sense, we generate a dictionary based on the top N candidate tags as discussed in Section 3.2. Table 1 shows the top 10 tags in the dictionary for the concept "bear" with the most common sense. As we want to distinguish single and plural noun for different visual concepts, we do not employ the stemming algorithm. Although the results are not ideal, we find that many highly related words are included in the dictionary.

Second, we infer the annotation tags of the image from the dictionary and use that to build the Semantic Fields. Figure 1 demonstrates the resulting of Semantic Fields for images in Table 2.

Third, we assign the tags to images based on their Semantic Fields. Because most of the tags in Figure 1(a) and 1(d) are highly relevant to "bear" with the most common sense, we assign the semantics to these two images with high probabilities. Thus, the problem of incomplete tags is tackled in this case. On the other hand, since most of the tags in Figure 1(b) and 1(c) fail to support the concept "bear" with the most

common sense (the Semantic Field obtains less than 20% of tags' support), we only assign the semantics with very low probabilities. Thus, the ambiguous and noisy problem can be partially tackled.

Semantic Field for Figure 1 (a)	{zoo, bear, polar}
Semantic Field for Figure 1 (b)	{bear, bears}
Semantic Field for Figure 1 (c)	{bear}
Semantic Field for Figure 1 (d)	{animals, pandas, zoos}

Table 2: Semantic Fields of images in Figure 1.

4 Experiments

In this section, we first introduce the test-bed and measurement of the experiments. We then report the results and compare them with the state-of-the-art systems tested on NUS-WIDE corpus.

The NUS-Wide corpus (Chua et al. 2009) includes 269,648 images with 5,018 user-provided tags, and the ground-truth for 81 concepts for the entire database. These concepts are grouped into six different categories: graph, program, scene /location, event/activities, people and object. The choice of concepts is based on the generality and popularity in Flickr, the distributions in different categories and the common interests of the multimedia community. This corpus includes two parts. The first part contains 161,789 images to be used for training and the second part contains 107,859 images is used for testing.

The performance of the system is measured using the mean average precision (MAP) based on all the test images for all the 81 concepts. This is the same as the evaluation used in TRECVID. The MAP combines precision and recall into one performance value. Let $p^k = \{i_1, i_2, ..., i_k\}$ be a ranked version of the resulting set A. At any given rank k, let $R \bigcap p^k$ be the number of relevant images in the top k of p, where |R| is the total number of relevant images. Then MAP for the 81 concepts C_i is defined as:

$$MAP = \frac{1}{81}\sum_{C_i=1}^{81}[\frac{1}{|R|}\sum_{k=1}^{A}\frac{R \bigcap p^k}{k}\varphi(i_k)] \quad (6)$$

where the indictor function $\varphi(i_k) = 1$ if $i_k \in R$ and 0 otherwise.

4.1 Comparison with the State-of-the-Art Systems

We compare our approach against the reported systems on NUS-WIDE corpus.

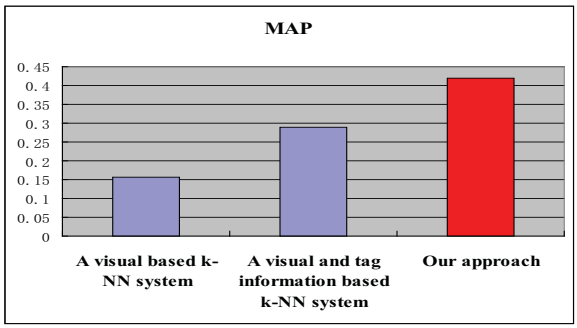
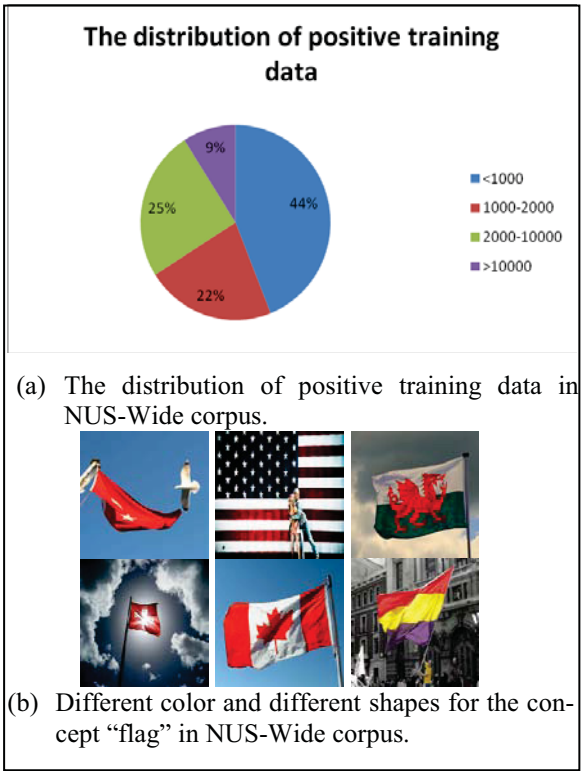


Figure 5: The comparison with the state-of-the-art system

In our approach, we employ the Semantic Field to annotate the images, which requires neither training data nor visual analysis, and is running directly on the test data. In contrast to the two previous approaches in Section 2.3, the input to Semantic Field is simply the tag list of an image. Figure 5 shows the performance comparisons among the three tested approaches. As compared to (Chua et al. 2009) and (Gao et al. 2009), which exhibit the best performance on NUS-WIDE so far, Semantic Field achieves a MAP of 0.4198 which shows a 45.4% improvement.

The reason for the superior performance of our approach is that there is insufficient training data, which means that most learning-based systems could not perform well. As seen in Figure 6(a), 44% of concepts have less than 1,000 positive training data. This is insufficient for training the classifiers for the visual concepts. Take the visual concept "flag" as the example. Considering that there are at least 200 national flags from different countries and regions, not to mention other types of flags such as holiday flag, there are large variations in concept "flag" as shown in Figure 6(b). Hence it is difficult to train a classifier with visual analysis by having only 214 positive training samples. This suggests that there may be a large

gap between the training and test data. On the other hand, because web images include not only visual features but also text information, we could employ text analysis to infer the visual concept. The advantages of our Semantic Field approach are that we could analyze multiple information sources to reduce the text variations and the performance of our approach is independent of the training data and visual features. With the increasing size of the corpus, the problems of few positive training data and large visual diversity between training and test data will be exacerbated. This is the reason why our approach is more robust than those based on visual analysis and traditional learning-based approaches.



(a) The distribution of positive training data in NUS-Wide corpus.

(b) Different color and different shapes for the concept "flag" in NUS-Wide corpus.

Figure 6: Various visual patterns need a lot of training data

4.2 The Noisy, Ambiguous and Incomplete Tag Problems

We design the second experiment to evaluate the ability of our algorithm to tackle the noisy and ambiguous and incomplete tag problem in user-supplied tags. The baseline system is a keyword (tag) matching algorithm. That is, if the image contains the keyword in the tag list, the algorithm will regard it as relevant to the concept; otherwise, it is irrelevant. The results are shown in Figure 7.

We found that our approach achieves a relative improvement of 38% as compared to the keyword matching approach. This is because the Semantic Field approach selects and analyzes a group of tags as a whole, which provides essential context information and reduces the effects of noisy, ambiguous and incomplete tags.

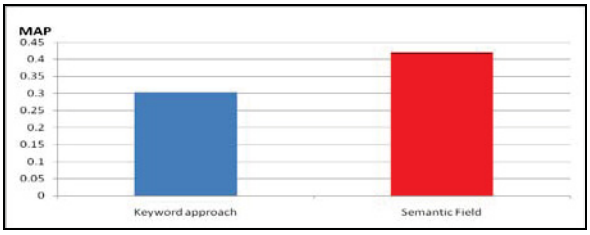


Figure 7: Comparison with keyword matching approach

For completeness, we also evaluate the system using the Equations (7) and (8) according to the top k images (k=1000, 2000, 3000, 4000, 5000).

$$P(tag) = \frac{\sum_{i=1}^{N} \frac{\#(p_i)}{\#(A_i)}}{N} \quad (7)$$

$$R(tag) = \frac{\sum_{i=1}^{N} \frac{\#(p_i)}{\#(T_i)}}{N} \quad (8)$$

We use p_i to represent the number of images with the target concept and A_i to represent the number of retrieved images for tag i. N denotes the number of different detected concepts (tags) in the ground truth set. In this corpus, the value of the N is 81. T_i is the number of the ground truth for a certain target concept.

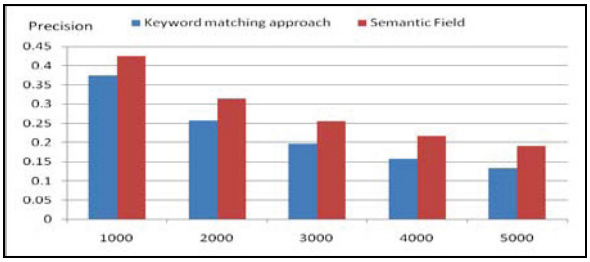


Figure 8: Comparison in precision on top-k image ranking. The x-axis indicates the value of k, while the y-axis shows the P(tag).

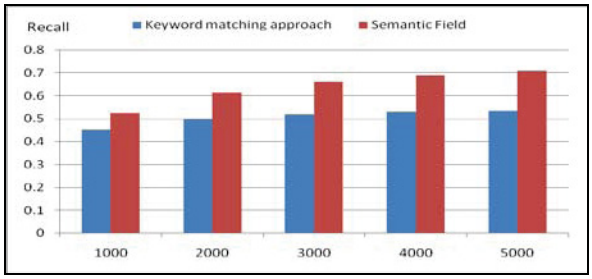


Figure 9: Comparison in recall on top-k image ranking. The x-axis indicates the value of k, while the y-axis shows the R(tag).

Figures 8 and 9 report the performance in precision and recall respectively. From the results, we find that our approach is better than that of the baseline system in both precision and recall. This is because on one hand the Semantic Field tackles the ambiguous and noisy tag problems so that we could improve the precision. On the other hand, the Semantic Field analysis includes many highly related tags, which tackle the incomplete tags problem so that it could improve the performance in recall.

4.3 Importance of Multi-source Information

Semantic Fields combine three information sources: WordNet, Wikipedia and the tag's co-occurrence information in the NUS-Wide corpus. We design the third experiment to evaluate the contribution of each information source. The results are shown in Figure 10.

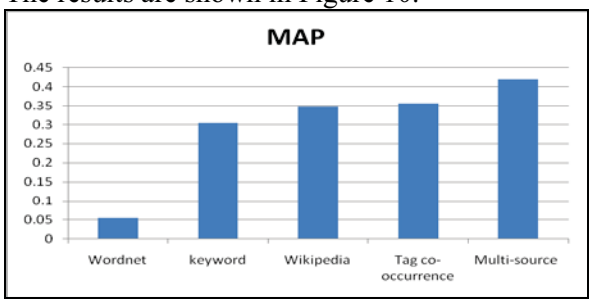


Figure 10: The comparison between using single information source and fusion of multiple information sources.

From Figure 10, we find that the performance of using WordNet alone obtains the worst result. This is because the number of tags carries the most common sense is limited and there are some noisy words in the description. For example, in Figure 2, the occurrence of the word "long" does not imply the occurrence of the concept "bear". Due to the lack of further information, using WordNet alone can hardly remove the noisy tag "long". The test result shows that such noisy information significantly degrade the performance of the system. This suggests the importance of incorperating other sources of informaiton to provide more complete information for the analysis.

We can also observe that using Wikipedia or tag co-occurrence shows comparatively better performance. This is because both information sources include abundance information for analysis. Thus, compared to the keyword-based approach, the performance of the systems shows around 17% improvement. Finally, fusing the three information sources results in the best MAP performance. This is because information from different sources complements each other and helps in reducing the effects of the noisy, ambiguous and incomplete tags.

5 Conclusion

In this paper, we proposed the use of Semantic Field to annotate web images. It could reduce the influences of noisy, ambiguous and incomplete tags so that the quality of the tags assigned to the web image can be improved. Our experiments showed that our approach is more robust and could achieve 38% improvement in MAP as compared to the learning-based and visual analysis approaches when there is sufficient text information. Also the fusion of multiple information sources could further boost the performance of the system.

The work is only the beginning. Future works include the followings. First, as multimedia data includes multiple modality features, how to fuse them to improve the performance of the system is an important problem. Second, current version of our algorithm only could identify one sense of the concept. How to distinguish among different senses of the concept is also an urgent task. Third, we will explore more semantic relations from Wordnet, Wikipedia and so on.

References

M. Ames and M. Naaman (2007), "Why We Tag: Motivations for Annotation in Mobile and online Media". In Proceedings of the SIGCHI confe-

rence on Human factors in computing systems, pp. 971 – 980.

C. F. Baker and C. J. Fillmore and J. B. Lowe (1998) "The Berkeley FrameNet Project", Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics pp. 86-90.

K. Bischoff, C. S. Firan, W. Nejdl, R. Paiu (2008), "Can All Tags be Used for Search", In Proceedings of the 17th ACM conference on Information and knowledge management, pp. 193-202.

T. S. Chua, J. H. Tang, R. C. Hong, H. J. Li, Z. P. Luo, and Y. T. Zheng (2009), "NUS-WIDE: A Real-World Web Image Database from National University of Singapore", ACM International Conference on Image and Video Retrieval.

B. M. David, A. Y. Ng and M. I. Jordan (2003), "Latent Dirichlet Allocation", Journal of Machine Learning Research 3: 993-1022.

P. Duygulu and K. Barnard (2002), "Object recognition as machine translation: learning a lexicon for a fixed image vocabulary", In Proceedings of the 7th European Conference on Computer Vision, 4: 97-112.

W. A. Gale and K. Church and D. Yarowsky (1992), "A method for disambiguating word sense in a corpus". Computers and the Humanities. 26 pp. 415-439.

S. H. Gao, L. T. Chia and X. G. Cheng, (2009) "Understanding Tag-Cloud and Visual Features for Better Annotation of Concepts in NUS-Wide DataBase" , In Proceedings of WSMC 2009.

M. F. Garrett (1992), "Lexical Retrieval Processes: Semantic Filed Effects", in Lehrer and Kittay Eds. Frames, Fields and Contrasts: New Essays in Semantic and Lexical Organization. pp. 377-396 Hillsdale: Lawrence Erlbaum.

R. E. Grandy (1992), "Semantic Fields, Prototypes, and the Lexicon", in Lehrer and Kittay Eds. Frames, Fields and Contrasts: New Essays in Semantic and Lexical Organization. pp. 103-122 Hillsdale: Lawrence Erlbaum.

T. Hofmann (1999), "Probabilitic Latent Semantic Indexing", In Proceedings of the 22rd Annual International SIGIR Conference on Research and Development in Information Retrieval.

J. Jeon, V. Lavrenko, and R. Manmatha (2003), "Automatic Image annotation and retrieval using cross-media relevance modes", In Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 119-126.

Y. Jin, L. Khan, L. Wang and M. Awad (2005), " Image Annnotations by Combining multiple Evidence & WordNet", In Proceedings of the ACM Multimedia Conference, pp. 706-715.

D. Jurafsky and J. H. Martin (2000), "Speech and language processing", published by Prentice-Hall Inc.

L. S. Kennedy, S. F. Chang and I. V. Kozintsev (2006), "To search or To Label", In Proceedings of MIR 2006, pp. 249-258.

R. M. V. Lavrenko and J. Jeon (2003), " A model for learning the semantic of pictures", In Proceedings of the 17th Annual Conference on Neural Information Processing Systems.

C. Manning and H. Schutze (1999). "Foundations of Statistical Natural Language Processing". MIT Press, Cambridge, MA.

K. Matusiak (2006), "Towards user-centered indexing in digitial image collections", OCLC systems and Services, 22(4): pp. 283-298.

R. Navigli (2009), "Word Sense Disambiguation: A Survey", ACM Computing Surveys, Vol. 41, No. 2. Article 10.

N. C. Rowe (1994) "Inferring depictions in natural language captions for efficient access to picture data", Information Process & Management Vol. 30 No 3. pp. 379-388.

G. Wang, T. S. Chua and Y. C. Wang (2003), "Extracting Key Semantic Terms from Chinese Speech Query for Web Searches". In proceeding of 41st Annual Meeting of the Association for Computational Linguistics pp. 248-255.

G. Wang, T. S. Chua, M. Zhao (2008), "Exploring Knowledge of Sub-domain in a Multi-resolution Bootstrapping Framework for Concept Detection in News Video", In Proceeding of the 16th ACM international Conference on Multimedia. pp. 249-258.

Merriam Webster Online dictionary (2010), Available at http://www.merriam-webster.com/

Automatic Extraction of Cue Phrases for Cross-Corpus Dialogue Act Classification

Nick Webb and **Michael Ferguson**
ILS Institute, SUNY Albany
nwebb@albany.edu, ferguson@cs.albany.edu

Abstract

In this paper, we present an investigation into the use of cue phrases as a basis for dialogue act classification. We define what we mean by cue phrases, and describe how we extract them from a manually labelled corpus of dialogue. We describe one method of evaluating the usefulness of such cue phrases, by applying them directly as a classifier to unseen utterances. Once we have extracted cue phrases from one corpus, we determine if these phrases are general in nature, by applying them directly as a classification mechanism to a different corpus to that from which they were extracted. Finally, we experiment with increasingly restrictive methods for selecting cue phrases, and demonstrate that there are a small number of core cue phrases that are useful for dialogue act classification.

1 Motivation

In this paper we present a recent investigation into the role of linguistic cues in dialogue act (DA) classification. Dialogue acts (Bunt, 1994) are annotations over segments of dialogue that characterise the function of those segments. Linguistic cues, which can take many forms including lexical and syntactic structures, are features that can serve as useful indicators of discourse structure (Hirschberg and Litman, 1993; Grosz and Sidner, 1986). In prior work, several researchers have shown that cue phrases can be a powerful feature for DA classification (Samuel et al., 1999; Webb et al., 2005a). Webb and Liu (2008) have previously shown that cue phrases automatically extracted from one corpus can be used to classify utterances from a new corpus. We take this approach and apply it to two established corpora with manually encoded dialogue act annotations, to investigate both the existence and the usefulness of cue phrases shared between the two corpora.

2 Related Work

In parallel with the increased availability of manually annotated dialogue corpora there has been a proliferation of literature detailing dialogue act labelling as a classification task. Prior work describes the selection of features from the corpus (including word n-grams, cue phrases, syntactic structures, dialogue history and prosodic cues) which are then passed to some machine learning algorithm. Most studies have concentrated on a single corpus, and optimised feature selection and learning algorithm accordingly. In this work we focus on two corpora, Switchboard and ICSI-MRDA, and discuss prior classification efforts relating to these two corpora.

2.1 Switchboard Corpus

The Switchboard corpus contains a large number of approximately 5-minute conversations between two people who are unknown to each other, who were asked to converse about a range of everyday topics with little or no constraint. The DA annotated portion of the Switchboard corpus (Jurafsky et al., 1997) consists of 1155 annotated conversations, containing some 225,000 utterances, of which we use 200,000 utterances, the rest being held out for separate experiments. The dialogues are annotated with a non-hierarchical variant of the DAMSL annotation scheme (Core et al., 1999). The resulting Switchboard-DAMSL annotation was a set of more than 220 distinct labels. To obtain enough data per class for statistical modelling purposes, a clustered tag set was devised, which distinguishes 42 mutually exclu-

Coling 2008: Poster Volume, pages 1310–1317,
Beijing, August 2010

sive DA types. Classification over the Switchboard corpus has been demonstrated using Decision Trees (Verbree et al., 2006), Memory-Based Learning (Rotaru, 2002) and Hidden Markov Models (HMM) (Stolcke et al., 2000). The work of Stolcke et al. (2000) is often cited as the best performing, achieving a classification accuracy of 71% over the 42 labels, although there is no cross-validation of these results. The approach of Stolcke et al. (2000) combines HMM modelling of utterances with a tri-gram model of DA sequences. Webb et al. (2005a) report a slightly lower cross-validated score (of 69%) containing an individual classification high of 72%, using an intra-utterance, cue-based classification model.

2.2 ICSI-MRDA Corpus

Like the Switchboard corpus, the ICSI Meeting Room DA (MRDA) corpus (Shriberg et al., 2004) was annotated using a variant of the DAMSL tagset, similar but not identical to the Switchboard-DAMSL annotation. The differences (and a translation between the two sets) can be seen in Shriberg et al. (2004). The underlying domain of the dialogues in the ICSI-MRDA corpus was that of multi-party meetings, with multiple participants discussing an agenda of items in a structured meeting. This application required the introduction of new tags specifically for this scenario, such as a label introduced to indicate when an utterance was used to take control of the meeting. The ICSI-MRDA corpus comprises 75 naturally occurring meetings, each around an hour in length. The section of the corpus we use consists of around 105,000 utterances. For each utterance in the corpus, one general tag was assigned, with zero or more additional specific tags. Excluding non-labelled cases, there are 11 general tags and 39 specific tags resulting in 1,260 unique dialogue acts used in the annotation. As with the Switchboard corpus, processing steps were introduced that compressed the number of unique DAs to 55. In later work, the dimensionality was further reduced, resulting in a subset of just 5 labels.

Over the ICSI-MRDA corpus, we also see DA classification efforts using Decision Trees (Verbree et al., 2006) and Memory-Based Learning (Lendvai and Geertzen, 2007), in addition to Graph Models (Ji and Bilmes, 2006) and Maximum Entropy (Ang et al., 2005). Comparatively few approaches have been applied to the 55-label annotated corpus, with most choosing to focus on the 5-label clustering, presumably for the resulting increase in score. When Ji and Bilmes (2005) apply a Graph Model to the 55 category corpus, they achieve a classification accuracy of 66%. However, when they apply the exact same method to the 5-label corpus (Ji and Bilmes, 2006), classification accuracy is boosted to 81%. The best reported classification score on the the 5-label version of the corpus is reported by Verbree et al. (2006), who achieve 89% classification accuracy by modelling the words of the utterance, the DA history and some orthographic information (such as the presence of question marks).

It remains very difficult to directly compare approaches, even when applied to the *same* corpus, so cross-corpora comparisons must be carefully considered. There are issues of the DA label set used, the labels considered and those ignored, the pre-processing of the corpus, the use of orthographic information, or prosody and so on. What seems clear is that there are no obvious leading contender for algorithm best suited to the DA classification task. Instead, we focus on the features used for DA classification.

3 Automatic Cue Extraction

When examining prior approaches, we noticed that they used a range of different features for the DA classification task, including lexical, syntactic, prosodic and dialogue context features. Most classifiers used some lexical features (the words in the utterances under consideration), frequently employing some kind of Hidden Markov Modelling to every utterance (Levin et al., 2003; Stolcke et al., 2000; Reithinger and Klesen, 1997), a technique popular in speech processing. We were inspired by the work of Samuel et al. (1999), who instead of modelling entire utterances, extract significant *cue phrases* from the VerbMobil corpus of dialogues. We use a method for cue extraction unused by Samuel et al. (1999).

What defines a good cue phrase? We are looking for words or phrases in a corpus that regularly co-occur with individual dialogue acts. We use

the term *predictivity* to indicate how predictive a phrase is of a particular DA. We want to select phrases that are highly indicative, and so concern ourselves with the highest predictivity of a particular cue phrase. We call this score the maximal predictivity. There are several other thresholds that should also be apparent. First, below some maximal predictivity score, we assume that phrases will no longer be discriminative enough to be useful for labelling DAs. Second, the number of occurrences of each phrase in the corpus as a whole is important. In their experiments, Samuel et al. (1999) constructed all n-grams of lengths 1 through 3 from the corpus, and then applied a range of measures which pruned the n-gram list until only candidate cue phrases remained. In order to test the effectiveness of these automatically acquired cue phrases, Samuel et al. (1999) passed them as features to a machine learning method, in their case transformation-based learning.

More formally, we can describe our criteria, predictivity, for selecting cue phrases from the set of all possible cue phrases in the following way. The predictivity of phrase c for DA d is the conditional probability $P(d|c)$, where:

$$P(d|c) = \frac{\#(c\&d)}{\#(c)}$$

We represent the set of all *possible* cue phrases (all n-grams length 1–4 from the corpus) as C, so given $c \in C : c$ represents some possible cue phrase. Similarly, D is the set of all dialogue act labels, and $d \in D : d$ represents some dialogue act label. Therefore $\#(c)$ is the count of (possible) cue phrase c in corpus, and $\#(c\&d)$ is the count of occurrences of phrase c in utterances with dialogue act d in the training data. The *maximal predictivity* of a cue phrase c, written as $mp(c)$, is defined as:

$$mp(c) = \max_{d \in D} P(d|c)$$

In their experiments, Samuel et al. (1999) also experimented with conditional probability, using $P(c|d)$, or the probability of some phrase occurring given some Dialogue Act. For our experiments, the word n-grams used as potential cue phrases during are automatically extracted from training data. All word n-grams of length 1–4 within the data are considered as candidates. The maximal predictivity of each cue phrase can be computed directly from the corpus. We can use this value as one threshold for pruning potential cue phrases from our model. Removing n-grams below some predictivity threshold will improve the compactness of the model produced. Another reasonable threshold would appear to be the frequency count of each potential cue phrase. Phrases which have a low frequency score are likely to have very high predictivity scores, possibly skewing the model as a whole. For example, any potential cue phrase which occurs only once will de-facto have a 100% predictivity score. We can use a minimal count value ($t_{\#}$) and minimal predictivity thresholds (t_{mp}) to prune the set C^* of 'useful' cue phrases derived from the training data, as defined by:

$$C^* = \{c \in C \mid mp(c) \geq t_{mp} \wedge \#(c) \geq t_{\#}\}$$

The n-grams that remain after this thresholding process are those we identify as cue phrases. For our initial experiments, we used a predictivity of 30% and a frequency of 2 as our thresholds for cue extraction.

4 Cue-Based DA Classification

Having defined our mechanism to extract cue phrases from a corpus, we need some way to evaluate their effectiveness. Samuel et al. (1999) passed their cue phrases as a feature to a machine learning method. We chose instead a method where the cue phrases extracted from a corpus could be used *directly* as a method of classification. If our extracted cues are indeed reliable predictors of dialogue acts, then a classifier that uses these cues directly should perform reasonably well. If, on the other hand, this mechanism did not work, it would not necessarily mean that our cue phrases are not effective, only that we need to pass them to a subsequent machine learning process as others had done. The benefit of our direct classification approach is that it is very fast to evaluate, and gives us immediate feedback as to the possible effectiveness of our automatically extracted cue phrases.

The predictivity of a cue phrase can be exploited directly in a simple model of Dialogue Act classification. We can extract potential cue phrases as described in Section 3. The resulting cue phrases selected using our measure of predictivity are then used directly to classify unseen utterances in the following manner. We identify all the potential cue phrases a target utterance contains, and determine which has the highest predictivity of some dialogue act category, then assign that category. Given the notation we define earlier, we can obtain the DA predicted by a particular cue ($dp(c)$) by:

$$dp(c) = \underset{d \in D}{\mathsf{argmax}}\, P(d|c)$$

If multiple cue phrases share the same maximal predictivity, but predict different categories, we select the DA category for the phrase which has the higher number of occurrences (that is, the n-gram with the highest frequency). If the combination of predictivity and occurrence count is insufficient to determine a single DA, then a random choice is made amongst the remaining candidate DAs. If $ng(u)$ defines the set of ngrams of length 1..4 in utterance u, and C_u^* is the set of n-grams in the utterance u that are also in the threshold model C^* then C_u^* is defined as:

$$C_u^* = ng(u) \cap C^*$$

Given our thresholds, the $mpu(u)$ (the utterance maximal prediction, or mp value for the highest scoring cue in utterance u) is defined as:

$$mpu(u) = \underset{c \in C_u^*}{\mathsf{max}}\, mp(c)$$

The maximally predictive cues of an utterance ($mpcu(u)$) are:

$$mpcu(u) = \{c \in C_u^* \mid mp(c) = mpu(u)\}$$

Then the maximal cue of utterance ($mcu(u)$), i.e. one of its maximally predictive cues that has a maximal count (from within that set), is:

$$mcu(u) = \underset{c \in mpcu(u)}{\mathsf{argmax}}\, \#(c)$$

Finally, for our classification model, $dpu(u)$ utterance DA prediction — the DA predicted by model for utterance u, is defined as:

$$dpu(u) = dp(mcu(u))$$

If no cue phrases are present in the utterance under consideration, then a default tag is assigned.

To this basic model, we added three further elaborations. The first used models sensitive to utterance length. When examining the ICSI-MRDA corpus, Ji and Bilmes (2006) found that the mean length of <STATEMENT> utterances was 8.60 words, <BACKCHANNEL> utterances were 1.04 words, <PLACE-HOLDERS> utterances were 1.31 words and <QUESTIONS> utterances were 6.50 words. Taking this as a start point, we grouped utterances into those of length 1 (i.e. short, or one word utterances), those with lengths 2–4 (we call medium length utterances), and those of length 5+ (the long length model, that comprises everything else), and produced separate cue-based models for each group.

Second, we introduced <*start*> and <*finish*> tags to each utterance (independent of the calculation of utterance length), to capture position specific information for particular cues. For example "<*start*> *okay*" identifies the occurrence of the word *'okay'* as the first word in the utterance. Finally, in the Switchboard annotation, there are other markers dealing with various linguistic issues, as outlined in Meteer (1995). A primary example is the label <+>, which indicated the presence of overlapping speech. One approach to better utilise this data is to 'reconnect' the divided utterances, i.e. appending any utterance assigned tag <+> to the last utterance by the *same* speaker. We base the selection of these model elaborations and the values for the parameters of frequency and predictivity on prior research (cf. (Webb et al., 2005a; Webb et al., 2005b; Webb et al., 2005c)).

5 Cue-Based Classification Results

Ultimately, we want to compare classification performance of a set of automatically extracted cue phrases across the two corpora, Switchboard and ICSI-MRDA. Both are annotated with similar variants of the DAMSL annotation scheme,

Condition	Cue Source	Cue Count	Accuracy
(1)	Switchboard training data	136,942	80.72%
(2)	ICSI-MRDA training data	48,856	70.78%
(3)	Intersection of Switchboard and ICSI-MRDA Training Data	25,053	72.34%
(4)	As above, discard <STATEMENT> cue phrases	577	72.62%
(5)	As above, retain only cue phrases containing <*start*> tags	242	72.52%
(6)	As above, retain only cue phrases appearing in every training intersection	148	72.09%

Table 1: Switchboard Classification Results

but there are differences. For example, the ICSI-MRDA corpus introduces several new labels that do not exist in the Switchboard annotation. Some labels in the Switchboard annotation are clustered into a single corresponding label in the ICSI-MRDA corpus, such as the two labels from Switchboard, <STATEMENT-OPINION> and <STATEMENT-NON-OPINION>, which are represented by a single label <STATEMENT> in the ICSI-MRDA corpus. To facilitate cross-corpus classification, we will cluster these labels as described in Shriberg et al. (2004). Of course, any clustering of labels has an impact on classifier performance, usually resulting in an increase. Webb et al. (2005c) indicate that clustering statement labels in the Switchboard corpus should improve performance by 8-10% percentage points.

5.1 Baseline Results

We need to establish baseline classification performance for both corpora. Our baseline for this classification task is to the most frequently occurring label for all utterances. For a number of dialogue corpora, the most frequently occurring label is some sort of statement or assertion, which is true for both the Switchboard and ICSI-MRDA corpora, where <STATEMENT> is the most frequent label. For the Switchboard corpus, selecting this label results in 51.05% accuracy. Remember that we are working with a version of the Switchboard corpus where we have clustered the original labels <STATEMENT-OPINION> and <STATEMENT-NON-OPINION> into a single label. In the original Switchboard annotation, the most frequently occurring label is <STATEMENT-NON-OPINION>, which occurs 36% of the time. Further analysis on the Switchboard corpus by Webb et al. (2005c) highlights that a significant number of <STATEMENT-OPINION> utterances in Switchboard are mislabelled as <STATEMENT-NON-OPINION> by human annotators. For the ICSI-MRDA corpus, an accuracy of 31.77% is achieved by labelling each utterance as <STATEMENT>.

Now we have established a simple baseline of performance, we want to know how well our cue-based classification method works applied to these corpora, as an evaluation of how well our cue extraction method works for each of these corpora. We ran a 10-fold stratified cross-validation exercise (referred to as Condition (1) in Tables 1 and 2) using the cue-based extraction mechanism described in Section 3, selecting cue phrases from the training data (which averaged 180k utterances for Switchboard, and 95k utterances for ICSI-MRDA), resulting in an average of 135k cue phrases from Switchboard and 50k cue phrases from ICSI-MRDA. We then applied these cue-based models to the held out test data as described in Section 4, applying Switchboard extracted cue phrases to Switchboard test data, and likewise with the ICSI-MRDA data. This establishes the best performance by our algorithm over these data sets. For Switchboard, we achieve 80.72% accuracy, as predicted by the work reported in Webb et al. (2005c). For ICSI-MRDA we obtain an accuracy of 58.14%. Remember, this model is applied to the 55-label annotated ICSI-MRDA corpus. Best reported classification accuracy for this corpus is the 66% reported by Ji and Bilmes (2005), using a graph-based model that models both utterances and sequences of DA labels. For both corpora, the cue-based model of classification outperforms the baseline, using no dialogue context whatsoever.

Condition	Cue Source	Cue Count	Accuracy
(1)	ICSI-MRDA training data	48,856	58.14%
(2)	Switchboard training data	136,942	47.07%
(3)	Intersection of Switchboard and ICSI-MRDA Training Data	25,053	47.86%
(4)	As above, discard <STATEMENT> cue phrases	577	48.05%
(5)	As above, retain only cue phrases containing <*start*> tags	242	47.30%
(6)	As above, retain only cue phrases appearing in every training intersection	148	46.34%

Table 2: ICSI-MRDA Classification Results

5.2 Cross-Corpus Results

The focus of our effort is not to maximise raw performance over individual corpora, but to examine the effectiveness of our automatically extracted cue phrases, and one mechanism to do this is to compare classification *cross-corpora*. If our cue phrases are sufficiently general predictors of DA labels across corpora, we believe that to be a powerful claim for cue phrases as a DA classification feature. Therefore, our next step was to take the cue-phrases generated from each fold of the Switchboard experiment, and apply them to the held out test data from the corresponding fold of the ICSI-MRDA experiment, and vice-versa. This is a test to see how generally applicable are the cue phrases extracted from each corpus.

When we take cues extracted from the Switchboard corpus, and apply them to the held out portion of the ICSI-MRDA corpus, we achieve an average classification accuracy (over our 10-folds) of 47.07%. This score represents 81% of the accuracy achieved by our prior result when ICSI-MRDA test data is classified using ICSI-MRDA training data. It also represents 71% of the best published score on this corpus (Ji and Bilmes, 2005). When we classify held out Switchboard test data with cue phrases extracted from the ICSI-MRDA corpus, we achieve an average classification accuracy of 70.78%, which corresponds to 88% of our best score on this corpus using Switchboard training data. These results correspond to Condition (2) in Tables 1 and 2.

These are very positive results for both directions of classification, indicating that the cue phrases we automatically extract from our corpora are generally applicable as a feature for DA classification.

5.3 Cue Phrase Reduction

We have successfully shown that we can use cue phrases extracted from one corpus to classify utterances from a different corpus. We used an inclusive approach, using all cue phrases extracted from the source corpus training data. Intuitively however, we might expect to get comparable performance by using only those cue phrases that appear in both corpora. For these intersection cue phrases, we require a strict overlap. Once the cues phrases are extracted from each individual training fold for each corpus, they are compared and retained if and only if:

- the cue phrase itself is a direct match, including any position specific label
- the DA the phrases predicts is a match
- the model number (as defined in Section 4) is a match

For each fold of our cross-validation, we take cues phrases extracted from the training data that appear in *both* corpora, pruning out cue phrases that only appear in one of the corpora. We then retain *only* those cue phrases that meet these criteria from both corpora for each specific fold, and apply them to the held out test data from that fold for each corpus.

Average classification performance for both corpora rises very slightly in comparison to using all extracted cue phrases. These results can be seen as Condition (3) in Tables 1 and 2. When applying the intersection cue phrases to the Switchboard test data, we achieve an average classification accuracy score of 72.34%. When we apply the intersection cues to the ICSI-MRDA test data, the average score is 47.86%. The average number of cue phrases that are used in this experiment

(i.e that appear in all training folds for both corpora, with matching model information) is around 25k. This represents 50% of the average number of cues extracted from the ICSI-MRDA corpus, and only 19% of the average number of cue phrases extracted from the Switchboard corpus.

We describe earlier that our default label as applied by our classifier when no cue phrase can be found is the <STATEMENT> label, the most frequent single label in both corpora. Given this, we can safely remove cue phrases that predict <STATEMENT> labels from our cue phrase set. The absence of such cue phrases should have no impact on our classification performance, but should reduce our total number of cue phrases. As can be seen in Condition (4) in Tables 1 and 2, this is indeed the case, with no statistical significance between the results with and without <STATEMENT> cue phrases. However, there is a drop in the number of cue phrases. When we remove all <STATEMENT> cue phrases from the intersection of cue phrases, we are left with an average of 577 cue phrases.

Further analysis of classifier performance indicates that a high percentage of actual labelling is performed using a subset of even the cue phrases extracted under Condition (4). We observed that cue phrases that contain a <*start*> tag (as described in Section 4) were used in the majority of cases. Our final experiment was to extract, from the 577 cue phrases, only those phrases that contain the <*start*> tag. This reduced the average number of cue phrases to 242. Classification performance remains unaffected, scoring an average of 72.52% for Switchboard and 47.30% for ICSI-MRDA, as seen in Condition (5) in the results tables. We note that of those 242 phrases, 148 appear in the intersection of *every* training fold of our 10-fold cross-validation. When we use only those 148 cue phrases for classification, as seen in Condition (6), average classification accuracy remains the same; 72.09% for Switchboard, and 46.34% for ICSI-MRDA.

6 Conclusions

In this paper, we investigate a cue-based approach to DA classification, applied to two corpora, Switchboard and ICSI-MRDA. We automatically extracted cue phrases from both corpora, and used them directly to classify unseen utterances from the corresponding corpus, demonstrating that our automatically discovered cue phrases are a sufficiently useful feature for this task.

We then explored the generality of our cue phrases, by applying them directly as a classifier to data from the alternate corpus. Whilst there was some expected drop in performance, the classification accuracy for both experiments is good, given such a small number of features and the simple design of the classifier. The result indicates that cue phrases are a highly useful feature for DA classification, and can be used to classify data from new corpora, possibly as some part of some quasi-automatic first annotation effort.

We experimented with reducing the set of cue phrases, using increasingly restrictive measures of retaining our automatically discovered cues. We found that we did not have a drop in performance compared to the cross-corpus classification accuracy, even when the cue set is drastically reduced (to 0.001% of the original Switchboard cue phrases, and 0.003% of the ICSI-MRDA cue phrases). This appears to be a strong indicator of the discriminative power of some small number of automatically discovered core cue phrases.

References

Ang, J., Y. Liu, and E. Shriberg. 2005. Automatic Dialog Act Segmentation and Classification in Multiparty Meetings. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pages 1061–1064, Philadelphia.

Bunt, H. 1994. Context and Dialogue Control. *THINK*, 3:19–31.

Core, M., M. Ishizaki, J. Moore, and C. Nakatani. 1999. The Report of the Third Workshop of the Discourse Resource Initiative. *Chiba University and Kazusa Academia Hall*.

Grosz, B. and C. Sidner. 1986. Attention, Intentions, and the Structure of Discourse. *Computational Linguistics*, 19(3).

Hirschberg, J. and D. Litman. 1993. Empirical Studies on the Disambiguation of Cue Phrases. *Computational Linguistics*, 19(3):501–530.

Ji, G. and J. Bilmes. 2005. Dialog Act Tagging Using Graphical Models. In *Proceedings of the IEEE*

International Conference on Acoustics, Speech, and Signal Processing, Philadelphia, PA.

Ji, G. and J. Bilmes. 2006. Backoff Model Training using Partially Observed Data: Application to Dialog Act Tagging. In *Proceedings of the Human Language Technology/ American chapter of the Association for Computational Linguistics(HLT/NAACL'06)*.

Jurafsky, D., R. Bates, N. Coccaro, R. Martin, M. Meteer, K. Ries, E. Shriberg, A. Stolcke, P. Taylor, and C. Van Ess-Dykema. 1997. Automatic Detection of Discourse Structure for Speech Recognition and Understanding. In *Proceedings of the 1997 IEEE Workshop on Speech Recognition and Understanding, Santa Barbara.*

Lendvai, P. and J. Geertzen. 2007. Token-Based Chunking of Turn-Internal Dialogue Act Sequences. In *Proceedings of the 8th SIGDIAL Workshop on Discourse and Dialogue*, pages 174–181, Antwerp, Belgium.

Levin, L., C. Langley, A. Lavie, D. Gates, and D. Wallace. 2003. Domain Specific Speech Acts for Spoken Language Translation. In *Proceedings of 4th SIGdial Workshop on Discourse and Dialogue*.

Meteer, M. 1995. Dysfluency Annotation Stylebook for the Switchboard Corpus. Working paper, Linguistic Data Consortium.

Reithinger, N. and M. Klesen. 1997. Dialogue Act Classification Using Language Models. In *Proceedings of EuroSpeech-97*.

Rotaru, M. 2002. Dialog Act Tagging using Memory-Based Learning. Term project, University of Pittsburgh.

Samuel, K., S. Carberry, and K. Vijay-Shanker. 1999. Automatically Selecting Useful Phrases for Dialogue Act Tagging. In *Proceedings of the Fourth Conference of the Pacific Association for Computational Linguistics, Waterloo, Ontario, Canada*.

Shriberg, E., R. Dhillon, S. Bhagat, J. Ang, and H. Carvey. 2004. The ICSI Meeting Recorder Dialog Act (MRDA) Corpus. In *Special Interest Group on Discourse and Dialogue (SIGdial)*, Boston, USA.

Stolcke, A., K. Ries, N. Coccaro, E. Shriberg, R. Bates, D. Jurafsky, P. Taylor, R. Martin, C. Van Ess-Dykema, and M. Meteer. 2000. Dialogue Act Modeling for Automatic Tagging and Recognition of Conversational Speech. In *Computational Linguistics 26(3), 339–373.*

Verbree, D., R. Rienks, and D. Heylen. 2006. Dialogue Act Tagging using Smart Feature Selection; Results on Multiple Corpora. *Spoken Language Technology Workshop, 2006. IEEE*, pages 70–73.

Webb, N. and T. Liu. 2008. Investigating the Portability of Corpus-Derived Cue Phrases for Dialogue Act Classification. In *Proceedings of the 22nd International Conference on Computational Linguistics (COLING 2008)*, Manchester, United Kingdom.

Webb, N., M. Hepple, and Y. Wilks. 2005a. Dialogue Act Classification Based on Intra-Utterance Features. In *Proceedings of the AAAI Workshop on Spoken Language Understanding*, at the Twentieth National Conference on Artificial Intelligence, Pittsburgh, PA.

Webb, N., M. Hepple, and Y. Wilks. 2005b. Empirical Determination of Thresholds for Optimal Dialogue Act Classification. In *Proceedings of the Ninth Workshop on the Semantics and Pragmatics of Dialogue*.

Webb, N., M. Hepple, and Y. Wilks. 2005c. Error Analysis of Dialogue Act Classification. In *Proceedings of the 8th International Conference on Text, Speech and Dialogue*, Carlsbad, Czech Republic.

Search with Synonyms: Problems and Solutions

Xing Wei, Fuchun Peng, Huishin Tseng, Yumao Lu, Xuerui Wang, Benoit Dumoulin

Yahoo! Labs at Sunnyvale

{xwei,fuchun,huihui,yumaol,xuerui,benoitd}@yahoo-inc.com

Abstract

Search with synonyms is a challenging problem for Web search, as it can easily cause intent drifting. In this paper, we propose a practical solution to this issue, based on co-clicked query analysis, i.e., analyzing queries leading to clicking the same documents. Evaluation results on Web search queries show that synonyms obtained from this approach considerably outperform the thesaurus based synonyms, such as WordNet, in terms of keeping search intent.

1 Introduction

Synonym discovery has been an active topic in a variety of language processing tasks (Baroni and Bisi, 2004; Fellbaum, 1998; Lin, 1998; Pereira et al., 1993; Sanchez and Moreno, 2005; Turney, 2001). However, due to the difficulties of synonym judgment (either automatically or manually) and the uncertainty of applying synonyms to specific applications, it is still unclear how synonyms can help Web scale search task. Previous work in Information Retrieval (IR) has been focusing mainly on related words (Bai et al., 2005; Wei and Croft, 2006; Riezler et al., 2008). But Web scale data handling needs to be precise and thus synonyms are more appropriate than related words for introducing less noise and alleviating the efficiency concern of query expansion. In this paper, we explore both manually-built thesaurus and automatic synonym discovery, and apply a three-stage evaluation by separating synonym accuracy from relevance judgment and user experience impact.

The main difficulties of discovering synonyms for Web search are the following:

1. Synonym discovery is context sensitive. Although there are quite a few manually built thesauri available to provide high quality synonyms (Fellbaum, 1998), most of these synonyms have the same or nearly the same meaning only in some senses. If we simply replace them in search queries in all occurrences, it is very easy to trigger search intent drifting. Thus, Web search needs to understand different senses encountered in different contexts. For example, "baby" and "infant" are treated as synonyms in many thesauri, but "Santa Baby" has nothing to do with "infant". "Santa Baby" is a song title, and the meaning of "baby" in this entity is different than the usual meaning of "infant".

2. Context can not only limit the use of synonyms, but also broaden the traditional definition of synonyms. For instance, "dress" and "attire" sometimes have nearly the same meaning, even though they are not associated with the same entry in many thesauri; "free" and "download" are far from synonyms in traditional definition, but "free cd rewriter" may carry the same query intent as "download cd rewriter".

3. There are many new synonyms developed from the Web over time. "Mp3" and "mpeg3" were not synonyms twenty years ago; "snp newspaper" and "snp online" carry the same query intent only after snponline.com was published. Manually editing synonym list is prohibitively expensive. Thus, we need an automatic synonym discovery system that can learn from huge amount of data and update the dictionary frequently.

Coling 2008: Poster Volume, pages 1318–1326,
Beijing, August 2010

In summary, synonym discovery for Web search is different from traditional thesaurus mining; it needs to be context sensitive and needs to be updated timely. To address these problems, we conduct context based synonym discovery from co-clicked queries, i.e., queries that share similar document click distribution. To show the effectiveness of our synonym discovery method on Web search, we use several metrics to demonstrate significant improvements: (1) synonym discovery accuracy that measures how well it keeps the same search intent; (2) relevance impact measured by Discounted Cumulative Gain (DCG) (Jarvelin and Kekalainen., 2002); and (3) user experience impact measured by online experiment.

The rest of the paper is organized as follows. In Section 2, we first discuss related work and differentiate our work from existing work. Then we present the details of our synonym discovery approach in Section 3. In Section 4 we show our query rewriting strategy to include synonyms in Web search. We conduct experiments on randomly sampled Web search queries and run the three-stage evaluation in Section 5 and analyze the results in Section 6. WordNet based synonym reformulation and a current commercial search engine are the baselines for the three-stage evaluation respectively. Finally we conclude the paper in Section 7.

2 Related Works

Automatically discovering synonyms from large corpora and dictionaries has been popular topics in natural language processing (Sanchez and Moreno, 2005; Senellart and Blondel, 2003; Turney, 2001; Blondel and Senellart, 2002; van der Plas and Tiedemann, 2006), and hence, there has been a fair amount of work in calculating word similarity (Porzel and Malaka, 2004; Richardson et al., 1998; Strube and Ponzetto, 2006; Bollegala et al., 2007) for the purpose of discovering synonyms, such as information gain on ontology (Resnik, 1995) and distributional similarity (Lin, 1998; Lin et al., 2003). However, the definition of synonym is application dependent and most of the work has been applied to a specific task (Turney, 2001) or restricted in one domain (Baroni and Bisi, 2004). Synonyms extracted using these traditional approaches cannot be easily adopted in Web search where keeping search intent is critical.

Our work is also related to semantic matching in IR: manual techniques such as using handcrafted thesauri and automatic techniques such as query expansion and clustering all attempts to provide a solution, with varying degrees of success (Jones, 1971; van Rijsbergen, 1979; Deerwester et al., 1990; Liu and Croft, 2004; Bai et al., 2005; Wei and Croft, 2006; Cao et al., 2007). These works focus mainly on adding in loosely semantically related words to expand literal term matching. But related words may be too coarse for Web search considering the massive data available.

3 Synonym Discovery based on Co-clicked Queries

In this section, we discuss our approach to synonym discovery based on co-clicked queries in Web search in detail.

3.1 Co-clicked Query Clustering

Clustering has been extensively studied in many applications, including query clustering (Wen et al., 2002). One of the most successful techniques for clustering is based on distributional clustering (Lin, 1998; Pereira et al., 1993). We adopt a similar approach to our co-clicked query clustering. Each query is associated with a set of clicked documents, which in turn associated with the number of views and clicks. We then compute the distance between a pair of queries by calculating the Jensen-Shannon(JS) divergence (Lin, 1991) between their clicked URL distributions. We start with that every query is a separate cluster, and merge clusters greedily. After clusters are generated, pairs of queries within the same cluster can be considered as co-clicked/related queries with a similarity score computed from their JS divergence.

$$Sim(q_k|q_l) = D_{JS}(q_k||q_l) \quad (1)$$

3.2 Query Pair Alignment

To make sure that words are replacement for each other in the co-clicked queries, we align words in the co-clicked query pairs that have the same length (number of terms), and have the same terms for all positions except one. This is a simplification for complicated aligning processes. Previous work on machine translation (Brown et al., 1993) can be used when complete alignment is needed for modeling. However, as we have tremendous amount of co-clicked query data, our restricted version of alignment is sufficient to obtain a reasonable number of synonyms. In addition, this restricted approach eliminates much noise introduced in those complicated aligning processes.

3.2.1 Synonym Discovery from Co-clicked Query Pair

Synonyms discovered from co-clicked queries have two aspects of word meaning: (1) general meaning in language and (2) specific meaning in the query. These two aspects are related. For example, if two words are more likely to carry the same meaning in general, then they are more likely to carry the same meaning in specific queries; on the other hand, if two words often carry the same meaning in a variety of specific queries, then we tend to believe that the two words are synonyms in general language. However, neither of these two aspects can cover the other. Synonyms in general language may not be used to replace each other in a specific query. For example, "sea" and "ocean" have nearly the same meaning in language, but in the specific query "sea boss boat", "sea" and "ocean" cannot be treated as synonyms because "sea boss" is a brand; also, in the specific query "women's wedding attire", "dress" can be viewed as a synonym to "attire", but in general language, these two words are not synonyms. Therefore, whether two words are synonyms or not for a specific query is a synthesis judgment based on both of general meaning and specific context.

We develop a three-step process for synonym discovery based on co-clicked queries, considering the above two aspects.

Step 1: Get all synonym candidates for word w_i in general meaning.

In this step, we would like to get all synonym candidates for a word. This step corresponds to Aspect (1) to catch the general meaning of words in language. We consider all the co-clicked queries with the word and sum over them, as in Eq. 2

$$P(w_j|w_i) = \frac{\sum_k sim_k(w_i \to w_j)}{\sum_{w_j} \sum_k sim(w_i \to w_j)} \quad (2)$$

where $sim_k(w_i \to w_j)$ represents the similarity score (see Section 3.1) of a query q_k that aligns w_i to w_j. So intuitively, we aggregate scores of all query pairs that align w_i to w_j, and normalize it to a probability over the vocabulary.

Step 2: Get synonyms for word w_i in query q_k.

In this step, we would like to get synonyms for a word in a specific query. We define the probability of reformulating w_i with w_j for query q_k as the similarity score shown in Eq. 3.

$$P\left(w_j|w_i, q_k\right) = sim_k(w_i \to w_j) \quad (3)$$

Step 3: Combine the above two steps.

Now we have two sets of estimates for the synonym probability, which is used to reformulate w_i with w_j. One set of values are based on general language information and another set of values are based on specific queries. We apply three combination approaches to integrate the two sets of values for a final decision of synonym discovery: (1) two independent thresholds for each probability, (2) linear combination with a coefficient, and (3) linear combination in log scale as in Eq. 4, with λ as a mixture coefficient.

$$\begin{aligned} P_{q_k}(w_j|w_i) \propto \lambda \log P(w_j|w_i) \\ +(1-\lambda) \log P(w_j|w_i, q_k) \end{aligned} \quad (4)$$

In experiments we found that there is no significant difference with the results from different combination methods by finely tuned parameter setting.

3.2.2 Concept based Synonyms

The simple word alignment strategy we used can only get the synonym mapping from single

term to single term. But there are a lot of phrase-to-phrase, term-to-phrase, or phrase-to-term synonym mappings in language, such as “babe in arms” to “infant”, and “nyc” to ”new york city”. We perform query segmentation on queries to identify concept units from queries based on an unsupervised segmentation model (Tan and Peng, 2008). Each unit is a single word or several consecutive words that represent a meaningful concept.

4 Synonym Handling in Web Search

The automatic synonym discovery methods described in Section 3 generate synonym pairs for each query. A simple and straightforward way to use the synonym pairs would be “equalizing” them in search, just like the “OR” function in most commercial search engines.

Another method would be to re-train the whole ranking system using the synonym feature, but it is expensive and requires a large size training set. We consider this to be future work.

Besides general equalization in all cases, we also apply a restriction, specially, on whether or not to allow synonyms to participate in document selection. For the consideration of efficiency, most Web search engines has a document selection step to pre-select a subset of documents for full ranking. For the general equalization, the synonym pair is treated as the same even in the document selection round; in a conservative variation, we only use the original word for document selection but use the synonyms in the second phase finer ranking.

5 Experiments

In this section, we present the experimental results for our approaches with some in-depth discussion.

5.1 Evaluation Metrics

We have several metrics to evaluate the synonym discovery system for Web search queries. They corresponds to the three stages during the system development. Each of them measures a different aspect.

Stage 1: accuracy. Because we are more interested in the application of reformulating Web search queries, our guideline to the editorial judgment focuses on the query intent change and context-based synonyms. For example, “transporters” and “movers” are good synonyms in the context of “boat” because “boat transporters” and “boat movers” keep the same search intent, but “ocean” is not a good synonym to “sea” in the query of “sea boss boats” because “sea boss” is a brand name and “ocean boss” does not refer to the same brand. Results are measured with accuracy by the number of discovered synonyms (which reflects coverage).

Stage 2: relevance. To evaluate the effectiveness of our semantic features we use DCG, a widely-used metric for measuring Web search relevance.

Stage 3: user experience. In addition to the search relevance, we also evaluate the practical user experience after logging all the user search behaviors during a two-week online experiment.

Web CTR: the Web click through rate (Sherman and Deighton, 2001; Lee et al., 2005) is defined as

$$CTR = \frac{\text{number of clicks}}{\text{total page views}},$$

where a page view (PV) is one result page that a search engine returns for a query.

Abandon rate: the percentage of queries that are abandoned by user neither clicking a result nor issuing a query refinement.

5.2 Data

A period of Web search query log with clicked URLs are used to generate co-clicked query set. After word alignment that extracts the co-clicked query pairs with same number of units and with only one different unit, we obtain 12.1M unsegmented query pairs and 11.9M segmented query pairs.

Since we run a three-stage evaluation, there are three independent evaluation set respectively:

1. accuracy test set. For the evaluation of synonym discovery accuracy, we randomly sampled 42K queries from two weeks of query log, and

evaluate the effectiveness of our synonym discovery model with these queries. To test the synonym discovery model built on the segmented data, we segment the queries before using them as evaluation set.

2. relevance test set. To evaluate the relevance impact by the synonym discovery approach, we run experiments on another two weeks of query log and randomly sampled 1000 queries from the affected queries (queries that have differences in the top 5 results after synonym handling).

3. user experience test set. The user experience test is conducted online with a commercial search engine.

5.3 Results of Synonym Discovery Accuracy

Here we present the results of WordNet thesaurus based query synonym discovery, co-clicked based term-to-term query synonym discovery, and co-click concept based query synonym discovery.

5.3.1 Thesaurus-based Synonym Replacement

The WordNet thesaurus-based synonym replacement is a baseline here. For any word that has synonyms in the thesaurus, thesaurus-based synonym replacement will rewrite the word with synonyms from the thesaurus.

Although thesaurus often provides clean information, synonym replacement based on thesaurus does not consider query context and introduces too many errors and noise. Our experiments show that only **46%** of the discovered synonyms are correct synonyms in query. The accuracy is too low to be used for Web search queries.

5.3.2 Co-clicked Query-based Context Synonym Discovery

Here we present the results from our approach based on co-clicked query data (in this section the queries are all original queries without segmentation). Figure 1 shows the accuracy of synonyms by the number of discovered synonyms. By applying different thresholds as cut-off lines to Eq. 4, we get different numbers of synonyms from the same test set. As we can see, loosening the threshold can give us more synonym pairs, but it could hurt the accuracy.

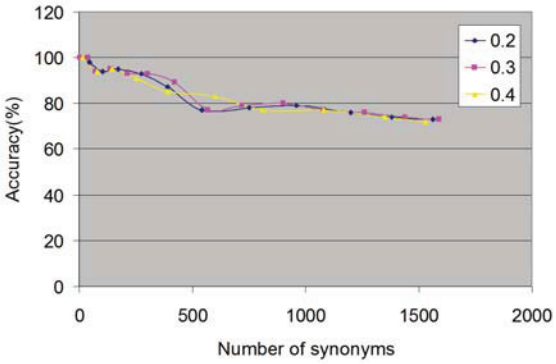


Figure 1: Accuracy versus number of synonyms with term based synonym discovery

Figure 1 demonstrates how accuracy changes with the number of synonyms. Y-axis represents the percentage of correctly discovered synonyms, and X-axis represents the number of discovered synonyms, including both of correct ones and wrong ones. The three different lines represents three different parameter settings of mixture weights (λ in Eq. 4, which is 0.2, 0.3, or 0.4 in the figure). The figure shows accuracy drops by increasing the number of synonyms. More synonym pairs lead to lower accuracy.

From Figure 1 we can see: Firstly, three curves with different thresholds almost overlap, which means the effectiveness of synonym discovery is not very sensitive to the mixture weight. Secondly, accuracy is monotonically decreasing as more synonyms are detected. By getting more synonyms, the accuracy decreases from **100%** to less than **80%** (we are not interested in accuracies lower than 80% due to the high precision requirement of Web search tasks, so the graph contains only high-accuracy results). This trend also confirms the effectiveness of our approach (the accuracy for a random approach would be a constant).

5.3.3 Concept based Context Synonym Discovery

We present results from our model based on segmented co-clicked query data in this section.

Original Query	New Query with Synonyms	Intent
Examples of thesaurus-based based synonym replacement		
basement window wells drainage billabong boardshorts sale bigger stronger faster documentary	basement window wells drain billabong boardshorts sales event larger stronger faster documentary	same
yahoo maryland judiciary case search free cell phone number lookup	hayseed maryland judiciary pillowcase search free cell earpiece number lookup	different
Examples of term-to-term synonym discovery		
airlines jobs area code finder acai berry	airlines careers area code search acai fruit	same
acai berry ace crest toothpaste coupon	acai juice hardware crest whitestrips coupon	different
Examples of concept based synonym discovery		
ae apartments_for_rent arizona time_zone	american_eagle outfitters apartment_rentals arizona time	same
cortrust bank credit_card david_beckham dodge_caliber	cortrust bank mastercard beckham dodge	different

Table 1: Examples of query synonym discovery: the first section is thesaurus based, second section is co-clicked data based term-to-term synonym discovery, and the last section is concept based synonym discovery.

The modeling part is the same as the one for Section 5.3.2, and the only difference is that the data were segmented. We have shown in Section 5.3.2 that the mixture weight is not an crucial factor within a reasonable range, so we present only the result with one mixture weight in Figure 2. As in Section 5.3.2, the figure shows that the accuracy of synonym discovery is sensitive to the threshold. It confirms that our model is effective and setting threshold to Eq. 4 is a feasible and sound way to discover not only single term synonyms but also phrase synonyms.

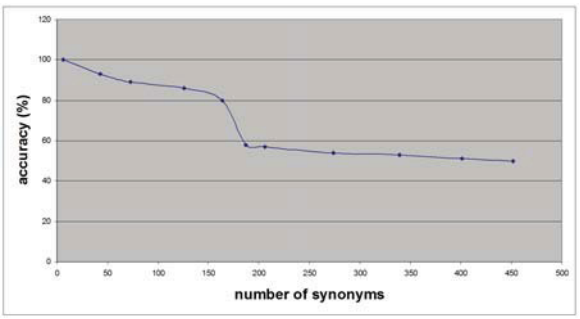


Figure 2: Accuracy versus number of synonyms with concept based synonym discovery

Table 1 shows some anecdotal examples of query synonyms with the thesaurus-based synonym replacement, context sensitive synonym discovery, and concept based context sensitive synonym discovery. In contrast, the upper part of each section shows positive examples (query intents remain the same after synonym replacement) and the lower part shows negative examples (query intents change after synonym replacement).

5.4 Results of Relevance Impact

We run relevance test on 1000 randomly sampled affected queries. With the automatic synonym discovery approach we apply our synonym handling method described in Section 4. Results of DCG improvements by different thresholds and synonym handling settings are presented in Table 2. Thresholds are selected empirically from the accuracy test in Section 5.3 (we run a small size relevance test on the accuracy test set and set the range of thresholds based on that). Note that in our relevance experiments we use term-to-term synonym pairs only. For the relevance impact of concept-based synonym discovery, we would like to study it in our future work.

From Table 2 we can see that the automatic synonym discovery approach we presented significantly improves search relevance on various settings, which confirms the effectiveness of our synonym discovery for Web search queries. We conjecture that avoiding synonym in document selection is of help. This is because precision is more important to Web search than recall for the huge amount of data available on the Web.

Relevance impact with synonym handling			
threshold1	threshold2	doc-selection participation	DCG
0.8	0.02	no	**+1.7%**
0.8	0.02	yes	**+1.3%**
0.8	0.05	no	**+1.8%**
0.8	0.05	yes	**+1.4%**

Table 2: Relevance impact with synonym handling by different parameter settings. "Threshold1" is the threshold for context-based similarity score–Eq. 3; "threshold2" is the threshold for general case similarity score–Eq. 2; "doc-selection participation" refers to whether or not let synonym handling participate in document selection. All improvements are statistically significant by Wilcox significance test.

5.5 Results of User Experience Impact

In addition to the relevance impact, we also evaluated the practical user experience impact by CTR and abandon rate (defined in Section 5.1) through a two-week online run. Results show that the synonym discovery method presented in this paper improves Web CTR by **2%**, and decreases abandon rate by **11.4%**. All changes are statistically significant, which indicates synonyms are indeed beneficial to user experience.

6 Discussion and Error Analysis

From Table 1, we can see that our approach can catch not only traditional synonyms, which are the synonyms that can be found in manually-built thesaurus, but also context-based synonyms, which may not be treated as synonyms in a standard dictionary or thesaurus. There are a variety of synonyms our approach discovered:

1. Synonyms that are not considered as synonyms in traditional thesaurus, such as "berry" and "fruit" in the context of "acai". "acai berry" and "acai fruit" refer to the same fruit.

2. Synonyms that have different part-of-speech features than the corresponding original words, such as "finder" and "search". Users searching "area code finder" and users searching "area code search" are looking for the same content. In the context of Web search queries, part-of-speech is not an important factor as most queries are not grammatically perfect.

3. Synonyms that show up in recent concepts, such as "webmail" and "email" in the context of "cox". The new concept of "webmail" or "email" has not been added to many thesauri yet.

4. Synonyms not limited by length, such as "crossword puzzles" and "crossword", "homes for sale" and "real estate". The segmenter helps our system discover synonyms in various lengths.

With these many variations, the synonyms discovered by our approach are not the "synonyms" in the traditional meaning. They are context sensitive, Web data oriented and search effective synonyms. These synonyms are discovered by the statistical model we presented and based on Web search queries and clicked data.

However, the click data themselves contain a huge amount of noise. Although they can reflect the users' intents in some big picture, in many specific cases synonyms discovered from co-clicked data are biased by the click noise. In our application—Web search query reformulation with synonyms, accuracy is the most important thing and thus we are interested in error analysis. The errors that our model makes in synonym discovery are mainly caused by the following reasons:

(1) There are some concepts well accepted such as "cnn" means "news" and "amtrak" means "train". And users searching "news" tend to click CNN Web site; users searching "train" tend to click Amtrak Web site. With our model, "cnn" and "news", "amtrak" and "train" are discovered to be synonyms, which may hurt the search of "news" or "train" in general meaning.

(2) Same clicks by different intents. Although clicking on same documents generally indicates same search intent, different intents could result in same or similar clicks, too. For example, the queries of "antique style wedding rings" and "antique style engagement rings" carry different intents, but very usually, these two different intents lead to the clicks on the same Web site. "Booster seats" and "car seats", "brighton handbags" and "brighton shoes" are other two examples in the same case. For these examples, clicking on Web URLs are not precise enough to reflect the subtle difference of language concepts.

(3) Bias from dominant user intents. Most people searching "apartment" are looking for an apartment to rent. So "apartment for rent" and "apartment" have similar clicked URLs. But these two are not synonyms in language. In these cases, popular user intents dominate and bias the meaning of language, which causes problems. "Airline baggage restrictions" and "airline travel restrictions" is another example.

(4) Antonyms. Many context-based synonym discovery methods suffer from the antonym problem, because antonyms can have very similar contexts. In our model, the problem has been reduced by integrating clicked-URLs. But still, there are some examples, such as "spyware" and "antispyware", resulting in similar clicks. To learn how to "protect a Web site", a user often needs to learn what are the main methods to "attack a Web site", and these different-intent pairs lead to the same clicks because different intents do not have to mean different interests in many specific cases.

Although these problems are not common, but when they happen, they cause a bad user search experience. We believe a solution to these problems might need more advanced linguistic analysis.

7 Conclusions

In this paper, we have developed a synonym discovery approach based on co-clicked query data, and improved search relevance and user experience significantly based on the approach.

For future work, we are investigating more synonym handling methods to further improve the synonym discovery accuracy, and to handle the discovered synonyms in more ways than just the query side.

References

Bai, J., D. Song, P. Bruza, J.Y. Nie, and G. Cao. 2005. Query Expansion using Term Relationships in Language Models for Information Retrieval. In *Proceedings of the ACM 14th Conference on Information and Knowledge Management*.

Baroni, M. and S. Bisi. 2004. Using Cooccurrence Statistics and the Web to Discover Synonyms in a Technical Language. In *LREC*.

Blondel, V. and P. Senellart. 2002. Automatic Extraction of Synonyms in a Dictionary. In *Proc. of the SIAM Workshop on Text Mining*.

Bollegala, D., Y. Matsuo, and M. Ishizuka. 2007. Measuring Semantic Similarity between Words using Web Search Engines. In *Proceedings of the 16th international conference on World Wide Web (WWW)*.

Brown, P. F., S. A. Della Pietra, V. J. Della Pietra, and R. L. Mercer. 1993. The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics*, 19(2):263.

Cao, G., J.Y. Nie, and J. Bai. 2007. Using Markov Chains to Exploit Word Relationships in Information Retrieval. In *Proceedings of the 8th Conference on Large-Scale Semantic Access to Content*.

Deerwester, S., S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. 1990. Indexing by Latent Semantic Analysis. *Journal of the American Society for Information Science*, 41(6):391–407.

Fellbaum, C., editor. 1998. *WordNet: An Electronic Lexical Database.* MIT Press, Cambridge, Mass.

Jarvelin, K. and J. Kekalainen. 2002. Cumulated Gain-Based Evaluation Evaluation of IR Techniques. *ACM TOIS*, 20:422–446.

Jones, K. S., 1971. *Automatic Keyword Classification for Information Retrieval*. London: Butterworths.

Lee, Uichin, Zhenyu Liu, and Junghoo Cho. 2005. Automatic Identification of User Goals in Web Search. In *In the World-Wide Web Conference (WWW)*.

Lin, D., S. Zhao, L. Qin, and M. Zhou. 2003. Identifying Synonyms among Distributionally Similar Words. In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*.

Lin, J. 1991. Divergence measures based on the shannon entropy. *IEEE Transactions on Information Theory*, 37(1):145–151.

Lin, D. 1998. Automatic Retrieval and Clustering of Similar Words. In *Proceedings of COLING/ACL-98*, pages 768–774.

Liu, X. and B. Croft. 2004. Cluster-based Retrieval using Language Models. In *Proceedings of SIGIR*.

Pereira, F., N. Tishby, and L. Lee. 1993. Distributional Clustering of English Words. In *Proceedings of ACL*, pages 183 – 190.

Porzel, R. and R. Malaka. 2004. A Task-based Approach for Ontology Evaluation. In *ECAI Workshop on Ontology Learning and Population*.

Resnik, P. 1995. Using Information Content to Evaluate Semantic Similarity in a Taxonomy. In *Proceedings of IJCAI-95*, pages 448 – 453.

Richardson, S., W. Dolan, and L. Vanderwende. 1998. MindNet: Acquiring and Structuring Semantic Information from Text. In *36th Annual meeting of the Association for Computational Linguistics*.

Riezler, Stefan, Yi Liu, and Alexander Vasserman. 2008. Translating Queries into Snippets for Improved Query Expansion. In *Proceedings of the 22nd International Conference on Computational Linguistics (COLING'08)*.

Sanchez, D. and A. Moreno. 2005. Automatic Discovery of Synonyms and Lexicalizations from the Web. In *Proceedings of the 8th Catalan Conference on Artificial Intelligence*.

Senellart, P. and V. D. Blondel. 2003. Automatic Discovery of Similar Words. In Berry, M., editor, *A Comprehensive Survey of Text Mining*. Springer-Verlag, New York.

Sherman, L. and J. Deighton. 2001. Banner advertising: Measuring effectiveness and optimizing placement. *Journal of Interactive Marketing*, 15(2):60–64.

Strube, M. and S. P. Ponzetto. 2006. WikiRelate! Computing Semantic Relatedness Using Wikipedia. In *Proceedings of AAAI*.

Tan, B. and F. Peng. 2008. Unsupervised Query Segmentation using Generative Language Models and Wikipedia. In *Proceedings of the 17th International World Wide Web Conference (WWW)*, pages 347–356.

Turney, P. 2001. Mining the Web for Synonyms: PMI-IR versus LSA on TOEFL. In *Proceedings of the Twelfth European Conference on Machine Learning*.

van der Plas, Lonneke and Jorg Tiedemann. 2006. Finding Synonyms using Automatic Word Alignment and Measures of Distributional Similarity. In *Proceedings of the COLING/ACL 2006*, pages 866–873.

van Rijsbergen, C.J., 1979. *Information Retrieval*. London: Butterworths.

Wei, X. and W. B. Croft. 2006. LDA-based Document Models for Ad-hoc Retrieval. In *Proceedings of SIGIR*, pages 178–185.

Wen, J.R., J.Y. Nie, and H.J. Zhang. 2002. Query Clustering Using User Logs. *ACM Transactions on Information Systems*, 20(1):59–81.

MIEA: a Mutual Iterative Enhancement Approach for Cross-Domain Sentiment Classification

Qiong Wu[1,2], Songbo Tan[1], Xueqi Cheng[1] and Miyi Duan[1]

[1]Institute of Computing Technology, Chinese Academy of Sciences
[2] Graduate University of Chinese Academy of Sciences
{wuqiong,tansongbo}@software.ict.ac.cn

Abstract

Recent years have witnessed a large body of research works on cross-domain sentiment classification problem, where most of the research endeavors were based on a supervised learning strategy which builds models from only the labeled documents or only the labeled sentiment words. Unfortunately, such kind of supervised learning method usually fails to uncover the full knowledge between documents and sentiment words. Taking account of this limitation, in this paper, we propose an iterative reinforcement learning approach for cross-domain sentiment classification by simultaneously utilizing documents and words from both source domain and target domain. Our new method can make full use of the reinforcement between documents and words by fusing four kinds of relationships between documents and words. Experimental results indicate that our new method can improve the performance of cross-domain sentiment classification dramatically.

1 Introduction

Sentiment classification is the task of determining the opinion (e.g., negative or positive) of a given document. In recent years, it has drawn much attention with the increasing reviewing pages and blogs etc., and it is very important for many applications, such as opinion mining and summarization (e.g., (Ku et al., 2006; McDonald et al., 2007)).

In most cases, a variety of supervised classification methods can perform well in sentiment classification. This kind of methods requires a condition to guarantee the accuracy of classification: training data should have the same distribution with test data so that test data could share the information got from training data. So the labeled data in the same domain with test data is considered as the most valuable resources for the sentiment classification. However, such resources in different domains are very imbalanced. In some traditional domains or domains of concern, many labeled sentiment data are freely available on the web, but in other domains, labeled sentiment data are scarce and it involves much human labor to manually label reliable sentiment data. The challenge is how to utilize labeled sentiment data in one domain (that is, source domain) for sentiment classification in another domain (that is, target domain). This raises an interesting task, cross-domain sentiment classification (or sentiment transfer). In this work, we focus on one typical kind of sentiment transfer problem, which utilizes only training data from source domain to improve sentiment classification performance for target domain, without any labeled data for the target domain (e.g., (Andreevskaia and Bergler, 2008)).

In recent years, some studies have been conducted to deal with sentiment transfer problems. However, most of the attempts rely on only the labeled documents (Aue and Gamon, 2005; Tan et al., 2007; Tan et al., 2009; Wu et al., 2009) or the labeled sentiment words (Gamon and Aue, 2005) to improve the performance of sentiment transfer, so this kind of methods fails to uncover the full knowledge between the documents and the sentiment words.

In fact, the opinion of a document can be determined by the interrelated documents as well as by the interrelated words, and this rule is also tenable when determining the opinion of a sentiment word. This rule is based on the following intuitive observations:

(1) A document strongly linked with other positive (negative) documents could be considered as positive (negative); in the same way, a word strongly linked with other positive (negative) words could be considered as positive (negative).

Coling 2008: Poster Volume, pages 1327–1335,
Beijing, August 2010

(2) A document containing many positive (negative) words could be considered as positive (negative); similarly, a word appearing in many positive (negative) documents could be considered as positive (negative).

Inspired by these observations, we aim to take into account all the four kinds of relationships among documents and words (i.e. the relationships between documents, the relationships between words, the relationships between words and documents, and the relationships between documents and words) in both source domain and target domain under a unified framework for sentiment transfer.

In this work, we propose an iterative reinforcement approach to implement the above idea. The proposed approach makes full use of all the relationships among documents and words from both source domain and target domain to transfer information between domains. In our approach, the opinion of a document (word) is reinforced by the opinion of all its interrelated documents and words; and the updated opinion of the document (word) will conversely reinforce the opinions of its interrelated documents and words. That is to say, it is an iterative reinforcement process until it converges to a final result.

The contribution of our work is twofold. First, we extend the traditional sentiment-transfer methods by utilizing the full knowledge between interrelated documents and words. Second, we present a reinforcement approach to get the opinions of documents by making use of graph-ranking algorithm.

The proposed approach is evaluated on three domain-specific sentiment data sets. The experimental results show that our approach can dramatically improve the accuracy when transferred to another target domain. And we also conduct extensive experiments to investigate the parameters sensitivity. The results show that our algorithm is not sensitive to these parameters.

2 Proposed Methods

2.1 Problem Definition

In this paper, we have two document sets: the test documents $D^U = \{d_1,\ldots,d_{nd}\}$ where d_i is the term vector of the i^{th} text document and each $d_i \in D^U (i = 1,\ldots,nd)$ is unlabeled; the training documents $D^L = \{d_{nd+1},\ldots,d_{nd+md}\}$ where d_j represents the term vector of the j^{th} text document and each $d_j \in D^L (j = nd+1,\ldots,nd+md)$ should have a label from a category set C = {negative, positive}. We assume the training dataset D^L is from the interrelated but different domain with the test dataset D^U. Also, we have two word sets: $W^U = \{w_1,\ldots,w_{nw}\}$ is the word set of D^U and each $w_i \in W^U$ $(i = 1,\ldots,nw)$ is unlabeled; $W^L = \{w_{nw+1},\ldots,w_{nw+mw}\}$ is the word set of D^L and each $w_j \in W^L (j = nw+1,\ldots,nw+mw)$ has a label from C. Our objective is to maximize the accuracy of assigning a label in C to $d_i \in D^U$ $(i = 1,\ldots,nd)$ utilizing the training data D^L and W^L in another domain.

The proposed algorithm is based on the following presumptions:

(1) $W^L \cap W^U \neq \Phi$.

(2) The labels of documents appear both in the training data and the test data should be the same.

2.2 Overview

The proposed approach is inspired by graph-ranking algorithm whose idea is to give a node high score if it is strongly linked with other high-score nodes. Graph-ranking algorithm has been successfully used in many fields (e.g. PageRank (Brin et al, 1999), LexRank (Erkan and Radev, 2004)). We can get the following thoughts based on the ideas of PageRank and HITS (Kleinberg, 1998):

(1) If a document is strongly linked with other positive (negative) documents, it tends to be positive (negative); and if a word is strongly linked with other positive (negative) words, it tends to be positive (negative).

(2) If a document contains many positive (negative) words, it tends to be positive (negative); and if a word appears in many positive (negative) documents, it tends to be positive (negative).

Given the data points of documents and words, there are four kinds of relationships in our problem:

- DD-Relationship: It denotes the relationships between documents, usually computed by their content similarity.
- WW-Relationship: It denotes the relationships between words, usually computed by knowledge-based approach or corpus-based approach.
- DW-Relationship: It denotes the relationships between documents and words, usu-

ally computed by the relative importance of a word in a document.

- WD-Relationship: It denotes the relationships between words and documents, usually computed by the relative importance of a document to a word.

Meanwhile, our problem refers to both source domain and target domain, so our approach considers eight relationships altogether: DDO-Relationship (the relationships between D^U and D^L), DDN-Relationship (the relationships between D^U), WWO-Relationship (the relationships between W^U and W^L), WWN-Relationship (the relationships between W^U and W^U), DWO-Relationship (the relationships between D^U and W^L), DWN-Relationship (the relationships between D^U and W^U), WDO-Relationship (the relationships between W^U and D^L), WDN-Relationship (the relationships between W^U and D^U). The first four relationships are used to compute the sentiment scores of the documents, and the others are used to compute the sentiment scores of the words.

The iterative reinforcement approach could make full use of all the relationships in a unified framework. The framework of the proposed approach is illustrated in Figure 1.

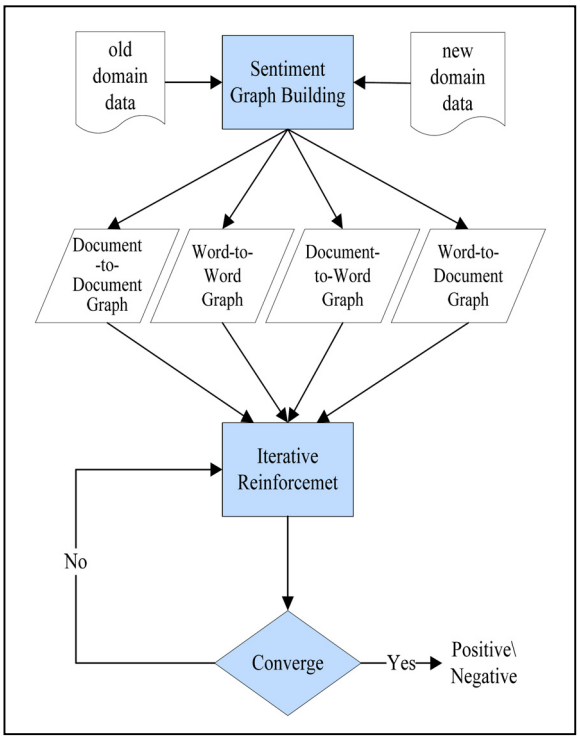


Figure 1. Framework of the proposed approach

The framework consists of a graph-building phase and an iterative reinforcement phase. In the graph-building phase, the input includes both the labeled data from source domain and the unlabeled data from target domain. The proposed approach builds four graphs based on these data to reflect the above relationships respectively. For source-domain data, we initialize every document and word a score ("1" denotes positive, and "-1" denotes negative) to represent its degree of sentiment orientation, and we call it sentiment score; for target-domain data, we set the initial sentiment scores to 0.

In the iterative reinforcement phase, our approach iteratively computes the sentiment scores of the documents and words based on the graphs. When the algorithm converges, all the documents get their sentiment scores. If its sentiment score is between 0 and 1, the document should be classified as "positive". The closer its sentiment score is near 1, the higher the "positive" degree is. Otherwise, if its sentiment score is between 0 and -1, the document should be classified as "negative". The closer its sentiment score is near -1, the higher the "negative" degree is.

The algorithms of sentiment graph building and iterative reinforcement are described in details in the next sections, respectively.

2.3 Sentiment-Graph Building

Symbol Definition

In this section, we build four graphs to reflect eight relationships, and the meanings of symbols are shown in Table 1.

Relationship	Similarity matrix	Normalized form	Neighbor matrix
DDO	$U^L=[U^L_{ij}]_{ndxmd}$	$\hat{U}^L$	$Un^L=[Un^L_{ij}]_{nd\times K}$
DDN	$U^U=[U^U_{ij}]_{ndxnd}$	$\hat{U}^U$	$Un^U=[Un^U_{ij}]_{nd\times K}$
WWO	$V^L=[V^L_{ij}]_{nwxmw}$	$\hat{V}^L$	$Vn^L=[Vn^L_{ij}]_{nw\times K}$
WWN	$V^U=[V^U_{ij}]_{nwxnw}$	$\hat{V}^U$	$Vn^U=[Vn^U_{ij}]_{nw\times K}$
DWO	$M^L=[M^L_{ij}]_{ndxmw}$	$\hat{M}^L$	$Mn^L=[Mn^L_{ij}]_{nd\times K}$
DWN	$M^U=[M^U_{ij}]_{ndxnw}$	$\hat{M}^U$	$Mn^U=[Mn^U_{ij}]_{nd\times K}$
WDO	$N^L=[N^L_{ij}]_{nwxmd}$	$\hat{N}^L$	$Nn^L=[Nn^L_{ij}]_{nw\times K}$
WDN	$N^U=[N^U_{ij}]_{nwxnd}$	$\hat{N}^U$	$Nn^U=[Nn^U_{ij}]_{nw\times K}$

Table 1: Symbol definition

In this table, the first column denotes the name of the relationship; the second column denotes

the similarity matrix to reflect the corresponding relationship; in consideration of convergence, we normalize the similarity matrix, and the normalized form is listed in the third column; in order to compute sentiment scores, we find the neighbors of a document or a word and the neighbor matrix is listed in the fourth column.

Document-to-Document Graph

We build an undirected graph whose nodes denote documents in both D^L and D^U and edges denote the content similarities between documents. If the content similarity between two documents is 0, there is no edge between the two nodes. Otherwise, there is an edge between the two nodes whose weight is the content similarity. The edges in this graph are divided into two parts: edges between D^U and D^L; edges between D^U itself, so we build the graph in two steps.

(1) Create D^U and D^L Edges

The content similarity between two documents is computed with the cosine measure. We use an adjacency matrix U^L to denote the similarity matrix between D^U and D^L. $U^L=[U^L_{ij}]_{ndxmd}$ is defined as follows:

$$U^L_{ij} = \frac{d_i \bullet d_{j+nd}}{\|d_i\| \times \|d_{j+nd}\|},\ i = 1,...,nd, j = 1,...,md \qquad (1)$$

The weight associated with word w is computed with $tf_w idf_w$ where tf_w is the frequency of word w in the document and idf_w is the inverse document frequency of word w, i.e. $1+\log(N/n_w)$, where N is the total number of documents and n_w is the number of documents containing word w in a data set.

In consideration of convergence, we normalize U^L to $\hat{U}^L$ by making the sum of each row equal to 1:

$$\hat{U}^L_{ij} = \begin{cases} U^L_{ij} \Big/ \sum_{j=1}^{md} U^L_{ij}, & if \sum_{j=1}^{md} U^L_{ij} \neq 0 \\ 0, & otherwise \end{cases} \qquad (2)$$

In order to find the neighbors (in another word, the nearest documents) of a document, we sort every row of $\hat{U}^L$ to $\tilde{U}^L$ in descending order. That is: $\tilde{U}^L_{ij} \geq \tilde{U}^L_{ik}$ ($i = 1,\ldots,nd; j,k = 1,\ldots,md; k \geqslant j$).

Then for $d_i \in D^U$ ($i = 1,\ldots,nd$), $\tilde{U}^L_{ij}$ ($j = 1,\ldots,K$) corresponds to K neighbors in D^L. We use a matrix $Un^L = [Un^L_{ij}]_{nd \times K}$ to denote the neighbors of D^U in source domain, with Un^L_{ij} corresponding to the j^{th} nearest neighbor of d_i.

(2) Create D^U and D^U Edges

Similarly, the edge weight between D^U itself is computed by the cosine measure. We get the similarity matrix $U^U=[U^U_{ij}]_{ndxnd}$, the normalized similarity matrix $\hat{U}^U$, and the neighbors of D^U in target domain: $Un^U = [Un^U_{ij}]_{nd \times K}$.

Word-to-Word Graph

Similar to the Document-to-Document Graph, we build an undirected graph to reflect the relationship between words in W^L and W^U, in which each node corresponds to a word and the edge weight between any different words corresponds to their semantic similarity. The edges in this graph are divided into two parts: edges between W^U and W^L; edges between W^U itself, so we also build the graph in two steps.

(1) Create W^U and W^L Edges

We compute the semantic similarity using corpus-based approach which computes the similarity between words utilizing information from large corpora. There are many measures to identify word semantic similarity, such as mutual information (Turney, 2001), latent semantic analysis (Landauer et al., 1998) etc. In this study, we compute word semantic similarity based on the sliding window measure, that is, two words are semantically similar if they co-occur at least once within a window of maximum K_{win} words, where K_{win} is the window size. We use an adjacency matrix V^L to denote the similarity matrix between W^U and W^L. $V^L=[V^L_{ij}]_{nwxmw}$ is defined as follows:

$$V^L_{ij} = \begin{cases} \log \frac{N \times p(w_i, w_{j+nw})}{p(w_i) \times p(w_{j+nw})}, & if\ w_i \neq w_{j+nw} \\ 0, & otherwise \end{cases} \qquad (3)$$

where N is the total number of words in D^U; $p(w_i, w_j)$ is the probability of the co-occurrence of w_i and w_j within a window, i.e. $num(w_i, w_j)/N$, where $num(w_i, w_j)$ is the number of the times w_i and w_j co-occur within the window; $p(w_i)$ and $p(w_j)$ are the probabilities of the occurrences of w_i and w_j respectively, i.e. $num(w_i)/N$ and $num(w_j)/N$, where $num(w_i)$ and $num(w_j)$ are the

numbers of the times w_i and w_j occur. We normalize V^L to $\hat{V}^L$ to make the sum of each row equal to 1. Then we sort every row of $\hat{V}^L$ to $\tilde{V}^L$ in descending order, and we use a matrix $Vn^L = [Vn^L{}_{ij}]_{nw \times K}$ to denote the neighbors of W^U in source domain.

(2) Create W^U and W^U Edges

Then we also compute the edge weight between any different nodes which denote words in W^U by the sliding window measure. We get the similarity matrix $V^U=[V^U{}_{ij}]_{nwxnw}$, the normalized similarity matrix $\hat{V}^U$, and the neighbors of W^U in target domain: $Vn^U = [Vn^U{}_{ij}]_{nw \times K}$.

Document-to-Word Graph

We can build a weighted directed bipartite graph from documents in D^U and words in W^L and W^U in the following way: each node in the graph corresponds to a document in D^U or a word in W^L and W^U; if word w_j appears in document d_i, we create an edge from d_i to w_j. The edges in this graph are divided into two parts: edges from D^U to W^L; edges from D^U to W^U, so we also build the graph in two steps.

(1) Create D^U to W^L Edges

The edge weight from a document in D^U to a word in W^L is proportional to the importance of word w_j in document d_i. We use an adjacency matrix M^L to denote the similarity matrix from D^U to W^L. $M^L=[M^L{}_{ij}]_{ndxmw}$ is defined as follows:

$$M^L_{ij} = \frac{tf_{w_{j+nw}} \times idf_{w_{j+nw}}}{\sum_{w \in d_i} tf_w \times idf_w} \tag{4}$$

where w represents a unique word in d_i and tf_w, idf_w are respectively the term frequency in the document and the inverse document frequency. We normalize M^L to $\hat{M}^L$ to make the sum of each row equal to 1. Then we sort every row of $\hat{M}^L$ to $\tilde{M}^L$ in descending order, and we use a matrix $Mn^L = [Mn^L{}_{ij}]_{nd \times K}$ to denote the neighbors of D^U in W^L.

(2) Create D^U to W^U Edges

Similarly, we can also compute the edge weight from a document in D^U to a word in W^U in the same way. We get the similarity matrix $M^U=[M^U{}_{ij}]_{ndxnw}$, the normalized similarity matrix $\hat{M}^U$, and the neighbors of D^U in W^U: $Mn^U = [Mn^U{}_{ij}]_{nd \times K}$.

Word-to-Document Graph

In this section, we build a weighted directed bipartite graph from words in W^U and documents in D^L and D^U in which each node in the graph corresponds to a word in W^U and a document in D^L or D^U; if word w_j appears in document d_i, we create an edge from w_j to d_i. The edges in this graph are also divided into two parts: edges from W^U to D^L; edges from W^U to D^U.

(1) Create W^U to D^L Edges

Similar to 3.3.4, the edge weight from a word in W^U to a document in D^L is proportional to the importance of word w_i in document d_j. We use an adjacency matrix $N^L=[N^L{}_{ij}]_{nwxmd}$ to denote the similarity matrix from W^U to D^L. We normalize N^L to $\hat{N}^L$ to make the sum of each row equal to 1. Then we sort every row of $\hat{N}^L$ to $\tilde{N}^L$ in descending order, and we use a matrix $Nn^L = [Nn^L{}_{ij}]_{nw \times K}$ to denote the neighbors of W^U in D^L.

(2) Create W^U to D^U Edges

We can also compute the edge weight from a word in W^U to a document in D^U in the same way. We get the similarity matrix $N^U=[N^U{}_{ij}]_{nwxnd}$, the normalized similarity matrix $\hat{N}^U$, and the neighbors of W^U in D^U: $Nn^U = [Nn^U{}_{ij}]_{nw \times K}$.

2.4 Proposed Method

Based on the two thoughts introduced in Section 2.2, we fuse the eight relationships abstracted from the four graphs together to iteratively reinforce sentiment scores, and we can obtain the iterative equation as follows:

$$\begin{aligned} ds_i = {} & \varphi \sum_{g \in Un^L{}_{i\bullet}} (\hat{U}^L{}_{ig} \times ds_g) + \mu \sum_{h \in Un^U{}_{i\bullet}} (\hat{U}^U{}_{ih} \times ds_h) \\ & + \gamma \sum_{l \in Mn^L{}_{i\bullet}} (\hat{M}^L{}_{il} \times ws_l) + \delta \sum_{r \in Mn^U{}_{i\bullet}} (\hat{M}^U{}_{ir} \times ws_r) \end{aligned} \tag{5}$$

$$\begin{aligned} ws_j = {} & \varphi \sum_{g \in Vn^L{}_{j\bullet}} (\hat{V}^L{}_{jg} \times ws_g) + \mu \sum_{h \in Vn^U{}_{j\bullet}} (\hat{V}^U{}_{jh} \times ws_h) \\ & + \gamma \sum_{l \in Nn^L{}_{j\bullet}} (\hat{N}^L{}_{jl} \times ds_l) + \delta \sum_{r \in Nn^U{}_{j\bullet}} (\hat{N}^U{}_{jr} \times ds_r) \end{aligned} \tag{6}$$

where $i\bullet$ means the i^{th} row of a matrix; $Ds = \{ds_1,\ldots,ds_{nd}, ds_{nd+1},\ldots, ds_{nd+md}\}$ represents the sentiment scores of D^U and D^L; $Ws = \{ws_1,\ldots,ws_{nw}, ws_{nw+1},\ldots, ws_{nw+mw}\}$ represents the sentiment scores of W^U and W^L; φ and μ show

the relative contributions to the final sentiment scores from source domain and target domain when calculating DD-Relationship and WW-Relationship, and $\varphi + \mu = 1$; γ and δ show the relative contributions to the final sentiment scores from source domain and target domain when calculating DW-Relationship and WD-Relationship, and $\gamma + \delta = 1$.

For simplicity, we merge the relationships from source domain and target domain. That is, for formula (5), we merge the first two items into one, the last two items into one; for formula (6), we merge its first two items into one, its last two items into one. Thus, (5) and (6) are transformed into (7) and (8) as follows:

$$ds_i = \alpha \times \sum_{g \in Un_{i\bullet}} (\hat{U}_{ig} \times ds_g) + \beta \times \sum_{l \in Mn_{i\bullet}} (\hat{M}_{il} \times ws_l) \quad (7)$$

$$ws_j = \alpha \times \sum_{g \in Nn_{j\bullet}} (\hat{N}_{jg} \times ds_g) + \beta \times \sum_{l \in Vn_{j\bullet}} (\hat{V}_{jl} \times ws_l) \quad (8)$$

where α and β show the relative contributions to the final sentiment scores from document sets and word sets, and $\alpha + \beta = 1$.

In consideration of the convergence, Ds and Ws are normalized separately after each iteration as follows to make the sum of positive scores equal to 1, and the sum of negative scores equal to -1:

$$ds_i = \begin{cases} ds_i \Big/ \sum_{j \in D^U_{neg}} (-ds_j), & if \quad ds_i < 0 \\ ds_i \Big/ \sum_{j \in D^U_{pos}} ds_j, & if \quad ds_i > 0 \end{cases} \quad (9)$$

$$ws_j = \begin{cases} ws_j \Big/ \sum_{i \in W^U_{neg}} (-ws_i), & if \quad ws_j < 0 \\ ws_j \Big/ \sum_{i \in W^U_{pos}} ws_i, & if \quad ws_j > 0 \end{cases} \quad (10)$$

where D^U_{neg} and D^U_{pos} denote the negative and positive document set of D^U respectively; W^U_{neg} and W^U_{pos} denote the negative and positive word set of W^U respectively.

Here is the complete algorithm:

1. Initialize the sentiment score *vector* ds_i of $d_i \in D^L$ ($i = nd+1,\ldots, nd+md$) with 1 when d_i is labeled "positive", and with -1 when d_i is labeled "negative", and initialize the sentiment score *vector* ws_i of $w_i \in W^L$ ($i = nw+1,\ldots, nw+mw$) with 1 when w_i is labeled "positive", and with -1 when w_i is labeled "negative". And we normalize ds_i ($i = nd+1,\ldots, nd+md$) (ws_i ($i = nw+1,\ldots, nw+mw$)) to make the sum of positive scores of D^L (W^L) equal to 1, and the sum of negative scores of D^L (W^L) equal to -1. Also, the initial sentiment scores of D^U and W^U are set to 0.
2. Alternate the following two steps until convergence:
   2.1. Compute and normalize ds_i ($i = 1,\ldots, nd$) using formula (7) and (9):
   2.2. Compute and normalize ws_j ($j=1,\ldots,nw$) using formula (8) and (10):

   where $ds_i^{(k)}$ and $ws_j^{(k)}$ denote the ds_i and ws_j at the k^{th} iteration.
3. According to $ds_i \in Ds$ ($i = 1,\ldots,nd$), assign each $d_i \in D^U$ ($i = 1,\ldots,nd$) a label. If ds_i falls in the range [-1,0], assign d_i the label "negative"; if ds_i falls in the range [0,1], assign d_i the label "positive".

3 Experiments

In this section, we evaluate our approach on three different domains and compare it with some state-of-the-art algorithms, and also evaluate the approach's sensitivity to its parameters. Note that we conduct experiments on Chinese data, but the main idea in the proposed approach is language-independent in essence.

3.1 Data Preparation

We use three Chinese domain-specific data sets from on-line reviews, which are: Book Reviews[1] (B, www.dangdang.com/), Hotel Reviews[2] (H, www.ctrip.com/) and Notebook Reviews[3] (N, www.360buy.com/). Each dataset has 4000 labeled reviews (2000 positives and 2000 negatives).

We use ICTCLAS (http://ictclas.org/), a Chinese text POS tool, to segment these Chinese reviews. Then, utilizing the part-of-speech tagging function provided by ICTCLAS, we take all adjectives, adverbs and adjective-noun phrases as candidate sentiment words. After removing the repeated words and ambiguous words, we get a list of words in each domain.

For the list of words in each domain, we manually label every word as "negative", "posi-

[1] www.searchforum.org.cn/tansongbo/corpus/Dangdang_Book_4000.rar
[2] www.searchforum.org.cn/tansongbo/corpus/Ctrip_htl_4000.rar
[3] www.searchforum.org.cn/tansongbo/corpus/Jingdong_NB_4000.rar

tive" or "neutral", and we take those "negative" and "positive" words as a sentiment word set.

Note that we use the sentiment word set only for source domain, while using the candidate sentiment words for target domain.

Lastly, the documents are represented by vector space model. In this model, each document is converted into bag-of-words presentation in the remaining term space. We compute term weight with the frequency of the term in the document.

We choose one of the three data sets as source-domain data D^L, and its corresponding sentiment word set as W^L; we choose another data set as target-domain data D^U, and its corresponding candidate sentiment words as W^U.

3.2 Baseline Methods

In this paper we compare our approach with the following baseline methods:

Proto: This method applies a traditional supervised classifier, prototype classifier (Tan et al., 2005), for the sentiment transfer. And it only uses source domain documents as training data.

LibSVM: This method applies a state-of-the-art supervised learning algorithm, Support Vector Machine, for the sentiment transfer. In detail, we use LibSVM (Chang and Lin, 2001) with a linear kernel and set all options as default. This method only uses source domain documents as training data.

TSVM: This method applies transductive SVM (Joachims, 1999) for the sentiment transfer which is a widely used method for improving the classification accuracy. In our experiment, we use Joachims's SVM-light package (http://svmlight.joachims.org/) for TSVM. We use a linear kernel and set all parameters as default. This method uses both source domain data and target domain data.

3.3 Overall Performance

In this section, we compare proposed approach with the three baseline methods. There are three parameters in our algorithm, K, K_{win}, α (β can be calculated by 1-α). We set K to 50, and K_{win} to 10 respectively. With different α, our approach can be considered as utilizing different relative contributions from document sets and word sets. In order to identify the importance of both document sets and word sets for sentiment transfer, we separately set α to 0, 1, 0.5 to show the accuracy of utilizing only word sets (referred to as WORD), only document sets (referred to as DOC), and both the document and word sets (referred to as ALL). It is thought that the algorithm achieves the convergence when the changing between the sentiment score ds_i computed at two successive iterations for any $d_i \in D^U$ (i = 1,…,nd) falls below a given threshold, and we set the threshold 0.00001 in this work. The parameters will be studied in parameters sensitivity section.

Table 2 shows the accuracy of Prototype, LibSVM, TSVM and our algorithm when training data and test data belong to different domains.

As we can observe from Table 2, our algorithm produces much better performance than supervised baseline methods. Compared with the traditional classifiers, our approach outperforms them by a wide margin on all the six transfer tasks. The great improvement compared with the baselines indicates that our approach performs very effectively and robustly.

	Traditional Classifier		TSVM	Our Approach		
	Proto	LibSVM		DOC	WORD	ALL
B->H	0.735	0.747	0.749	**0.772**	0.734	0.763
B->N	0.651	0.652	0.769	0.714	0.785	**0.795**
H->B	0.645	0.675	0.614	0.671	0.668	**0.703**
H->N	0.729	0.669	0.726	**0.749**	0.727	0.734
N->B	0.612	0.608	0.622	0.638	0.667	**0.726**
N->H	0.724	0.711	0.772	0.764	0.740	**0.792**
Average	0.683	0.677	0.709	0.718	0.720	**0.752**

Table 2: Accuracy comparison of different methods

Table 2 shows the average accuracy of TSVM is higher than both traditional classifiers, since it utilizes the information of both source domain and target domain. However, the proposed approach outperforms TSVM: the average accuracy of the proposed approach is about 4.3% higher than TSVM. This is caused by two reasons. First, TSVM is not dedicated for sentiment-transfer learning. Second, TSVM requires the ratio between positive and negative examples in the test data to be close to the ratio in the training data, so its performance will be affected if this requirement is not met.

Results of "DOC" and "WORD" are shown in column 4 and 5 of Table 2. As we can observe, they produce better performance than all the baselines. This is caused by two reasons. First, "DOC" and "WORD" separately utilize the sen-

timent information of documents and words. Second, both "DOC" and "WORD" involve an iterative reinforcement process to improve their performance. The great improvement indicates that the iterative reinforcement approach is effective for sentiment transfer.

Besides, Table 2 also shows both document sets and word sets are important for sentiment transfer. The approach "ALL" outperforms the approaches "DOC" and "WORD" on almost all the six transfer tasks except "B->H" and "H->N". The average increase of accuracy over all the six tasks is 3.4% and 3.2% respectively. The reason is: at every iteration, the classification accuracy of documents and words is improved by each other, and then the accuracy of sentiment transfer is improved by the documents and words that are classified more accurately. As for "B->H" and "H->N", the performance of utilizing only document sets is so good that the word sets couldn't improve the performance any more. The improvement of the approach "ALL" convinces us that not a single one of the four relationships can be omitted.

3.4 Parameters Sensitivity

The proposed algorithm has an important parameter, α (β can be calculated by 1-α). In this section, we conduct experiments to show that our algorithm is not sensitive to this parameter.

To investigate the sensitivity of proposed method involved with the parameter α, we set K to 50, and K_{win} to 10. And we change α from 0 to 1, an increase of 0.1 each. We also evaluate α on the six tasks mentioned in section 3.1, and the results are shown in figure 2.

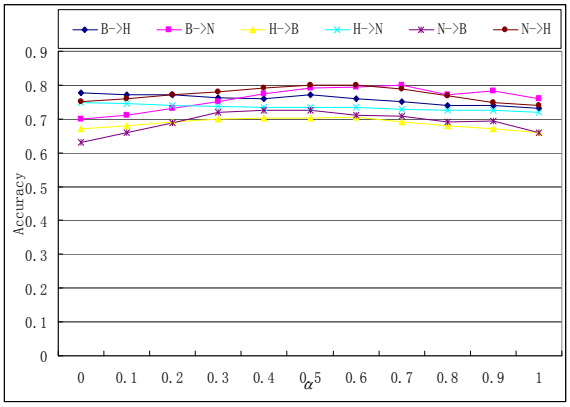


Figure 2: Accuracy for Different α

We can observe from Figure 2 that the accuracy first increases and then decreases when α is increased from 0 to 1. The accuracy changes gradually when α is near 0 or 1, and it changes less when α is between 0.2 and 0.8. It is easy to explain this phenomenon. When α is set to 0, this indicates our algorithm only uses word sets to aid classification, without the information of document sets. And if α is set to 1, our algorithm only uses document sets to calculate sentiment score, without the help of word sets. Both cases above don't use all information of four relationships, so their accuracies are worse than to equal the contributions of both document and word sets. This experiment shows that the proposed algorithm is not sensitive to the parameter α as long as α is not 0 or 1. We set α to 0.5 in our overall-performance experiment.

3.5 Convergence

Our algorithm is an iterative process that will converge to a local optimum. We evaluate its convergence on the six tasks mentioned above. Figure 3 shows the change of accuracy with respect to the number of iterations. We can observe from figure 3 that the curve rises sharply during the first 6 iterations, and it is very stable after 10 iterations are performed. This experiment indicates that our algorithm could converge very quickly to get a local optimum.

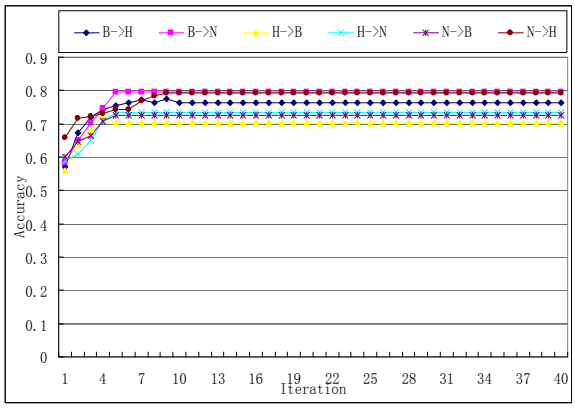


Figure 3: Performance for Iteration

4 Conclusions

In this paper, we propose a novel cross-domain sentiment classification approach, which is an iterative reinforcement approach for sentiment transfer by utilizing all the relationships among documents and words from both source domain and target domain to transfer information between domains. First, we build three graphs to reflect the above relationships respectively. Then,

we assign a score for every unlabelled document to denote its extent to "negative" or "positive". We then iteratively calculate the score by making use of the graphs. Finally, the final score for sentiment classification is achieved when the algorithm converges, so we can label the target-domain data based on these scores.

We conduct experiments on three domain-specific sentiment data sets. The experimental results show that the proposed approach could dramatically improve the accuracy when transferred to a target domain. To investigate the parameter sensitivity, we conduct experiments on the same data sets. It is observed that our approach is not very sensitive to its four parameters, and could converge very quickly to get a local optimum.

In this study, we employ only cosine measure, sliding window measure and vector measure to compute similarity. These are too general, and perhaps not so suitable for sentiment classification. In the future, we will try other methods to calculate the similarity. Furthermore, we experiment our approach on only three domains, and we will apply our approach to many more domains.

5 Acknowledgments

This work was mainly supported by two funds, i.e., 60933005 & 60803085, and two another projects, i.e., 2007CB311100 & 2007AA01Z441.

References

Alina Andreevskaia and Sabine Bergler. 2008. When Specialists and Generalists Work Together: Overcoming Domain Dependence in Sentiment Tagging. In Proceedings of ACL: 290-298.

Anthony Aue and Michael Gamon. 2005. Customizing sentiment classifiers to new domains: a case study. In Proceedings of RANLP.

Sergey Brin, Lawrence Page, Rajeev Motwami, and Terry Winograd. 1999. The PageRank citation ranking: bringing order to the web. Technical Report 1999-0120, Stanford, CA.

Chinchung Chang and Chinjen Lin. 2001. LIBSVM: a library for support vector machines. http://www.csie.ntu.edu.tw/~cjlin/libsvm.

Gunes Erkan and Dragomir Radev. 2004. LexRank: Graph-based Centrality as Salience in Text Summarization. Journal of Artificial Intelligence Research, 22 (2004): 457-479.

Michael Gamon and Anthony Aue. 2005. Automatic identification of sentiment vocabulary: exploiting low association with known sentiment terms. In Proceedings of the ACL Workshop on Feature Engineering for Machine Learning in NLP: 57-64.

Songbo Tan, Xueqi Cheng, Moustafa Ghanem, Bin Wang, Hongbo Xu. 2005. A novel refinement approach for text categorization. In Proceedings of CIKM 2005: 469-476

Thorsten Joachims. 1999. Transductive inference for text classification using support vector machines. In Proceedings of ICML.

Jon Kleinberg. 1998. Authoritative sources in a hyperlinked environment. Journal of the ACM, 46(5): 604-632.

Lunwei Ku, Yuting Liang, and Hsinhsi Chen. 2006. Opinion extraction, summarization and tracking in news and blog corpora. In Proceedings of AAAI.

Thomas Landauer, Peter Foltz, and Darrell Laham. 1998. Introduction to latent semantic analysis. Discourse Processes 25: 259-284.

Ryan McDonald, Kerry Hannan, Tyler Neylon, Mike Wells, and Jeff Reynar. 2007. Structured models for fine-to-coarse sentiment analysis. In Proceedings of ACL.

Songbo Tan, Yuefen Wang, Gaowei Wu, and Xueqi Cheng. 2007. A novel scheme for domain-transfer problem in the context of sentiment analysis. In Proceedings of CIKM.

Songbo Tan, Xueqi Cheng, Yuefen Wang, and Hongbo Xu. 2009. Adapting Naïve Bayes to Domain Adaptation for Sentiment Analysis. In Proceedings of ECIR.

Qiong Wu, Songbo Tan and Xueqi Cheng. 2009. Graph Ranking for Sentiment Transfer. In Proceedings of ACL-IJCNLP.

Peter Turney. 2001. Mining the web for synonyms: PMI-IR versus LSA on TOEFL. In Proceedings of ECML: 491-502.

Exploring the Use of Word Relation Features for Sentiment Classification

Rui Xia and **Chengqing Zong**
National Laboratory of Pattern Recognition
Institute of Automation, Chinese Academy of Sciences
{rxia, cqzong}@nlpr.ia.ac.cn

Abstract

Word relation features, which encode relation information between words, are supposed to be effective features for sentiment classification. However, the use of word relation features suffers from two issues. One is the sparse-data problem and the lack of generalization performance; the other is the limitation of using word relations as additional features to unigrams. To address the two issues, we propose a generalized word relation feature extraction method and an ensemble model to efficiently integrate unigrams and different type of word relation features. Furthermore, aimed at reducing the computation complexity, we propose two fast feature selection methods that are specially designed for word relation features. A range of experiments are conducted to evaluate the effectiveness and efficiency of our approaches.

1 Introduction

The task of text sentiment classification has become a hotspot in the field of natural language processing in recent years (Pang and Lee, 2008). The dominating text representation method in sentiment classification is known as the bag-of-words (BOW) model. Although BOW is quite simple and efficient, a great deal of the information from original text is discarded, word order is disrupted and syntactic structures are broken. Therefore, more sophisticated features with a deeper understanding of the text are required for sentiment classification tasks.

With the attempt to capture the word relation information behind the text, word relation (WR) features, such as higher-order n-grams and word dependency relations, have been employed in text representation for sentiment classification (Dave et al., 2003; Gamon, 2004; Joshi and Penstein-Rosé, 2009).

However, in most of the literature, the performance of individual WR feature set was poor, even inferior to the traditional unigrams. For this reason, WR features were commonly used as additional features to supplement unigrams, to encode more word order and word relation information. Even so, the performance of joint features was still far from satisfactory (Dave et al., 2003; Gamon, 2004; Joshi and Penstein-Rosé, 2009).

We speculate that the poor performance is possibly due to the following two reasons: 1) in WR features, the data are sparse and the features lack generalization capability; 2) the use of joint features of unigrams and WR features has its limitation.

On one hand, there were attempts at finding better generalized WR (GWR) features. Gamon (2004) back off words in n-grams (and semantic relations) to their respective POS tags (e.g., *great-movie* to adjective-noun); Joshi and Rosé (2009) propose a method by only backing off the head word in dependency relation pairs to its POS tag (e.g., *great-movie* to *great*-noun), which are supposed to be more generalized than word pairs. Based on Joshi and Rosé's method, we back off the word in each word relation pairs to its corresponding POS cluster, making the feature space smarter and more effective.

On the other hand, we find that from unigrams to WR features, relevance between features is reduced and the independence is in-

Coling 2008: Poster Volume, pages 1336–1344,
Beijing, August 2010

creased. Although the discriminative model (e.g., SVM) is proven to be more effective on unigrams (Pang et al., 2002) for its ability of capturing the complexity of more relevant features, WR features are more inclined to work better in the generative model (e.g., NB) since the feature independence assumption holds well in this case.

Based on this finding, we therefore intuitively seek, instead of jointly using unigrams and GWR features, to efficiently integrate them to synthesize a more accurate classification procedure. We use the ensemble model to fuse different types of features under distinct classification models, with an attempt to overcome individual drawbacks and benefit from each other's merit, and finally to enhance the overall performance.

Furthermore, feature reduction is another important issue of using WR features. Due to the huge dimension of WR feature space, traditional feature selection methods in text classification perform inefficiently. However, to our knowledge, no related work has focused on feature selection specially designed for WR features.

Taking this point into consideration, we propose two fast feature selection methods (FMI and FIG) for GWR features with a theoretical proof. FMI and FIG regard the importance of a GWR feature as two component parts, and take the sum of two scores as the final score. FMI and FIG remain a close approximation to MI and IG, but speed up the computation by at most 10 times. Finally, we apply FMI and FIG to the ensemble model, reducing the computation complexity to a great extent.

The remainder of this paper is organized as follows. In Section 2, we introduce the approach to extracting GWR features. In Section 3, we present the ensemble model for integrating different types of features. In Section 4, the fast feature selection methods for WR features are proposed. Experimental results are reported in Section 5. Section 6 draws conclusions and outlines directions for future work.

2 Generalized Word Relation Features

A straightforward method for extracting WR features is to simply map word pairs into the feature vector. However, due to the sparse-data problem and the lack of generalization ability, the performance of WR is discounted. Consider the following two pieces of text:

1) *Avatar is a great movie. I definitely recommend it.*

2) *I definitely recommend this book. It is great.*

We lay the emphasis on the following word pairs: *great-movie, great-it, it-recommend*, and *book-recommend*. Although these features are good indicators of sentiment, due to the sparse-data problem, they may not contribute as importantly as we have expected in machine learning algorithms. Moreover, the effects of those features would be greatly reduced when they are not captured in the test dataset (for example, a new feature *great-song* in the test set would never benefit from *great-movie* and *great-it*).

Joshi and Rosé (2009) back off the head word in each of the relation pairs to its POS tag. Taking *great-movie* for example, the back-off feature will be *great*-noun. With such a transformation, original features like *great-movie*, *great-book* and other *great*-noun pairs are regarded as one feature, hence, the learning algorithms could learn a weight for a more general feature that has stronger evidence of association with the class, and any new test sentence that contains an unseen noun in a similar relationship with the adjective *great* (e.g., *great-song*) will receive some weight in favor of the class label.

With the attempt to make a further generalization, we conduct a POS clustering. Considering the effect of different POS tags in both unigrams and word relations, the POS tags are categorized as shown in Table 1.

POS-cluster	Contained POS tags
J	JJ, JJS, JJR
R	RB, RBS, RBR
V	VB, VBZ, VBD, VBN, VBG, VBP
N	NN, NNS, NNP, NNPS, PRP
O	The other POS tags

Table 1: POS Clustering (the Penn Corpus Style)

Since adjectives and adverbs have the highest correlation with sentiment, and some verbs and nouns are also strong indicators of sentiment, we therefore put them into separate clusters. All the other tags are categorized to one cluster because they contain a lot of noise rather than useful information. In addition, we assign pronouns to POS-cluster N, aimed at capturing the generality in WR features like *great-movie* and *great-it*, or *book-recommend* and *it-recommend*.

Taking "*Avatar is a great movie*" for example, different types of WR features are presented in Table 2, where Uni denotes unigrams; WR-Bi indicates traditional bigrams; WR-Dp indicates word pairs of dependency relation; GWR-Bi and GWR-Dp respectively denote generalized bigrams and dependency relations.

WR types	WR features
WR-Bi	*Avatar-is, is-a, a-great, great-movie*
WR-Dp	*Avatar-is, a-movie, great-movie, movie-is*
GWR-Bi	*Avatar*-V, *is*-O, *a*-J, *great*-N, N-*is*, V-*a*, O-*great*, J-*movie*
GWR-Dp	*Avatar*-V, *a*-N, *great*-N, *movie*-V, N-*is*, O-*movie*, J-*movie*

Table 2: Different types of WR features

3 An Ensemble Model for Integrating WR Features

3.1 Joint Features, Good Enough?

Although the unigram feature space is simple, and the WR features are more sophisticated, the latter was mostly used as extra features in addition to the former, rather than to substitute it. Even so, in most of the literature, the improvements of joint features are still not as good as we had expected. For example, Dave et al. (2003) try to extract a refined subset of WR pairs (adjective-noun, subject-verb, and verb-object pairs) as additional features to traditional unigrams, but do not get significant improvements. In the experiments of Joshi and Rosé (2009), the improvements of unigrams together with WR features (even generalized WR features) are also not remarkable (sometimes even worse) compared to simple unigrams.

One possible explanation might be that different types of features have distinct distributions, and therefore would probably yield vary performance on different machine learning algorithms. For example, the generative model is optimal if the distribution is well estimated; otherwise the performance will drop significantly (for instance, NB performs poorly unless the feature independence assumption holds well). While on the contrary, the discriminative model such as SVM is good at representing the complexity of relevant features.

Let us review the results reported by Pang and Lee (2002) that compare different classification algorithms: SVM performs significantly better than NB on unigrams; while the outcome is the opposite on bigrams. It is possibly due to that from unigrams to bigrams, the relevance between features is reduced (bigrams cover some relevance of unigram pairs), and the independence between features increases.

Since GWR features are less relevant and more independent in comparison, it is reasonable for us to infer that these features would work better on NB than on SVM. We therefore intuitively seek to employ the ensemble model for sentiment classification tasks, with an attempt to efficiently integrate different types of features under distinct classification models.

3.2 Model Formulation

The ensemble model (Kittler, 1998), which combines the outputs of several base classifiers to form an integrated output, has become an effective classification method for many domains.

For our ensemble task, we train six base classifiers (the NB and SVM model respectively on the Uni, GWR-Bi and GWR-Dp features). By mapping the probabilistic outputs (for C classes) of D base classifiers into the meta-vector

$$\hat{\mathbf{x}} = [o_{11},\ldots o_{1C},\ldots,o_{kj},\ldots,o_{D1},\ldots o_{DC}], \qquad (1)$$

the weighted ensemble is formulized by

$$O_j = g_j(\hat{\mathbf{x}}) = \sum_{k=1}^{D} \omega_k o_{kj} = \sum_{k=1}^{D} \omega_k \hat{x}_{k\times D+j}, \qquad (2)$$

where ω_k is the weight assigned to the k-th base classifier.

3.3 Weight Optimization

Inspired by linear regression, we use descent methods to seek optimization according to certain criteria. We employ two criteria, namely the perceptron criterion and the minimum classification error (MCE) criterion.

The perceptron cost function is defined as

$$J_p = \frac{1}{N}\sum_{i=1}^{N}\left[\max_{j=1,\ldots,C} g_j(\hat{\mathbf{x}}_i) - g_{y_i}(\hat{\mathbf{x}}_i)\right]. \qquad (3)$$

The minimization of J_p is approximately equal to seek a minimum misclassification rate.

The MCE criterion (Juang and Katagiri, 1992) is supposed to be more relevant to the classification error. A short version of MCE criterion function is given by

$$J_{mce} = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C} I(y_i = j)\delta(-g_j(\hat{\mathbf{x}}) + \max_{k \neq j} g_k(\hat{\mathbf{x}})) \quad (4)$$

where $\delta(\bullet)$ is the sigmoid function.

For both criteria, stochastic gradient descent (SGD) is utilized for optimization. SGD uses approximate gradients estimated from subsets of the training data and updates the parameters in an online manner:

$$\omega_h(k+1) = \omega_h(k) - \eta(k)\frac{\partial J}{\partial \omega_h}. \quad (5)$$

The gradients of perceptron and MCE cost functions are respectively

$$\frac{\partial J_{\mathrm{p}}}{\partial \omega_h} = -\frac{1}{N}\sum_{i=1}^{N}(\hat{x}_{h\times D+s_i} - \hat{x}_{h\times D+y_i}) \quad (6)$$

where $s_i = \arg\max_{j=1,\ldots,C} g_j(\hat{\mathbf{x}}_i)$, and

$$\frac{\partial J_{\mathrm{MCE}}}{\partial \omega_h} = -\frac{1}{N}\sum_{i=1}^{N} l_{y_i}(\hat{\mathbf{x}}_i)(1 - l_{y_i}(\hat{\mathbf{x}}_i))(\hat{x}_{h\times D+s_i} - \hat{x}_{h\times D+y_i}) \quad (7)$$

where $l_j(\hat{\mathbf{x}}_i) = \delta(-g_{y_i}(\hat{\mathbf{x}}_i) + \max_{h \neq j} g_k(\hat{\mathbf{x}}_i))$ and $s_i = \arg\max_{j=1,\ldots,C; j\neq y_i} g_j(\hat{\mathbf{x}}_i)$.

As for perceptron criterion, we employ the average perceptron (AvgP) (Freund and Schapire, 1999), a variation of perceptron model that averages the weights of all iteration loops, to improve the generalization performance.

4 Feature Selection for WR Features

In the past decade, feature selection (FS) studies mainly focus on topical text classification. (Yang and Pedersen, 1997) investigate five FS metrics and reported that good FS methods (such as IG and CHI) can improve the categorization accuracy with an aggressive feature removal. In sentiment classification tasks, traditional FS methods were also proven to be effective (Ng et al., 2006; Li et al., 2009).

With regard to WR features, since the dimension of feature space has sharply increased, the amount of computation is considerably large when employing traditional FS methods.

4.1 Fast MI and Fast IG

In order to address this problem, we propose a fast feature selection method that is specially designed for GWR features. In our method, the importance of a GWR feature *ws* (e.g., *great-movie*) is considered as two component parts: the non-back-off word *w* (*great*) and the POS pairs *s* (J-N). We calculate the score of *w* and *s* respectively using existing FS methods, and take the sum of them as the final score. By assuming the two parts are mutually independent, the importance of a relation feature can be taken separately. We now give a theoretical support.

First, the mutual information between a relation feature ws and class c_k is defined as

$$I(ws, c_k) = \log\frac{P(ws, c_k)}{P(ws)P(c_k)}. \quad (8)$$

If *w* and *s* are independent, they are conditionally independent. Thus we have

$$\begin{aligned} I(ws, c_k) &= \log\frac{P(ws \mid c_k)}{P(ws)} \\ &\approx \log\frac{P(w \mid c_k)P(s \mid c_k)}{P(w)P(s)} \\ &= \log\frac{P(w \mid c_k)}{P(w)} + \log\frac{P(s \mid c_k)}{P(s)} \\ &= I(w, c_k) + I(s, c_k). \end{aligned} \quad (9)$$

Formula (9) indicates that under the assumption that two component parts *w* and *s* of a relation feature *ws* are mutually independent, the mutual information of the relation feature $I(ws, c_k)$ equals the sum of two component parts $I(w, c_k)$ and $I(s, c_k)$.

Since the average mutual information across all classes $I(ws)$ is the probabilistic sum of each class, it can be written as:

$$I(ws) \approx I(w) + I(s). \quad (10)$$

Yang and Pedersen (1997) show that the information gain $G(t)$ is the weighted average of $I(t, c_k)$ and $I(\overline{t}, c_k)$. Therefore, with the same reason, we can consider the information gain of a relation feature $G(ws)$ as the sum of two component parts:

$$G(ws) \approx G(w) + G(s) \quad (11)$$

We refer to Formula (10) and (11) as fast MI (FMI) and fast IG (FIG) respectively. Now let us look back at the rationality of the independence assumption. In fact in a relation feature, two component parts are hardly independent since they are "related". Nonetheless, if we con-

sider a GWR feature as a combination of the non-back-off word and the POS pairs, the assumption will be easier to satisfy. Taking *great-movie* (*great*-N) for example, compared to *great* and N, *great* and J-N are more independent (J-N covers some relation information), therefore it is more feasible to take $G(great)+G(\text{J-N})$ as an approximation of $G(great\text{-N})$.

Laying aside the assumption, we place emphasis on the advantage of FIG (FMI) in computational efficiency. Assuming the dimension of the unigrams feature space is N, and ignoring the data-sparse problem, the dimension of the GWR feature space is $2\times5\times N$ (backing off head/modifier word to 5 POS-cluster). Traditional IG (MI) feature selection needs to calculate the score of all $10\times N$ features, while FIG (FMI) only needs to compute for N words and 25 POS pairs. That is to say, FIG (FMI) can speed up the computation of traditional IG (MI) by at most 10 times.

4.2 Integration with the Ensemble Model

We now present how FMI (FIG) is applied to the ensemble model described in section 3.2. In each of the six base-classifiers described in Section 3.2, feature selection is performed (traditional IG on unigrams, FIG on GWR features).

Note that when performing FIG on individual GWR feature sets, the computation of non-back-off word $G(w)$, is taken care of by having already computed IG on unigrams. Thus, we only need to compute the score of 25 POS pairs. From this point of view, FIG (FMI) is quite suitable for the ensemble model.

5 Experiments

We first present the performance of system performance, and then demonstrate the effectiveness of fast feature selection.

5.1 Experimental Setup

Datasets: The Cornell movie-review dataset[1] introduced by (Pang and Lee, 2004) is used in our experiments. It is a document-level polarity dataset that contains 1,000 positive and 1,000 negative processed reviews.

We also use the dataset[2] introduced in (Joshi and Penstein-Rosé, 2009) for comparison. It is a subset (200 sentences each for 11 different products) of the product review dataset released by (Hu and Liu, 2004). We will refer to it E-product dataset.

The Movie dataset is a domain-specific document-level dataset and the E-product dataset is at sentence-level and cross-domain. We conduct experiments on both of them to evaluate our approach in a wide range of tasks.

Classifier: We implement the NB classifier based on a multinomial event model (McCallum and Nigam, 1998) with Laplace smoothing. The tool LIBSVM[3] is chosen as the SVM classifier. Setting of kernel function is linear kernel, the penalty parameter is set to one, and the Platt's probabilistic output for SVM is applied to approximate the posterior probabilities. Term presence is used as the feature weighting.

Implementation: The Movie dataset is evenly divided into 5 folds, and all the experiments are conducted with a 5-fold cross validation. Following the settings by Joshi and Rosé, an 11-fold cross validation is applied to E-product dataset, where each test fold contains all the sentences for one of the 11 products, and the sentences for the remaining 10 products are used for training.

For ensemble learning, the stacking framework (Džeroski and Ženko, 2004) is employed. Taking the Movie dataset for example, in each loop of the 5-fold cross validation, the probabilistic outputs of the test fold are considered as test samples for ensemble leaning; and an inner 4-fold leave-one-out procedure is applied to the training data, where samples in each fold are trained on the remaining three folds to obtain the probabilistic outputs which serve as training samples for ensemble learning.

All the performance in the remaining tables and figures is in terms of average accuracy.

5.2 Results of Classification Accuracy

The results of classification accuracy are organized in three parts. We first compare the performance of individual WR and GWR; secondly we compare joint features and the ensemble

[1] http://www.cs.cornell.edu/people/pabo/movie-review-data/

[2] http://www.cs.cmu.edu/~maheshj/datasets/acl09short.html

[3] http://www.csie.ntu.edu.tw/~cjlin/libsvm/

model; thirdly we compare different ensemble strategies; finally we make a comparison with some related work.

5.2.1 WR vs. GWR

Table 3 presents the results of individual WR feature sets. Four types of WR features, including WR-Bi, WR-Dp, GWR-Bi and GWR-Dp, are examined under two classification models on two datasets. For each of the results, we report the best accuracy under feature selection.

Model	WR Feature	Movie	E-product
SVM	WR-Bi	83.05	63.27
	GWR-Bi	85.55	65.17
	WR-Dp	82.15	65.14
	GWR-Dp	83.40	67.09
NB	WR-Bi	84.60	66.86
	GWR-Bi	85.45	67.50
	WR-Dp	83.90	65.68
	GWR-Dp	83.65	67.41

Table 3: Accuracies (%) of Individual WR Feature Sets

At first, we place the emphasis on the performance of individual GWR and WR. With the SVM model, the performance of GWR features is remarkable compared to traditional WR pairs. Specifically, on the Movie dataset, GWR-Bi outperforms WR-Bi by 2.50%, and GWR-Dp outperforms WR-Dp by 1.35%; on the E-product dataset, the improvements are 1.90% and 1.95%. Under the NB model, on the Movie dataset, GWR-Bi outperforms WR-Bi by 0.85%; on the E-product dataset, GWR-Bi outperforms WR-Bi by 0.64% and GWR-Dp outperforms WR-Dp by 1.73%. One exception is GWR-Dp on the Movie dataset, but the decline is slight (0.25%).

WR Feature	Movie	E-product
WR-Bi	386k	21k
GWR-Bi	152k	16k
WR-Dp	455k	24k
GWR-Dp	151k	16k

Table 4: Dimension of Individual Feature Space

Secondly, we compare the dimensions of different feature space. Table 4 presents the average size of different types of feature spaces on two datasets. On the Movie dataset, the size of GWR feature space has been significantly reduced (386k vs. 152k in Bi; 455k vs. 151k in Dp). On the E-product dataset, since the training set are made up by 10 different domains, data are quite sparse, therefore, the extent of dimension reduction is not as sound as that on Movie dataset, but still considerable (21k vs. 16k in Bi; 24k vs. 16k in Dp).

5.2.2 Joint Features vs. Ensemble Model

The performance of individual feature sets, joint feature set and ensemble model is reported in Table 5. Uni, GWR-Bi and GWR-Dp are used as individual features sets in the ensemble model, and Joint Features denote the union of three individual sets. For feature selection, IG is used in Joint Features, and FIG is used in the ensemble model. The reported results are in terms of the best accuracy under feature selection.

Feature and Model		Movie	E-product
Uni	SVM	85.20	67.77
	NB	84.10	66.18
GWR-Bi	SVM	85.55	65.17
	NB	85.45	67.50
GWR-Dp	SVM	83.40	67.09
	NB	83.65	67.41
Joint Features	SVM	86.10	66.55
	NB	85.20	67.64
Ensemble Model	AvgP	**88.60**	70.14
	MCE	88.55	**70.18**

Table 5: Accuracies (%) of Component Features, Joint Features and Ensemble Model

To begin with, we observe the results of individual feature sets. Although we have demonstrated that GWR features are more effective than WR, it is a pity that they do not show significant superiority (sometimes even worse) compared to unigrams. That is to say, although GWR features encode more generalized word relation information than WR features, the role of unigrams still can not be replaced. This is in accordance with that, WR (GWR) features are used as additional features to assist unigrams in most of the literature.

Secondly, we focus on the performance of two classification models on different feature sets. SVM seems to work better than NB on unigrams (more than 1%); while on GWR-Bi and GWR-Dp feature sets, NB tends to be overall effective. This has confirmed our speculation that WR features perform better under NB than under SVM (since independence between features increases) and strengthened the confidence

of our motivation to ensemble different types of features under distinct classification models.

Finally, we make a comparison of Joint Features and Ensemble model. Observing the results on the Movie dataset, Joint Features exceed individual feature sets, but the improvements are not remarkable (less than 1 percentage compared to the best individual score). While the results of the ensemble model, as we have expected, are fairly good. AvgP and MCE respectively get the scores of 0.886 and 0.8855, robustly higher than that of Joint Features (0.8610 and 0.8520 respectively under SVM and NB).

On the E-product dataset, it is quite surprising that the result of Joint Features is even worse than some of the individual features sets. This also confirms that Joint Features are sometimes not so effective at exploring different types of features. With regard to the ensemble model, AvgP gets an accuracy of 0.7014 and MCE achieves the best score (0.7018), consistently superior to the results of Joint Features.

5.2.3 Different Ensemble Strategies

We also examine the performance of different strategies. In Table 6, three ensemble strategies are compared, where "(Uni & Bi & Dp) @ SVM" denotes ensemble of three kinds of feature sets with the fixed SVM classifier, "Uni @ (NB & SVM)" denotes ensemble of two classifiers on fixed unigram features, and "(Uni & Bi & Dp) @ (NB & SVM)" denotes ensemble of both classifiers and feature sets.

Ensemble Strategy		Movie	E-product
(Uni & Bi & Dp) @ SVM	AveP	86.60	69.50
	MCE	86.60	69.59
Uni @ (NB & SVM)	AveP	87.75	68.95
	MCE	87.80	69.14
(Uni & Bi & Dp) @ (NB & SVM)	AveP	**88.60**	70.14
	MCE	88.55	**70.18**

Table 6: Accuracies (%) of Different Ensemble Strategies.

Seen from Table 5 and 6, the performance of ensemble of either feature sets or classifiers is robustly better than any individual classifier, as well as the joint features on both datasets. With regard to ensemble of both feature sets and classification algorithms, it is the most effective compared to the above two ensemble strategies. This is in accordance with our motivation described in Section 3.1.

5.2.4 Comparison with Related Work

We take the performance of SVM on unigrams as the baseline for comparison. On the Movie dataset, Pang and Lee (2004) and Ng et al. (2006) reported the baseline accuracy of 0.871. But our baseline is 2 percentages lower (0.852). It is mainly due to that: 1) 0.871 was obtained by a 10-fold cross validation, and our result is get by 5-fold cross validation; 2) the result of the tool LibSVM is inferior of SVM^{light} by almost 1-2 percentages, since the penalty parameter in LibSVM is fixed, while in SVM^{light}, the value is automatically adapted; 3) the baseline in Ng et al. (2006) is obtained with length normalization which play a role in performance.

Ng et al. reported the state of art best performance (0.905), which outperforms the baseline (0.871) by 3.4%. Our best result of ensemble model (0.886) gets a comparable improvement (3.40%) compared to our obtained baseline (0.852).

On the E-product dataset, Joshi and Rosé reported the best result (0.679) on joint features of unigrams and their proposed GWR features. This is in accordance with our result of Joint Features (0.6655 by SVM and 0.6764 by NB). The superiority of our ensemble result is quite significant (0.7014 by AvgP and 0.7018 by MCE).

5.3 Results of Feature Selection

In this part, we examine FMI and FIG for GWR feature selection. The performance of MI and IG are also presented for comparison. The results on the Movie and E-product datasets are displayed in Figures 1 and 2 respectively. Due to space limit, we only report the results of GWR-Bi features for Movie and GWR-Dp features for E-product. In each of the figures, the results under NB and SVM are both presented.

At first, we observe the results of feature selection for GWR-Bi features on the Movie dataset. At first glance, IG and FIG have roughly the same performance. IG-based methods are shown to be quite effective in GWR feature reduction. For example under the NB model, top 2.5% (4000) GWR-Bi features ranked by IG and FIG achieve accuracies of 0.849 and 0.842

respectively, even better than the score with all features (0.8415).

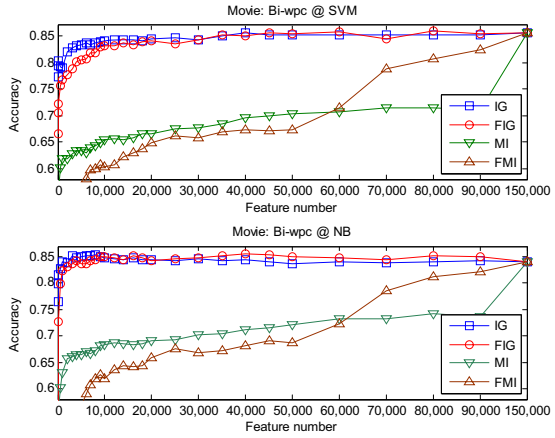


Figure 1: Feature Selection for GWR-Bi Features on the Movie Dataset

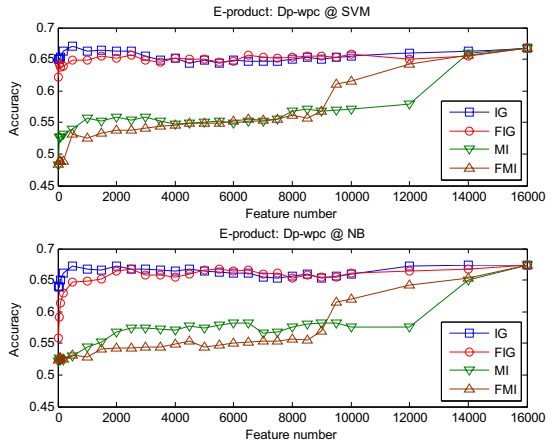


Figure 2: Feature Selection for GWR-Dp features on the E-product dataset

We then observe IG vs. FIG in a finer granularity. When the selected features are few (less than 5%), IG performs significantly better than FIG, while the latter gradually approaches the former when the feature number increases: as it comes to 10-15%, their performance is quite close. From then on, FIG is consistently comparable to IG, even sometimes slightly better.

With regard to MI and FMI, although the performance compared to IG and FIG is rather poor (the reason has been intensively studied by Yang and Pedersen, 1997). Our focus is the ability of FMI for approximating MI. From this point of view, FMI is by contrast effective, especially with more than 1/3 features.

Compared to the Movie dataset, the size of E-product dataset is much smaller, and the data are much sparser. Nevertheless, IG and FIG are still effective. On one hand, top 1.25% (2000) features ranked by IG yield a result better than (or comparable to) that with all features. On the other hand, FIG is still competent to be a good approximation to IG.

All of the above comparisons are made according to accuracies, and we now pay attention to computational efficiency. Taking the Movie dataset for example, IG needs to compute scores of information gain for all $152k$ features, while FIG only needs to compute $42k+5\times5$ scores, saving more than 70% of the computational load; on the E-product dataset, although the data are sparse, the rate of computation reduction is still significant (62.5%).

Note that in the ensemble model, when performing FIG for individual GWR feature set, part of its inherent complexity is already taken care of by having already computed IG on Uni feature set, and we only need to compute the scores for 25 POS pairs. From this perspective, FIG is even more attractive in the ensemble model.

6 Conclusions and Future Work

The focus of this paper is exploring the use of WR features for sentiment classification. We have proposed a GWR feature extraction approach and an ensemble model to efficiently integrate different types of features. Moreover, we have proposed two fast feature selection methods (FMI and FIG) for GWR features.

Individual GWR features outperform traditional WR features significantly, but they still can not totally substitute unigrams. The ensemble model is quite effective at integrating unigrams and different types of WR feature, and the performance is significantly better than joint features.

FIG is proved to be a good solution for selecting GWR features. It is also worthy noting that FIG is a general feature selection method for bigram features, even outside the scope of sentiment classification and text classification.

In the future, we plan to make an in-depth study about why individual WR features are inferior to unigrams, and how to make the joint features more effective. We also plan to extend the use of GWR features to the task of transfer learning, which we think is a promising direction for future work.

Acknowledgment

We thank Yufeng Chen, Shoushan Li, Ping Jian and the anonymous reviewers for valuable comments and helpful suggestions. The research work has been partially funded by the Natural Science Foundation of China under Grant No. 60975053, 90820303 and 60736014, the National Key Technology R&D Program under Grant No. 2006BAH03B02, the Hi-Tech Research and Development Program ("863" Program) of China under Grant No. 2006AA010108-4, and also supported by the China-Singapore Institute of Digital Media (CSIDM) project under grant No. CSIDM-200804.

References

Kushal Dave, Steve Lawrence and David M. Pennock, 2003. Mining the Peanut Gallery: Opinion Extraction and Semantic Classification of Product Reviews. In Proceedings of the international World Wide Web Conference (WWW), pages 519-528.

Sašo Džeroski and Bernard Ženko, 2004. Is combining classifiers with stacking better than selecting the best one? Machine Learning, 54 (3). pages 255-273.

Yoav Freund and Robert E. Schapire, 1999. Large margin classification using the perceptron algorithm. Machine Learning, 37 (3). pages 277-296.

Michael Gamon, 2004. Sentiment classification on customer feedback data: noisy data, large feature vectors, and the role of linguistic analysis. In Proceedings of the International Conference on Computational Linguistics (COLING). pages 841-847.

Minqing Hu and Bing Liu, 2004. Mining and summarizing customer reviews. In Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD), pages 168-177.

Mahesh Joshi and Carolyn Penstein-Rosé, 2009. Generalizing dependency features for opinion mining. In Proceedings of the Joint Conference of the 47th Annual Meeting of the Association for Computational Linguistics (ACL), pages 313-316.

Biing-Hwang Juang and Shigeru Katagiri, 1992. Discriminative learning for minimum error classification. IEEE Transactions on Signal Processing, 40 (12). pages 3043-3054.

J Kittler, 1998. Combining classifiers: A theoretical framework. Pattern Analysis and Applications, 1 (1). pages 18-27.

Shoushan Li, Rui Xia, Chengqing Zong and Chu-Ren Huang, 2009. A framework of feature selection methods for text categorization. In Proceedings of the Joint Conference of the 47th Annual Meeting of the Association for Computational Linguistics (ACL), pages 692-700.

Andrew McCallum and Kamal Nigam, 1998. A comparison of event models for naive bayes text classification. In Proceedings of the AAAI workshop on learning for text categorization.

Vincent Ng, Sajib Dasgupta and S. M. Niaz Arifin, 2006. Examining the Role of Linguistic Knowledge Sources in the Automatic Identification and Classification of Reviews. In Proceedings of the COLING/ACL, pages 611-618.

Bo Pang and Lillian Lee, 2004. A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts. In Proceedings of the Association for Computational Linguistics (ACL), pages 271-278.

Bo Pang and Lillian Lee, 2008. Opinion mining and sentiment analysis. Foundations and Trends in Information Retrieval, 2 (1-2). pages 1-135.

Bo Pang, Lillian Lee and Shivakumar Vaithyanathan, 2002. Thumbs up? Sentiment Classification using Machine Learning Techniques. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 79-86.

Yiming Yang and Jan O. Pedersen, 1997. A comparative study on feature selection in text categorization. In Proceedings of the 14th International Conference on Machine Learning (ICML), pages 412-420.

An Empirical Study of Translation Rule Extraction with Multiple Parsers

Tong Xiao[†‡], Jingbo Zhu[†‡], Hao Zhang[†], Muhua Zhu[†‡]

[†]Natural Language Processing Lab., Northeastern University
[‡]Key Laboratory of Medical Image Computing, Ministry of Education
{xiaotong,zhujingbo}@mail.neu.edu.cn
zhanghao@ics.neu.edu.cn, zhumuhua@gmail.com

Abstract

Translation rule extraction is an important issue in syntax-based Statistical Machine Translation (SMT). Recent studies show that rule coverage is one of the key factors affecting the success of syntax-based systems. In this paper, we first present a simple and effective method to improve rule coverage by using multiple parsers in translation rule extraction, and then empirically investigate the effectiveness of our method on Chinese-English translation tasks. Experimental results show that extracting translation rules using multiple parsers improves a string-to-tree system by over 0.9 BLEU points on both NIST 2004 and 2005 test corpora.

1 Introduction

Recently various syntax-based models have been extensively investigated in Statistical Machine Translation (SMT), including models between source trees and target strings (Quirk et al., 2005; Liu et al., 2006; Huang et al., 2006), source strings and target trees (Yamada and Knight, 2001; Galley et al., 2006; Shen et al., 2008), or source trees and target trees (Eisner, 2003; Ding and Palmer, 2005; Cowan et al., 2006; Zhang et al., 2008; Liu et al., 2009). In these models, automatic extraction of translation rules is an important issue, in which translation rules are typically extracted using parse trees on source/target-language side or both sides of the bilingual text. Exploiting the syntactic information encoded in translation rules, syntax-based systems have shown to achieve comparable performance with phrase-based systems, even outperform them in some cases (Marcu et al., 2006).

Among all the factors contributing to the success of syntax-based systems, *rule coverage* has been proved to be an important one that affects the translation accuracy of syntax-based systems (DeNeefe et al., 2007; Shen et al., 2008). However, these systems suffer from a problem that translation rules are extracted using only 1-best parse tree generated by a single parser, which generally results in relatively low rule coverage due to the limited scope in rule extraction (Mi and Huang, 2008). To alleviate this problem, a straightforward solution is to enlarge the scope of rule extraction, and obtain translation rules by using a group of diversified parse trees instead of a single parse tree. For example, Mi and Huang (2008) used *k*-best parses and forest to extract translation rules for improving the rule coverage in their forest-based SMT system, and achieved promising results. However, most previous work used the parse trees generated by only one parser, which still suffered somewhat from the relatively low diversity in the outputs of a single parser.

Addressing this issue, we investigate how to extract diversified translation rules using multiple parsers. As different parsers (or parsing models) can provide us with parse trees having relatively large diversity, we believe that it is beneficial to employ multiple different parsers to obtain diversified translation rules and thus enlarge the rule coverage. Motivated by this idea, we propose a simple and effective method to improve rule coverage by using multiple parsers

Coling 2008: Poster Volume, pages 1345–1353,
Beijing, August 2010

in rule extraction. Furthermore, we conduct an empirical study to investigate the effectiveness of our method on Chinese-English translation in a string-to-tree system. Experimental results show that our method improves the baseline system by over 0.9 BLEU points on both NIST 2004 and 2005 test corpora, even achieves a +1 BLEU improvement when working with the *k*-best extraction method. More interestingly, we observe that the MT performance is not very sensitive to the parsing performance of the parsers used in rule extraction. Actually, the MT system does not show different preferences for different parsers.

2 Related Work

In machine translation, some efforts have been made to improve rule coverage and advance the performance of syntax-based systems. For example, Galley et al. (2006) proposed the idea of rule composing which composes two or more rules with shared states to form a larger, composed rule. Their experimental results showed that the rule composing method could significantly improve the translation accuracy of their syntax-based system. Following Galley et al. (2006)'s work, Marcu et al. (2006) proposed SPMT models to improve the coverage of phrasal rules, and demonstrated that the system performance could be further improved by using their proposed models. Wang et al. (2007) described a binarization method that binarized parse trees to improve the rule coverage on non-syntactic mappings. DeNeefe et al. (2007) analyzed the phrasal coverage problem, and compared the phrasal coverage as well as translation accuracy for various rule extraction methods (Galley et al., 2006; Marcu et al., 2006; Wang et al., 2007).

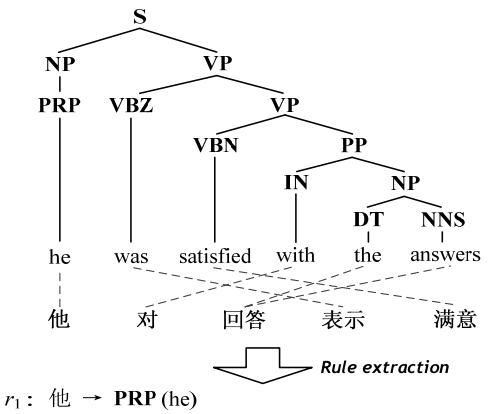
As another research direction, some work is focused on enlarging the scope of rule extraction to improve rule coverage. For example, (Venugopal et al., 2008) and (Mi and Huang, 2008) extracted rules from the *k*-best parses and forest generated by a single parser to alleviate the problem of the limited scope of 1-best parse, and achieved promising results.

Our work differs from previous work in that we are concerned with obtaining diversified translation rules using multiple different parsers (or parsing models) instead of a single parser (or parsing model). It can be regarded as an enhancement of previous studies. As shown in the following parts of this paper, it works very well with the existing techniques, such as rule composing (Galley et al., 2006), SPMT models (Marcu et al., 2006) and rule extraction with *k*-best parses (Venugopal et al., 2008).

3 Translation Rule Extraction

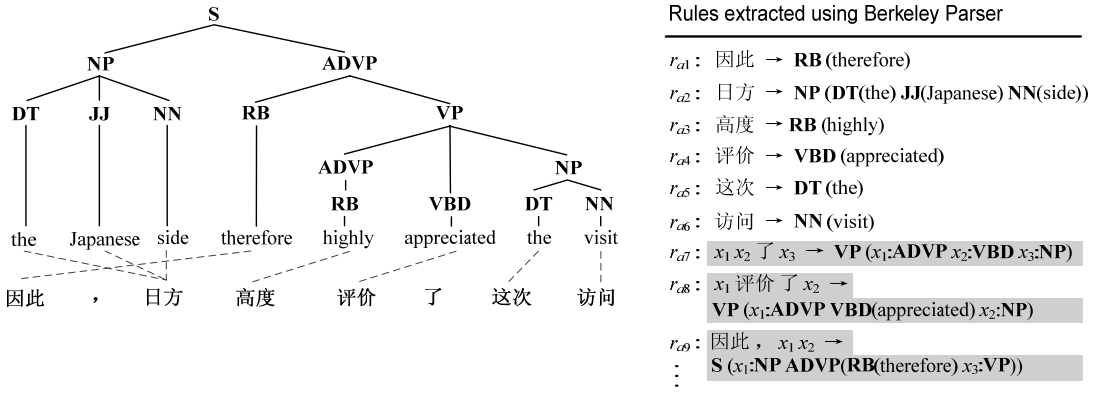
In this work, the issue of translation rule extraction is studied in the string-to-tree model proposed by Galley et al. (2006). We choose this model because it has been shown to be one of the state-of-the-art syntax-based models, and has been adopted in the most successful systems in NIST 2009 MT evaluation.

Typically, (string-to-tree) translation rules are learned from the word-aligned bilingual text whose target-side has been parsed using a syntactic parser. As the basic unit of translation, a translation rule consists of sequence words or variables in the source language, and a syntax tree in the target language having words (terminals) and variables (non-terminals) at leaves. Figure 1 shows the translation rules extracted from a word-aligned sentence pair with a target-side parse tree.

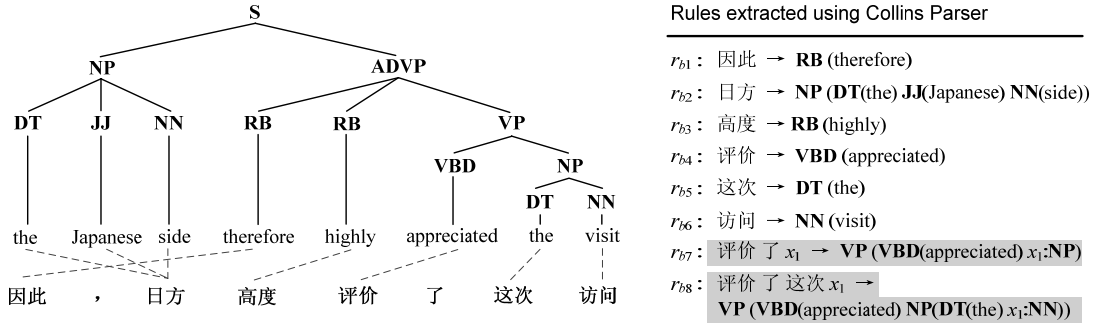


r_1: 他 → **PRP** (he)
r_2: 对 → **IN** (with)
r_3: 回答 → **NP** (**DT**(the) **NNS**(answers))
r_4: 表示 → **VBZ** (was)
r_5: 满意 → **VBN** (satisfied)
r_6: $x_1\,x_2$ → **PP** (x_1:**IN** x_2:**NP**)
r_7: 对 x_1 → **PP** (**IN**(with) x_1:**NP**)
r_8: $x_1\,x_2$ 表示 满意 →
S (x_1:**NP** **VP**(**VBZ**(was) **VP**(**VBN**(satisfied) x_2:**PP**)))
⋮

Figure 1: Translation rules extracted from a string-tree pair.



(a) rule extraction using Berkeley Parser



(b) rule extraction using Collins Parser (Model 2)

Figure 2: Rule extraction using two different parsers (Berkeley Parser and Collins Parser). The shaded rectangles denote the translation rules that can be extracted from the parse tree generated by one parser but cannot be extracted from the parse tree generated by the other parser.

To obtain basic translation rules, the (minimal) GHKM extraction method proposed in (Galley et al, 2004) is utilized. The basic idea of GHKM extraction is to compute the set of the minimally-sized translation rules that can explain the mappings between source-language string and target-language tree while respecting the alignment and reordering between the two languages. For example, from the string-tree pair shown at the top of Figure 1, we extract the minimal GHKM translation rules $r_{1\text{-}6}$. In addition to GHKM extraction, the SPMT models (Marcu et al., 2006) are employed to obtain *phrasal rules* that are not covered by GHKM extraction. For example, rule r_8 in Figure 1 is a SPMT rule that is not obtained in GHKM extraction. Finally, the rule composing method (Galley et al., 2006) is used to compose two or more minimal GHKM or SPMT rules having shared states to form larger rules. For example, rule r_7 in Figure 1 is generated by composing rules r_2 and r_6.

4 Differences in Coverage between Rule Extractions with Different Parsers

As described above, translation rule extraction relies on the outputs (parse trees) of parsers. As different parsers generally have large diversity between their outputs, rule extractions with different parsers generally result in very different sets of rules. For example, Figure 2 shows the rule extractions on a word-aligned sentence pair having two target-trees generated by Berkeley Parser and Collins Parser, respectively. It is observed that Figure 2 (a) and (b) cover different sets of rule due to the different target-trees used in rule extraction. Particularly, well-formed rules $r_{a7\text{-}a9}$ are extracted in Figure 2 (a), while they do not appear in Figure 2 (b). Also, rules $r_{b7\text{-}b9}$ in Figure 2 (b) have the similar situation. This observation gives us an intuition that there is a "complementarity" between the rules extracted using different parsers.

We also conduct a quantitative study to investigate the impact of using different parsers (Berkeley Parser and Collins Parser) on rule coverage. Tables 1 shows the statistics of the rules extracted from 370K Chinese-English parallel sentence pairs[1] using the method described in Section 3. In addition to the total number of rules extracted, the numbers of *phrasal rules* and *useful rules* are also reported to indicate the rule coverage of a rule set. Here *phrasal rule* refers to the rule whose source-side and the yield of its target-side contains only one phrase each, with optional surrounding variables. According to (DeNeefe et al., 2007), the number of phrasal rules is a good indicator of the coverage of a rule set. *useful rule* refers to the rule that can be applied when decoding the test sentences[2]. As shown in Table 1, the two resulting rule sets only have about 70% overlaps (Column 4), and the rule coverage increases by about 20% when we combine them together (Column 5). This finding confirms that the rule coverage can be improved by using multiple different parsers in rule extraction.

	# of rules	# of phrasal rules	# of useful rules
Berkeley	3,538,332	2,515,243	549,783
Collins	3,526,166	2,481,195	553,893
Overlap	2,542,380	1,907,521	386,983
Union	4,522,118	3,088,920	716,693

Table 1: Comparison of rule coverage between different rule sets.

5 Translation Rule Extraction with Multiple Parsers

5.1 Rule Extraction Algorithm

Motivated by the above observations, we propose a rule extraction method to improve the rule coverage by using multiple parsers.

Let $\langle f, e, a \rangle$ be a tuple of <source sentence, target sentence, bi-directional word alignments>, and $\{P_1, ..., P_N\}$ be N syntactic parsers in target-language. The following pseudocode formulizes the algorithm for extracting translation rules from $\langle f, e, a \rangle$ using parsers $\{P_1, ..., P_N\}$, where $P_i(e)$ returns the parse tree generated by the i-th parser P_i. Function GENERATERULES() computes the set of rules for $\langle f, t_i, a \rangle$ by using various rule extraction methods, such as the method described in Section 3.

Multi-Parser based Rule Extraction

Input: $\langle f, e, a \rangle$ and $P = \{P_1, ..., P_N\}$
Output: rule set R

Function MULTIPARERXTRACTOIN($\langle f, e, a \rangle$, P)
  for $i = 1$ **to** N **do** ◁ for each parser
    $t_i = P_i(e)$ ◁ target-tree
    R_i = GENERATERULES (f, t_i, a) ◁ rule extraction
    R.append(R_i)
  return R
Function GENERATERULES (f, t_i, a)
  return rules extracted from $\langle f, t_i, a \rangle$

5.2 Learning Rule Probabilities

In multi-parser based rule extraction, more than one parse trees are used, and each of them is associated with a parsing confidence (e.g. generative probability of the tree). Ideally, if the parse trees used in rule extraction can be accurately weighted, the rule probabilities will be better estimated according to the parse weights, for example, the rules extracted from a parse tree having a low weight should be penalized accordingly in the estimation of rule probabilities. Unfortunately, the tree probabilities are generally incomparable between different parsers due to the different parsing models used and ways of implementation. Thus we cannot use the posterior probability of a rule's target-side to estimate the *fractional count* (Mi and Huang, 2008; Liu et al., 2009), which is used in maximum-likelihood estimation of rule probabilities. In this work, to simplify the problem, we assume that all the parsers have the same and maximum degrees of confidence on their outputs. For a rule r extracted from a string-tree pair, the count of r is defined to be:

$$c(r) = \frac{\sum_{i=1}^{N} \tau(r,i)}{N} \quad (1)$$

where $\tau(r,i)$ is 1 if r is extracted by using the i-th parser, otherwise 0.

[1] LDC2005T10, LDC2003E07, LDC2003E14 and LDC2005T06

[2] In this experiment, the test sentences come from NIST 2004 and 2005 MT evaluation sets. It should be noted that due to the pruning in decoding we cannot count the exact number of rules that can be used during decoding. In this work, we use an alternative – the number of rules matched with test sentences – to estimate an upper-bound approximately.

Following Mi and Huang (2008)'s work, three conditional rule probabilities are employed for experimenting with our method.

$$\Pr(r \mid root(r)) = \frac{c(r)}{\sum_{r':root(r')=root(r)} c(r)} \quad (2)$$

$$\Pr(r \mid lhs(r)) = \frac{c(r)}{\sum_{r':lhs(r')=lhs(r)} c(r)} \quad (3)$$

$$\Pr(r \mid rhs(r)) = \frac{c(r)}{\sum_{r':rhs(r')=rhs(r)} c(r)} \quad (4)$$

where $lhs(r)$ and $rhs(r)$ are the source-hand and target-hand sides of r respectively, and $root(r)$ is the root of r's target-tree.

5.3 Parser Indicator Features

For each rule, we define N indicator features (i.e. $\tau(r,i)$) to indicate a rule is extracted by using which parsers, and add them into the translation model. By training the feature weights with Minimum Error Rate Training (MERT), the system can learn preferences for different parsers automatically.

6 Experiments

The experiments are conducted on Chinese-English translation in a state-of-the-art string-to-tree SMT system.

6.1 Experimental Setup

Our bilingual data consists of 370K sentence pairs (9M Chinese words + 10M English words) which have been used in the experiment in Section 4. GIZA++ is employed to perform the bidirectional word alignment between the source and target sentences, and the final word alignment is generated using the inter-sect-diag-grow method. A 5-gram language model is trained on the target-side of the bilingual data and the Xinhua portion of English Gigaword corpus. The development data set comes from NIST MT 2003 evaluation set. To speed up MERT, sentences with more than 20 Chinese words are removed. The test sets are the NIST MT evaluation sets of 2004 and 2005.

Our baseline MT system is built based on the string-to-tree model proposed in (Galley et al., 2006). In this system, both of minimal GHKM (Galley et al., 2004) and SPMT rules (Marcu et al., 2006) are extracted from the bilingual corpus, and the composed rules are generated by composing two or three minimal GHKM and SPMT rules[3]. We use a CKY-style decoder with cube pruning (Huang and Chiang, 2007) and beam search to decode new Chinese sentences. By default, the beam size is set to 30. For integrating n-gram language model into decoding efficiently, rules containing more than two variables or source word sequences are binarized using the synchronous binarization method (Zhang et al., 2006; Xiao et al., 2009).

The system is evaluated in terms of the case-insensitive NIST version BLEU (using the shortest reference length), and statistical significant test is conducted using the re-sampling method proposed by Koehn (2004).

6.2 The Parsers

Four syntactic parsers are chosen for the experiments. They are Stanford Parser[4], Berkeley Parser[5], Collins Parser (Dan Bikel's reimplementation of Collins Model 2)[6] and Charniak Parser[7]. The former two are state-of-the-art non-lexicalized parsers, while the latter two are state-of-the-art lexicalized parsers. All the parsers are trained on sections 02-21 of the Wall Street Journal (WSJ) Treebank, and tuned on section 22. Table 2 summarizes the performance of the parsers.

Parser	Recall	Precision	F1
Stanford	86.29%	87.21%	86.75%
Berkeley	90.18%	90.45%	90.32%
Collins	89.14%	88.85%	88.99%
Charniak	89.99%	90.28%	90.13%

Table 2: Performance of the four parsers on section 23 of the WSJ Treebank.

We parse the target-side of the bilingual data using the four parsers individually. From the 1-best parses generated by these parsers, we obtain four baseline rule sets using the method described in Section 3, as well as the rule sets usi-

[3] Generally a higher baseline can be obtained by combining more (unit) rules. However, we find that using more composed rules does not affect the impact of using multiple parsers. Thus, we choose this setting in order to finish all experiments in time.

[4] http://nlp.stanford.edu/software/lex-parser.shtml

[5] http://code.google.com/p/berkeleyparser/

[6] http://www.cis.upenn.edu/~dbikel/download.html

[7] http://www.cs.brown.edu/people/ec/#software

	Rule set	Rule Coverage			BLEU4 (%)		
		# of rules	# of phrasal rules	# of useful rules	Dev.	MT04	MT05
Baseline	Stanford (S)	3,679 K	2,581 K	573 K	39.36	36.02	36.98
	Berkeley (B)	3,538 K	2,515 K	549 K	39.32	36.05	36.98
	Collins (Co)	3,526 K	2,481 K	553 K	39.16	36.07	36.91
	Charniak (Ch)	3,450 K	2,435 K	540 K	39.24	35.90	36.89
2 parsers	S + B	4,567 K	3,105 K	726 K	39.87+	36.57+	37.47+
	S + Co	4,734 K	3,202 K	752 K	39.94+	36.57+	37.52+
	S + Ch	4,764 K	3,258 K	751 K	40.01+	36.51	37.59+
	B + Co	4,522 K	3,088 K	716 K	39.84+	36.60+	37.46+
	B + Ch	4,562 K	3,129 K	717 K	39.81+	36.49	37.41
	Co + Ch	4,592 K	3,125 K	727 K	39.75	36.55+	37.43+
3 parsers	S + B + Co	5,331 K	3,543 K	852 K	40.14++	36.83++	37.78++
	S + B + Ch	5,380 K	3,590 K	854 K	40.05+	36.82++	37.70+
	S + Co + Ch	5,551 K	3,663 K	877 K	40.35++	36.70+	37.70+
	B + Co + Ch	5,294 K	3,544 K	840 K	40.04+	36.76+	37.65+
4	S + B + Co + Ch	6,005 K	3,940 K	958 K	40.28++	36.99++	37.89++

Table 5: Evaluation results. + or ++ = significantly better than all the baseline systems (using single parser) at the 95% or 99% confidence level.

	Stanford	Berkeley	Collins	Charniak
Stanford	100%	76.72%	73.32%	74.89%
Berkeley	76.72%	100%	75.69%	76.76%
Collins	73.32%	75.69%	100%	74.84%
Charniak	74.89%	76.76%	74.84%	100%

Table 3: Agreement between different parsers.

ng the multi-parser based rule extraction method. Before conducting primary experiments, we first investigate the differences between the 1-best outputs of the parsers. Table 3 shows the agreement between each pair of parsers. Here the degree of agreement shown in each cell is computed by using one parser's output as a good standard to evaluate the other parser's output in terms of F1 score, and a higher agreement score (i.e. F1 score) means that the 1-best outputs of the two parsers are more similar to each other. We see that the agreement scores between different parsers are always below 80%. This result reflects a large diversity in parse trees generated by different parsers, and thus confirms our observations in Section 4.

We also examine the "complementarity" between the baseline rule sets generated by using different parsers individually. Table 4 shows the results, where the degree of "complementarity" between two rule sets is defined as the percentage of the rules in one rule set that are not covered by the other rule set. It can be regarded as a measure of the disagreement between two rule sets, and a higher number indicates large "complementarity". For example, in Row 2, Column 3 (Table 4), "25.09%" means that 25.09% rules in the first rule set (using Stanford Parser) are not covered by the second rule set (using Berkeley Parser). Table 4 shows that there is always a disagreement of over 25% between different rule sets. These results indicate that using different parsers can lead to a relatively large "complementarity" between the rule sets.

	Stanford	Berkeley	Collins	Charniak
Stanford	0%	25.09%	29.91%	31.43%
Berkeley	27.98%	0%	27.90%	29.68%
Collins	32.84%	28.15%	0%	30.89%
Charniak	35.70%	31.43%	32.37%	0%

Table 4: Disagreement between the rule sets obtained using different parsers individually.

6.3 Evaluation of Translations

We then study the impact of multi-parser based rule extraction on translation accuracy. Table 5 shows the BLEU scores as well as the rule coverage for various rule extraction methods. We see, first of all, that the rule coverage is improved significantly by multi-parser based rule extraction. Compared to the baseline method (i.e. single-parser based rule extraction), the multi-parser based rule extraction achieves over 20% coverage improvements when only two parsers are used, even yields gains of over 50 percentage

points when all the four parsers are used together. Also, BLEU score is improved by multi-parser based rule extraction. When two parsers are employed in rule extraction, there is generally a gain of over 0.4 BLEU points on both MT04 and MT05 test sets. Further improvements are achieved when more parsers are involved. On both test sets, using three parsers in rule extraction generally yields a +0.7 BLEU improvement, and using all the parsers together yields a +0.9 BLEU improvement which is the biggest improvement achieved in this set of experiment. All these results show that multi-parser based rule extraction is an effective way to improve the rule coverage as well as the BLEU score of the syntax-based MT system.

An interesting finding is that there seems no significant differences in BLEU scores between the baseline systems (using single parsers), though the parsing performance of the corresponding parsers is very different from each other. For example, the MT performance corresponding to Berkeley Parser is very similar to that corresponding to Stanford Parser despite a 4-point difference in F1 score between the two parsers. Another example is that Charniak parser performs slightly worse than the other three on MT task, though it achieves the 2nd best parsing performance in all the parsers. This interesting finding shows that the performance of syntax-based MT systems is not very sensitive to the parsing performance of the parsers used in rule extraction.

6.4 Preferences for Parsers

We also investigate the preferences for different parsers in our system. Table 6 shows the weights of the parser indicator features learned by MERT, as well as the number of edges generated by applying the rules corresponding to different parsers during decoding. Both of the metrics are used to evaluate the contributions of the parsers to MT decoding. We see that though Stanford Parser and Berkeley Parser are shown to be relatively more preferred by the decoder, there are actually no significant differences in the degrees of the contributions of different parsers. This result also confirms the fact observed in Table 5 that the MT system does not have special preferences for different parsers.

Indicator	Weight	# of edges (Dev.)	# of edges (MT04)	# of edges (MT05)
Stanford	0.1990	7.7 M	169.2 M	101.7 M
Berkeley	0.1982	7.7 M	166.3 M	100.2 M
Collins	0.1690	6.9 M	149.9 M	93.1 M
Charniak	0.1729	7.1 M	156.5 M	97.2 M

Table 6: Preferences for different parsers.

Though Table 6 provides some information about the contributions of different parsers, it still does not answer how often these rules are really used to generate final (1-best) translation. Table 7 gives an answer to this question. We see that, following the similar trend in Table 5, different parsers have nearly equal contributions in generating final translation.

Indicator	# of rules used in 1-best (Dev.)	# of rules used in 1-best (MT04)	# of rules used in 1-best (MT05)
Stanford	2,410	23,513	14,357
Berkeley	2,455	23,878	14,670
Collins	2,309	22,654	13,815
Charniak	2,269	22,406	13,731

Table 7: Numbers of rules used in generating final (1-best) translation.

6.5 Rule Extraction with *k*-best Parses

We also conduct experiments to compare the effectiveness of multi-parser based rule extraction and rule extraction with *k*-best parses generated by a single parser. As Berkeley parser is one of the best-performing parsers in previous experiments, we employ it to generate *k*-best parses in this set of experiment. As shown in Figure 3, both of the methods improve the BLEU scores by enlarging the set of parse trees used in rule extraction. Compared to *k*-best extraction, multi-parser extraction shows consiste-

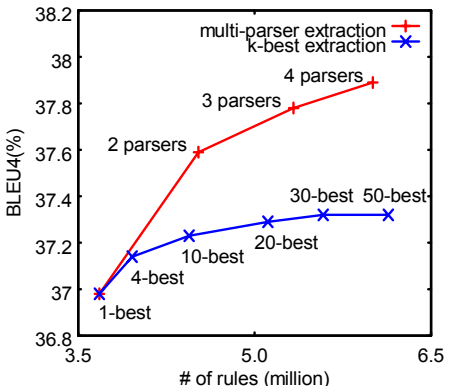


Figure 3: Multi-parser based rule extraction vs. rule extraction with *k*-best parses (MT05).

ntly better BLEU scores. Using 4 different parsers, it achieves an improvement of 0.6 BLEU points over *k*-best extraction where even 50-best parses are used.

Finally, we extend multi-parser based rule extraction to extracting rules from the *k*-best parses generated by multiple parsers. Figure 4 shows the results on "S + B + Co + Ch" system. We see that multi-parser based rule extraction can benefit from *k*-best parses, and yields a modest (+0.2 BLEU points) improvement when extracting from 10-best parses. However, since *k*-best extraction generally results in much slower extraction speed, it might not be a good choice to use *k*-best parses to improve our method in practice.

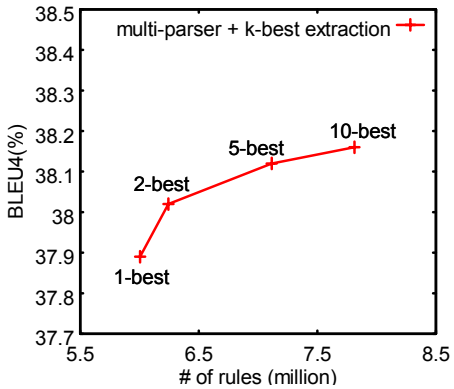


Figure 4: Multi-parser based rule extraction & rule extraction with *k*-best parses (MT05).

7 Discussion and Future Work

In this work, all the parsers are trained using the same treebank. To obtain diversified parse trees for multi-parser based rule extraction, an alternative way is to learn parsers on treebanks annotated by different organizations (e.g. Penn Treebank and ICE-GB corpus). Since different treebanks can provide us with more diversity in parsing, we believe that our system can benefit a lot from the parsers that are learned on multiple different treebanks individually. But here is a problem that due to the different annotation standards used, there is generally an incompatibility between treebanks annotated by different organizations. It will result in that we cannot straightforwardly mix the resulting rule sets (or *heterogeneous grammars* for short) for probability estimation as well as the use for decoding. To solve this problem, a simple solution might be that we transform the incompatible rules into a unified form. Alternatively, we can use *heterogeneous decoding* (or *parsing*) techniques (Zhu et al., 2010) to make use of heterogeneous grammars in the stage of decoding. Both topics are very interesting and worth studying in our future work.

Besides *k*-best extraction, our method can also be applied to other rule extraction schemes, such as forest-based rule extraction. As (Mi and Huang, 2008) has shown that forest-based extraction is more effective than *k*-best extraction in improving translation accuracy, it is expected to achieve further improvements by using multi-parser based rule extraction and forest-based rule extraction together.

8 Conclusions

In this paper, we present a simple and effective method to improve rule coverage by using multiple parsers in translation rule extraction. Experimental results show that

- Using multiple parsers in rule extraction achieves large improvements of rule coverage over the baseline method where only a single parser is used, as well as a +0.9 BLEU improvement on both NIST 2004 and 2005 test corpora.
- The MT system can be further improved by using multiple parsers and *k*-best parses together. However, with the consideration of extraction speed, it might not be a good choice to use *k*-best parses to improve multi-parser based rule extraction in practice.
- The MT performance is not influenced by the parsing performance of the parsers used in rule extraction very much. Actually, the MT system does not show different preferences for different parsers.

Acknowledgements

This work was supported in part by the National Science Foundation of China (60873091) and the Fundamental Research Funds for the Central Universities (N090604008). The authors would like to thank the anonymous reviewers and Tongran Liu for their pertinent comments for improving the early version of this paper, and Rushan Chen for building parts of the baseline system.

References

Brooke Cowan, Ivona Kučerová and Michael Collins. 2006. A discriminative model for tree-to-tree translation. In *Proc. of* EMNLP 2006, pages 232-241.

Steve DeNeefe, Kevin Knight, Wei Wang and Daniel Marcu. 2007. What Can Syntax-based MT Learn from Phrase-based MT? In *Proc. of* EMNLP 2007, pages 755-763.

Yuan Ding and Martha Palmer. 2005. Machine translation using probabilistic synchronous dependency insertion grammars. In *Proc. of* ACL 2005, Ann Arbor, Michigan, pages 541-548.

Jason Eisner. 2003. Learning non-isomorphic tree mappings for machine translation. In *Proc. of* ACL 2003, pages 205-208.

Michel Galley, Mark Hopkins, Kevin Knight and Daniel Marcu. 2004. What's in a translation rule? In *Proc. of* HLT-NAACL 2004, Boston, USA, pages 273-280.

Michel Galley, Jonathan Graehl, Kevin Knight, Daniel Marcu, Steve DeNeefe, Wei Wang and Ignacio Thayer. 2006. Scalable inferences and training of context-rich syntax translation models. In *Proc. of* COLING/ACL 2006, Sydney, Australia, pages 961-968.

Liang Huang and David Chiang. 2007. Forest rescoring: Faster decoding with integrated language models. In *Proc. of* ACL 2007, Prague, Czech Republic, pages 144-151.

Liang Huang, Kevin Knight and Aravind Joshi. 2006. Statistical syntax-directed translation with extended domain of locality. In *Proc. of* AMTA 2006, pages 66-73.

Philipp Koehn. 2004. Statistical Significance Tests for Machine Translation Evaluation. In *Proc. of* EMNLP 2004, Barcelona, Spain, pages 388-395.

Yang Liu, Qun Liu and Shouxun Lin. 2006. Tree-to-string alignment template for statistical machine translation. In *Proc. of* COLING/ACL 2006, Sydney, Australia, pages 609-616.

Yang Liu, Yajuan Lü and Qun Liu. 2009. Improving Tree-to-Tree Translation with Packed Forest. In *Proc. of* ACL 2009, pages 558-566.

Daniel Marcu, Wei Wang, Abdessamad Echihabi and Kevin Knight. 2006. SPMT: Statistical machine translation with syntactified target language phrases. In *Proc. of* EMNLP 2006, Sydney, Australia, pages 44-52.

Haitao Mi and Liang Huang. 2008. Forest-based Translation Rule Extraction. In *Proc. of* EMNLP 2008, pages 206-214.

Chris Quirk, Arul Menezes and Colin Cherry. 2005. Dependency treelet translation: Syntactically informed phrasal SMT. In *Proc. of* ACL 2005, pages 271-279.

Libin Shen, Jinxi Xu and Ralph Weischedel. 2008. A new string-to-dependency machine translation algorithm with a target dependency language model. In *Proc. of* ACL/HLT 2008, pages 577-585.

Ashish Venugopal, Andreas Zollmann, Noah A. Smith and Stephan Vogel. 2008. Wider Pipelines: K-best Alignments and Parses in MT Training. In *Proc. of* AMTA 2008, pages 192-201.

Wei Wang, Kevin Knight and Daniel Marcu. 2007. Binarizing Syntax Trees to Improve Syntax-Based Machine Translation Accuracy. In *Proc. of* EMNLP-CoNLL 2007, Prague, Czech Republic, pages 746-754.

Tong Xiao, Mu Li, Dongdong Zhang, Jingbo Zhu and Ming Zhou. 2009. Better Synchronous Binarization for Machine Translation. In *Proc. of* EMNLP 2009, Singapore, pages 362-370.

Kenji Yamada and Kevin Knight. 2001. A syntax-based statistical machine translation model. In *Proc. of* ACL 2001, pages 132-139.

Hao Zhang, Liang Huang, Daniel Gildea and Kevin Knight. 2006. Synchronous Binarization for Machine Translation. In *Proc. of* HLT-NAACL 2006, New York, USA, pages 256- 263.

Min Zhang, Hongfei Jiang, Aiti Aw, Haizhou Li, Chew Lim Tan and Sheng Li. 2008. A Tree Sequence Alignment-based Tree-to-Tree Translation Model. In *Proc. of* ACL/HLT 2008, pages 559-567.

Muhua Zhu, Jingbo Zhu and Tong Xiao. 2010. Heterogeneous Parsing via Collaborative Decoding. In *Proc. of* COLING 2010.

Boosting Relation Extraction with Limited Closed-World Knowledge

Feiyu Xu **Hans Uszkoreit** **Sebastian Krause** **Hong Li**
Language Technology Lab
German Research Center for Artificial Intelligence (DFKI GmbH)
{feiyu,uszkoreit,sebastian.krause,lihong}@dfki.de

Abstract

This paper presents a new approach to improving relation extraction based on minimally supervised learning. By adding some limited closed-world knowledge for confidence estimation of learned rules to the usual seed data, the precision of relation extraction can be considerably improved. Starting from an existing baseline system we demonstrate that utilizing limited closed world knowledge can effectively eliminate "dangerous" or plainly wrong rules during the bootstrapping process. The new method improves the reliability of the confidence estimation and the precision value of the extracted instances. Although recall suffers to a certain degree depending on the domain and the selected settings, the overall performance measured by F-score considerably improves. Finally we validate the adaptability of the best ranking method to a new domain and obtain promising results.

1 Introduction

Minimally supervised machine-learning approaches to learning rules or patterns for relation extraction (RE) in a bootstrapping framework are regarded as very effective methods for building information extraction (IE) systems and for adapting them to new domains (e. g., (Riloff, 1996), (Brin, 1998), (Agichtein and Gravano, 2000), (Yangarber, 2001), (Sudo et al., 2003), (Jones, 2005), (Greenwood and Stevenson, 2006), (Agichtein, 2006), (Xu et al., 2007), (Xu, 2007)). On the one hand, these approaches show very promising results by utilizing minimal domain knowledge as seeds. On the other hand, they are all confronted with the same problem, i.e., the acquisition of wrong rules because of missing knowledge for their validation during bootstrapping. Various approaches to confidence estimation of learned rules have been proposed as well as methods for identifying "so-called" negative rules for increasing the precision value (e.g., (Brin, 1998), (Agichtein and Gravano, 2000), (Agichtein, 2006), (Yangarber, 2003), (Pantel and Pennacchiotti, 2006), (Etzioni et al., 2005), (Xu et al., 2007) and (Uszkoreit et al., 2009)).

In this paper, we present a new approach to estimating or ranking the confidence value of learned rules by utilizing limited closed-world knowledge. As many predecessors, our ranking method is built on the "Duality Principle" (e. g., (Brin, 1998), (Yangarber, 2001) and (Agichtein, 2006)). We extend the validation method by an evaluation of extracted instances against some limited closed-world knowledge, while also allowing cases in which knowledge for informed decisions is not available. In comparison to previous approaches to negative examples or negative rules such as (Yangarber, 2003), (Etzioni et al., 2005) and (Uszkoreit et al., 2009), we implicitly generate many negative examples by utilizing the positive examples in the closed-world portion of our knowledge. Rules extracting wrong instances are lowered in rank.

In (Xu et al., 2007) and (Xu, 2007), we develop a generic framework for learning rules for relations of varying complexity, called ***DARE*** (Domain Adaptive Relation Extraction). Furthermore, there is a systematic error analysis of the base-

Coling 2008: Poster Volume, pages 1354–1362,
Beijing, August 2010

line system conducted in (Xu, 2007). We employ our system both as a baseline reference and as a platform for implementing and evaluating our new method.

Our first experiments conducted on the same data used in (Xu et al., 2007) demonstrate: 1) limited closed-world knowledge is very useful and effective for improving rule confidence estimation and precision of relation extraction; 2) integration of soft constraints boosts the confidence value of the good and relevant rules, but without strongly decreasing the recall value. In addition, we validate our method on a new corpus of newspaper texts about celebrities and obtain promising results.

The remainder of the paper is organized as follows: Section 2 explains the relevant related work. Sections 3 and 4 describe ***DARE*** and our extensions. Section 5 reports the experiments with two ranking strategies and their results. Section 6 gives a summary and discusses future work.

2 Related Work

In the existing minimally supervised rule learning systems for relation extraction based on bootstrapping, they already employ various approaches to confidence estimation of learned rules and different methods for identification of so-called negative rules. For estimation of confidence/relevance values of rules, most of the approaches follow the so-called "Duality Principle" as mentioned by Brin (1998) and Yangarber (2001), namely, the confidence value of learned rules is dependent on the confidence value of their origins, which can be documents or relation instances. For example, Riloff (1996), Yangarber (2001), Sudo et al. (2003) and Greenwood and Stevenson (2006) use domain relevance of documents in which patterns are discovered as well as the distribution frequency of these patterns in those relevant documents as an indication of good patterns. Their methods are aimed at detecting all patterns for a specific domain, but those patterns cannot be applied directly to a specific relation. In contrast, systems presented by Brin (1998), Agichtein and Gravano (2000), Agichtein (2006), Pantel and Pennacchiotti (2006) as well as our baseline system (Xu et al., 2007) are designed to learn rules for a specific relation. They start with some relation instances as their so-called "semantic seeds" and detect rules from texts matching with these instances. The new rules are applied to new texts for extracting new instances. These new instances in turn are utilized as new seeds. All these systems calculate their rule confidence based on the confidence values of the instances from which they stem. In addition to the confidence value of the seed instances, most of them also consider frequency information and include some heuristics for extra validation. For example, Agichtein (2006) intellectually defines certain constraints for evaluating the truth value of extracted instances. But it is not clear whether this strategy can be adapted to new domains and other relations. In (Xu et al., 2007) we make use of domain relevance values of terms occurring in rules. This method is not applicable to general relations.

Parallel to confidence estimation strategies, the learning of negative rules is useful for identifying wrong rules straightforwardly. Yangarber (2003) and Etzioni et al. (2005) utilize the so-called *Counter-Training* for detecting negative rules for a specific domain or a specific class by learning from multiple domains or classes at the same time. Examples of one certain domain or class are regarded as negative examples for the other ones. Bunescu and Mooney (2007) follow a classification-based approach to RE. They use positive and negative sentences of a target relation for a SVM classifier. Uszkoreit et al. (2009) exploit negative examples as seeds for learning further negative instances and negative rules. The disadvantage of the above four approaches is that the selected negative domains or classes or negative instances cover only a subset of the negative domains/classes/relations of the target domain/class/relation.

3 *DARE* Baseline System

Our baseline system ***DARE*** is a minimally supervised learning system for relation extraction, initialized by so-called "semantic seeds", i.e., examples of the target relations, labelled with their semantic roles. The system supports domain adaptation through a compositional rule representation and a bottom-up rule discovery strategy. In this

way, ***DARE*** can handle target relations of varying arity. The following example is a relation instance of the target relation from (Xu, 2007) concerning Nobel Prize awards: *<Mohamed ElBaradei, Nobel, Peace, 2005>*. The target relation contains four arguments: WINNER, PRIZE_NAME, PRIZE_AREA and YEAR. This example refers to an event mentioned in the sentence in example (1).

(1) *Mohamed ElBaradei, won the 2005 Nobel Prize for Peace on Friday because of ....*

Figure 1 is a simplified dependency tree of example (1). ***DARE*** utilizes a bottom-up rule discovery strategy to extract rules from such dependency trees. All sentences are processed with named entity recognition and dependency parsing.

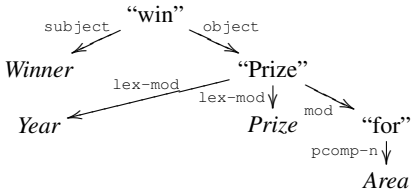


Figure 1: Dependency tree for example (1)

From the tree in Figure 1, ***DARE*** learns three rules. The first rule is dominated by the preposition "for", extracting the argument PRIZE_AREA (*Area*). The second rule is dominated by the noun "Prize", extracting the arguments YEAR (*Year*) and PRIZE_NAME (*Prize*), and calling the first rule for the argument PRIZE_AREA (*Area*). The rule "winner_prize_area_year_1" from Figure 2 extracts all four arguments from the verb phrase dominated by the verb "win" and calls the second rule to handle the arguments embedded in the linguistic argument "object".

Rule name :: winner_prize_area_year_1
Rule body ::

$$\begin{bmatrix} \text{head} & \begin{bmatrix} \text{pos} & \text{verb} \\ \text{mode} & \text{active} \\ \text{lex-form} & \text{"win"} \end{bmatrix} \\ \text{daughters} & < \left[\text{subject} \quad \left[\text{head} \quad \boxed{1}\textit{Winner}\right]\right], \\ & \left[\text{object} \quad \begin{bmatrix} \text{rule} & \text{year\_prize\_area\_1} :: \\ & < \boxed{4}\textit{Year}, \boxed{2}\textit{Prize}, \\ & \boxed{3}\textit{Area} > \end{bmatrix}\right] > \end{bmatrix}$$

Output :: $< \boxed{1}\textit{Winner}, \boxed{2}\textit{Prize}, \boxed{3}\textit{Area}, \boxed{4}\textit{Year} >$

Figure 2: ***DARE*** extraction rule.

We conduct a systematic error analysis based on our experiments with the Nobel Prize award data (Xu, 2007). The learned rules are divided into four groups: *good*, *useless*, *dangerous* and *bad*. The good rules are rules that only extract correct instances, while bad ones exclusively produce wrong instances. Useless rules are those that do not detect any new instances. Dangerous rules are dangerous because they extract both correct and wrong instances. Most good rules are rules with high specificity, namely, extracting all or most arguments of the target relation. The 14.7% extraction errors are from bad rules and dangerous rules. Other errors are caused by wrong reported content, negative modality, parsing and named entity recognition errors.

4 Our Approach: Boosting Relation Extraction

4.1 Closed-World Knowledge: Modeling and Construction

The error analysis of ***DARE*** confirms that the identification of bad rules or dangerous rules is important for the precision of an extraction system. Using closed-world knowledge with large numbers of implicit negative instances opens a possibility to detect such rules directly. In our work, closed-world knowledge for a target relation is the total set of positive relation instances for entire relations or for some selected subsets of individuals. For most real world applications, closed-world knowledge can only be obtained for relatively small subsets of individuals participating in the relevant relations. We store the closed-world knowledge in a relational database, which we dub "closed-world knowledge database" (abbr. ***cwDB***). Thus, a ***cwDB*** for a target relation should fill the following condition:

> A ***cwDB*** must contain all correct relation instances ($insts$) for an instantiation value (***argValue***) of a selected relation argument ***cwArg*** in the target relation.

Given $\mathbf{R}$ (the total set of relation instances of a target relation), a ***cwDB*** is defined as follows:

$$\textbf{\textit{cwDB}}=\{inst \in \mathbf{R} : \textbf{\textit{cwArg}}(inst) = \textbf{\textit{argValue}}\}.$$

An example of a ***cwDB*** is the set of all prize winners of a specific prize area such as *Peace*, where PRIZE_AREA is the selected ***cwArg*** and ***argValue*** is *Peace*. Note that the merger of two ***cwDB***s, for example with PRIZE_AREAS *Peace* and *Literature*, is again a ***cwDB*** (with two ***argValue***s in this case).

4.2 Modified Learning Algorithm

In Algorithm 1, we present the modification of the ***DARE*** algorithm (Xu, 2007). The basic idea of ***DARE*** is that it takes some initial seeds as input and learns relation extraction rules from sentences in the textual corpus matching the seeds. Given the learned rules, it extracts new instances from the texts. The modified algorithm adds the **validate** step to evaluate the new instances against the closed-world knowledge ***cwDB***. Based on the evaluation result, both new instances and learned rules are ranked with a confidence value.

INPUT: *initial seeds*
1 $i \leftarrow 0$ (iteration of bootstrapping)
2 $seeds \leftarrow initial\ seeds$
3 $all\ instances \leftarrow \{\}$
4 **while** $(seeds \neq \{\})$
5 $rules_i \leftarrow$ **getRules**$(seeds)$
6 $instances_i \leftarrow$ **getInstances**$(rules_i)$
7 $new\ instances_i \leftarrow instances_i - all\ instances$
8 **validate**$(new\ instances_i,$ ***cwDB***$)$
9 **rank**$(new\ instances_i)$
10 **rank**$(rules_i)$
11 $seeds \leftarrow new\ instances_i$
12 $all\ instances \leftarrow all\ instances + new\ instances_i$
13 $i \leftarrow i + 1$
OUTPUT: *all instances*

Algorithm 1: Extended ***DARE***

4.3 Validation against *cwDB*

Given a ***cwDB*** of a target relation and its ***argValue*** of its selected argument ***cwArg***, the validation of an extracted instance $(inst)$ against the ***cwDB*** is defined as follows.

$$\begin{aligned} inst\ correct &\Leftrightarrow inst \in \textbf{\textit{cwDB}} \\ inst\ wrong &\Leftrightarrow inst \notin \textbf{\textit{cwDB}} \wedge \\ &\quad \textbf{\textit{cwArg}}(inst) = \textbf{\textit{argValue}} \\ inst\ unknown &\Leftrightarrow (inst \notin \textbf{\textit{cwDB}} \wedge \\ &\quad \textbf{\textit{cwArg}}(inst) \neq \textbf{\textit{argValue}}) \\ &\vee (inst \notin \textbf{\textit{cwDB}} \wedge \\ &\quad \textbf{\textit{cwArg}}(inst) \text{ is } unspecified) \end{aligned} \quad (1)$$

4.4 Rule Confidence Ranking with *cwDB*

We develop two rule-ranking strategies for confidence estimation, in order to investigate the best way of integrating the closed-world knowledge: (a) **exclusive ranking**: This ranking strategy excludes every rule which extracts wrong instances after their validation against the closed-world knowledge; (b) **soft ranking**: This ranking strategy is built on top of the duality principle and takes specificity and the depth of learning into account.

Exclusive Ranking The exclusive ranking method is a very naive ranking method which estimates the confidence value of a learned rule (e.g., $rule$) depending on the truth value of its extracted instances (**getInstances**$(rule)$) against a ***cwDB***. Any rule with one *wrong* extraction is regarded as a bad rule in this method. This method works effectively in a special scenario where the total list of the instances of the target relation is available as the ***cwDB***.

$$\textbf{confidence}(rule) = \begin{cases} 1 & \text{if } \textbf{getInstances}(rule) \subseteq \textbf{\textit{cwDB}}, \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

Soft Ranking The soft ranking method works in the spirit of the "Duality Principle", the confidence value of rules is dependent on the truth value of their extracted instances and on the seed instances from which they stem. The confidence value of the extracted instances is estimated based on their validation against the ***cwDB*** or the confidence value of their ancestor seed instances from which their extraction rules stem. Furthermore, the *specificity* of the instances (percentage of the filled arguments) and the *learning depth* (iteration step of bootstrapping) are parameters too. The definition of instance scoring, namely, **score**$(inst)$, is given as follows:

$$\textbf{score}(inst) = \begin{cases} \gamma > 0 & \text{if } \textbf{validate}(inst, \textbf{\textit{cwDB}}) = correct, \\ 0 & \text{if } \textbf{validate}(inst, \textbf{\textit{cwDB}}) = wrong, \\ UN_{inst} & \text{if } \textbf{validate}(inst, \textbf{\textit{cwDB}}) = unknown. \end{cases} \quad (3)$$

As defined above, if a new instance is confirmed as correct by the ***cwDB***, it will obtain a positive value. In our experiment, we set γ=10 in order to boost the precision. In the case of *unknown* about its truth value, the confidence value of a new instance $(inst)$ is dependent on the confidence values of the seed instances (ancestor seeds) from which its mother rules (R_{inst}) stem. Below, the scoring of the *unknown* case, namely, UN_{inst}, is defined, where R_{inst} are rules that extract the new instance $inst$, while I_{rule} are instances from which a $rule$ in R_{inst} is learned and α is the specificity value of $inst$ while β is utilized to express the noisy potential of each further iteration during bootstrapping.

$$UN_{inst} = \frac{\sum_{rule \in R_{inst}} \left(\frac{\sum_{j \in I_{rule}} \textbf{score}(j)}{|I_{rule}|} \times \beta^{i_{rule}} \right)}{|R_{inst}|} \times \alpha$$

where
$R_{inst} = \textbf{getMotherRulesOf}(inst)$,
$I_{rule} = \textbf{getMotherInstancesOf}(rule)$,
$\alpha = specificity$,
$\beta = 0.8$,
$i_{rule} = i$-th iteration where $rule$ occurs (4)

Given the scoring of instance $inst$, the confidence estimation of a rule is the average score of all $inst$s extracted by this rule:

$$\textbf{confidence}(rule) = \frac{\sum_{inst \in \mathbb{I}} \textbf{score}(inst)}{|\mathbb{I}|}$$

where $\mathbb{I} = \textbf{getInstances}(rule)$ (5)

5 Experiments

5.1 Corpora and Closed-World Knowledge

We conduct our experiments with two different domains. We start with the Nobel Prize award domain reported in (Xu, 2007) and apply our method to the same corpus, a collection from various online newspapers. The target relation is the one with the four arguments as mentioned in Section 3. In this way, we can compare our results with those reported in (Xu, 2007). Furthermore, all Nobel Prize winners can be found from `http://nobelprize.org`, so it is easy to construct a ***cwDB*** for Nobel Prize winners. We take the PRIZE_AREA as our selected argument for closing sub-relations and construct various ***cwDB***s with the instantiation of this argument (e.g., all winners of Nobel *Peace* Prize). The second domain is about celebrities. Our text corpus is collected from tabloid newspaper texts, containing 6850 articles from the years 2001 and 2002. The target relation is the marriage relationship between two persons. We construct a ***cwDB*** of 289 persons in which we have listed all their (ex-)spouses as well as the time span of the marriage relation.

Table 1 summarizes the size of the corpus data of the two domains.

Domain	Space	#Doc.
Nobel Prize	18,4 MB	3328
Celebrity Marr.	16,6 MB	6850

Table 1: Corpus data.

5.2 Nobel Prize Domain

We apply the extended ***DARE*** system to the Nobel Prize corpus at first and conduct two rule ranking strategies with different sizes of the ***cwDB***. We conduct all our experiments with the seed *<Guenter Grass, Nobel, Literature, 1999>*. The ***DARE***-Baseline performance is shown in Table 2.

	Precision	Absolute Recall
Baseline	77.98%	89.01%

Table 2: ***DARE***-Baseline Performance

Exclusive Ranking

Given the complete list of Nobel Laureates, we can apply the exclusive ranking strategy to this domain. Our ***cwDB*** is the total list of Nobel Prize winners. The wrong instances will not be used as seed for the next iteration. Rules that extracted at least one wrong instance are marked as *bad*, the other rules as *good*. We utilize only the good rules for relation extraction.

Prec.	Rel. Recall	Rel. F-Measure
100.00%	82.88%	90.64%

Table 3: Performance of Exclusive Ranking in Nobel Prize award domain.

In comparison to the ***DARE*** baseline system, given the same seed setup, this experiment results in a precision boost from 77.98% to 100% (see Table 3). This is not surprising since the ***cwDB*** covers all relation instances for the target relation. Nevertheless, this experiment shows that the closed-world knowledge approach is effective to exclude bad rules. However, the recall decreases and is only 82.88% of the one of the baseline system. As we explain above, not all rules extracting wrong instances are bad rules because wrong extractions can also be caused by other error sources such as named entity recognition. Therefore, even good rules can be excluded because of other error sources. The exclusive ranking strategy is useful for application scenarios where people want to learn rules for achieving 100% precision performance and do not expect high recall. It is especially effective when a big ***cwDB*** is available.

Soft Ranking

This ranking strategy does not exclude any rules and assigns a score to each rule based on

the definition in Section 4.4. Rules which extract correct instances, more specific relation instances and stem from high-scored seed instances obtain a better value than others. In our approach, the *specificity* is dependent on the number of the arguments in the extracted instances. For this domain, the most specific instances contain all four arguments. In the following, we conduct two experiments with two different sizes of the ***cwDB***: 1) with the total list of winners (*complete* ***cwDB***) and 2) with only winners in one PRIZE_AREA (*limited* ***cwDB***).

1) Complete closed-world database Figure 3 displays the correlation between the score of rules and their extraction precision performance. Each point stands for a set of rules with the same score and extraction precision. In this setup, the higher the score, the higher the precision. Given the scored rules, Figure 4 depicts precision, recall and F-Measure for different score thresholds. For a given threshold j we take all $rule$s with **score**$(rule) \geq j$ and use the instances they extract. The recall value here is the relative recall w. r. t. to the ***DARE*** baseline performance: i. e. the number of correct extracted instances divided by the number of correct instances extracted by the ***DARE*** baseline system. The F-Measure value is calculated by using the relative recall values, we therefore refer to it as the *relative* F-Measure. If the system takes all rules with score ≥ 7, the system achieves the best relative F-Measure.

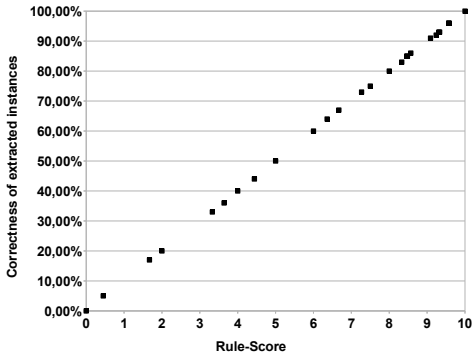


Figure 3: Rule scores vs. precisions with the complete closed-world database.

2) Limited closed-world database This experiment investigates the system performance in cases in which only a limited ***cwDB*** is available. This is the typical situation for most real world RE applications. Therefore, this experiment is much more important than the previous one. We construct a smaller database containing only *Peace* Nobel Prize winners, which is about 1/8 of the previous complete ***cwDB***.

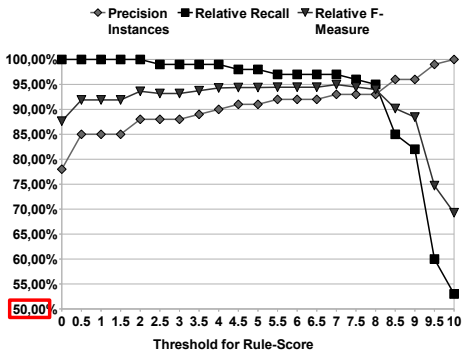


Figure 4: Performance with the complete closed-world database.

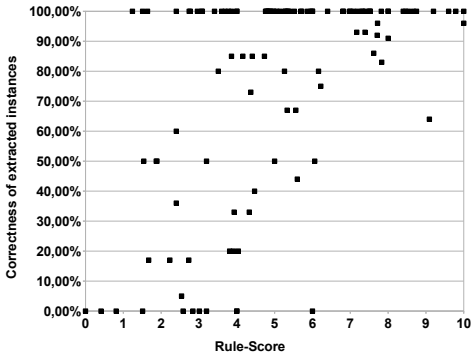


Figure 5: Rule score vs. precision with the limited closed-world database

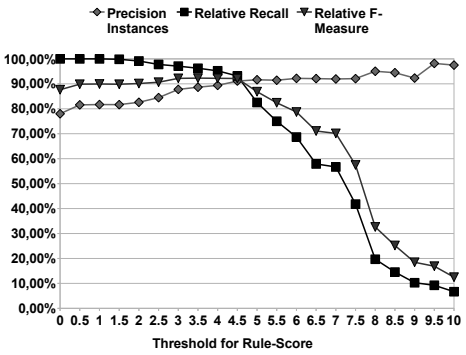


Figure 6: Performance with the limited closed-world database

Figure 5 shows the correlation between the score of the rules and their extraction precision. Although the development curve here is not as smooth as depicted in Figure 3, the higher scored rules have better precision values than most of the lower scored rules. However, we can observe that some very good rules are scored low, located in

Thresh.	Good	Dangerous	Bad
Baseline	58.94%	26.49%	14.57%
1	64.96%	29.20%	5.84%
2	66.67%	27.91%	5.43%
3	69.23%	26.50%	4.27%
4	**73.27%**	**23.76%**	**2.97%**
5	76.00%	22.67%	1.33%
6	77.59%	20.69%	1.72%
7	77.50%	22.50%	0.00%
8	87.50%	12.50%	0.00%
9	85.71%	14.29%	0.00%
10	90.00%	10.00%	0.00%

Table 4: Quality analysis of rules with the limited closed-world database

the left upper corner. The reason is that many of their extracted instances are *unknown*, even if their extracted instances are mostly correct.

As shown in Figure 6, even with the limited ***cwDB***, the precision values are comparable with the complete ***cwDB*** (see Figure 4). However, the recall value drops much earlier than with the complete ***cwDB***. With a threshold of score 4, the system achieves the best modified F-Measure 92,21% with an improvement of precision of about 11 percentage points compared to the ***DARE*** baseline system (89.39% vs. 77.98%). These results show that even with a limited ***cwDB*** this ranking system can help to improve the precision without loosing too much recall.

We take a closer look on the useful (actively extracting) rules and their extraction performance, using the same rule classification as (Xu, 2007). As shown in Table 4, more than one fourth of the extraction rules created by the baseline system are dangerous ones and almost 15% are plainly wrong. Applying the rule scoring with the limited ***cwDB*** increases the fraction of good rules to almost three quarters and nearly eliminates all bad rules at threshold 4. By choosing higher thresholds, surviving good rules raises to 90%. The total remaining set of rules then only consists of rules that at least partially extract correct instances.

5.3 Celebrity Domain

As presented above, the soft ranking method delivers very promising result. In order to validate this ranking method, we choose an additional domain and decide to learn marriage relations among celebrities, where the target relation consists of the following arguments: [NAME_OF_SPOUSE, NAME_OF_SPOUSE, YEAR].

The value of the marriage year is valid when the year is within the marriage time interval. The motivation of selecting this target relation is the large number of possible relations between two persons leading to dangerous or even bad rules. For example, the rule in Figure 7 is a very dangerous rule because "meeting" events of two married celebrities are often reported. A good confidence estimation method is very useful for boosting the good rules like the one in Figure 8. From our text corpus we extract 37.000 sentences that mention at least two persons. The ***cwDB*** consists of sample relation instances, in which one NAME_OF_SPOUSE is instantiated, i. e. we manually construct a database which contains all (ex-) spouses of 289 celebrities.

```
head([SPOUSE<ne_person>]),
mod({head(("meet", VB)),
    subj({head([SPOUSE<ne_person>])})})
```

Figure 7: A dangerous extraction rule example

```
head(("marry", VB)),
aux({head(("be", VB))}),
dep({head([SPOUSE<ne_person>]),
       dep({head([DATE<point>])})}),
nsubj({head([SPOUSE<ne_person>])})
```

Figure 8: Example of a positive rule

Since a gold standard of mentions for this corpus is not available, we manually validate 100 random samples from each threshold group. This evaluation gives us an opportunity to estimate the effect of a ***cwDB*** in this domain. Table 5 presents the performance of the rules with different thresholds. The precision value of the baseline system is very low. Threshold 3 slightly improves the precision of the ***DARE*** baseline without damaging recall too much. Step 4 excludes dangerous rules such as the one in Figure 7 which drastically boosts the precision. Unfortunately, the exclusion of such general rules leads to the loss of many correct relation instances too, therefore, the immense drop of recall from threshold 3 to 4 as well as from threshold 4 to 5. Positive extraction rules such as Figure 8 are quite highly scored. Because of the large number of rules and instances, we start the quality analysis of rules with score 3. As the table indicates, the use of the rule scoring in this domain clearly improves the quality of the created extraction rules. The error analysis shows that the major error resource for this domain is wrong coreference resolution or identity resolution. For ex-

Thresh.	# Instances	Prec.	Rel. Rec.	Rel. F-Meas.	# Rules	Good	Dangerous	Bad
Baseline	25183	9.00%	100.00%	16.51%	12258	———	———	———
1	19806	7.00%	61.17%	12.56%	562	———	———	———
2	14542	9.00%	57.75%	15.57%	159	———	———	———
3	11259	15.00%	74.51%	24.97%	121	19.83%	33.88%	46.28%
4	**788**	**65.00%**	**22.60%**	**33.54%**	**72**	**25.00%**	**27.78%**	**47.22%**
5	195	67.00%	5.76%	10.62%	29	37.93%	17.24%	44.83%
6	115	84.00%	4.26%	8.11%	11	45.45%	27.27%	27.27%
7	55	89.09%	2.16%	4.22%	6	50.00%	33.33%	16.67%
8	9	77.78%	0.31%	0.62%	4	75.00%	0.00%	25.00%
9	5	60.00%	0.13%	0.26%	3	66.67%	0.00%	33.33%
10	5	60.00%	0.13%	0.26%	3	66.67%	0.00%	33.33%

Table 5: Soft ranking for the celebrity marriage domain with a limited ***cwDB***.

ample, the inability to distinguish *Prince Charles* (former husband of British princess Diana) from *Charles Spencer* (her brother) is the reason that ***DARE*** crosses the border between the marriage and the sibling relation. In comparison to the Nobel Prize award event, the marriage relation between persons is often used as additional information to a person which is involved in a reported event. Therefore, anaphoric references occur more often in their mentionings, as the example relation in (3).

(3) *"My kids, I really don't like them to watch that much television," said Cruise, 40, who adopted Isabella and Connor while he was married to second wife Nicole Kidman.*

6 Summary

We propose a new way in which prior knowledge about domains can be efficiently used as additional criteria for confidence estimation of learned new rules or new instances in a minimally supervised machine learning framework. By introducing rule scoring on the basis of available domain knowledge (the cwDB), rules can be evaluated during the bootstrapping process with respect to their extraction precision. The results are rather promising. The rule score threshold is an easy way for users of an extraction system to adjust the precision-recall-trade-off to their own needs. The rule estimation method is also general enough to extend to integration of common sense knowledge. Although the relation instances in the closed-world knowledge database can also be used as seed in the beginning, the core idea of our research work is to develop a general confidence estimation strategy for discovered new information. As discussed in (Xu, 2007) and (Uszkoreit et al., 2009), the size of seed is not always relevant for the learning and extraction performance, in particular if the data corpus exhibits the small world property. Using all instances in the cwDB as seed, our experiments with the baseline system yield worse precision performance than the modified ***DARE*** algorithm with only one seed instance.

This approach is quite general and easily adaptable to many domains; the only prerequisite is the existence of a database with relation instances from the target domain with a fulfilled closed-world property on some relational argument. A database of this kind should be easily obtainable for many domains, e. g. by exploiting structured and semi-structured information sources in the Internet, such as *YAGO* (Suchanek et al. (2007)) and *DBpedia* (Bizer et al. (2009)). Furthermore, in some areas, such as Business Intelligence, there is nearly complete knowledge already present for past years, while the task is to extract information only from recent news articles. Constructing closed-worlds out of the present knowledge to improve the learning of new information is therefore a straightforward approach. Even the manual collection of suitable data might be a reasonable choice since appropriate closed worlds could be rather small if ***cwDB***is chosen properly.

Acknowledgments

The work presented here has been partially supported through the prject KomParse by the ProFIT program of the Federal State of Berlin which in turn is co-funded by the EFRE program of the European Union. It is additionally supported through a grant to the project TAKE, funded by the German Ministry for Education and Research (BMBF, FKZ: 01IW08003).

References

Agichtein, Eugene and Luis Gravano. 2000. Snowball: extracting relations from large plain-text collections. In *DL '00: Proceedings of the fifth ACM conference on Digital libraries*, pages 85–94, New York, NY, USA. ACM.

Agichtein, Eugene. 2006. Confidence estimation methods for partially supervised information extraction. In *Proceedings of the Sixth SIAM International Conference on Data Mining*, Bethesda, MD, USA, April. SIAM.

Bizer, Christian, Jens Lehmann, Georgi Kobilarov, Sören Auer, Christian Becker, Richard Cyganiak, and Sebastian Hellmann. 2009. DBpedia - a crystallization point for the web of data. *Journal of Web Semantics*, 7(3):154–165.

Brin, Sergey. 1998. Extracting patterns and relations from the world wide web. In *WebDB Workshop at 6th International Conference on Extending Database Technology, EDBT'98*.

Bunescu, Razvan C. and Raymond J. Mooney. 2007. Learning to extract relations from the web using minimal supervision. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*.

Etzioni, Oren, Michael Cafarella, Doug Downey, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S. Weld, and Alexander Yates. 2005. Unsupervised named-entity extraction from the web: An experimental study. *Artificial Intelligence*, 165(1):91 – 134.

Greenwood, Mark A. and Mark Stevenson. 2006. Improving semi-supervised acquisition of relation extraction patterns. In *Proceedings of the Workshop on Information Extraction Beyond The Document*, pages 29–35, Sydney, Australia, July. Association for Computational Linguistics.

Jones, R. 2005. *Learning to Extract Entities from Labeled and Unlabeled Text*. Ph.D. thesis, University of Utah.

Pantel, Patrick and Marco Pennacchiotti. 2006. Espresso: Leveraging generic patterns for automatically harvesting semantic relations. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, Sydney, Australia, July. The Association for Computer Linguistics.

Riloff, Ellen. 1996. Automatically generating extraction patterns from untagged text. In *Proceedings of Thirteenth National Conference on Artificial Intelligence (AAAI-96)*, pages 1044–1049. The AAAI Press/MIT Press.

Suchanek, Fabian M., Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: A Core of Semantic Knowledge. In *16th international World Wide Web conference (WWW 2007)*, New York, NY, USA. ACM Press.

Sudo, K., S. Sekine, and R. Grishman. 2003. An improved extraction pattern representation model for automatic IE pattern acquisition. *Proceedings of ACL 2003*, pages 224–231.

Uszkoreit, Hans, Feiyu Xu, and Hong Li. 2009. Analysis and improvement of minimally supervised machine learning for relation extraction. In *14th International Conference on Applications of Natural Language to Information Systems*. Springer.

Xu, Feiyu, Hans Uszkoreit, and Hong Li. 2007. A seed-driven bottom-up machine learning framework for extracting relations of various complexity. In *Proceedings of ACL 2007, 45th Annual Meeting of the Association for Computational Linguistics*, Prague, Czech Republic, June.

Xu, Feiyu. 2007. *Bootstrapping Relation Extraction from Semantic Seeds*. Phd-thesis, Saarland University.

Yangarber, Roman. 2001. *Scenarion Customization for Information Extraction*. Dissertation, Department of Computer Science, Graduate School of Arts and Science, New York University, New York, USA.

Yangarber, Roman. 2003. Counter-training in discovery of semantic patterns. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 343–350, Sapporo Convention Center, Sapporo, Japan, July.

Applying Syntactic, Semantic and Discourse Constraints in Chinese Temporal Annotation

Nianwen Xue
Brandeis University
xuen@brandeis.edu

Yuping Zhou
Brandeis University
yzhou@brandeis.edu

Abstract

We describe a Chinese temporal annotation experiment that produced a sizable data set for the TempEval-2 evaluation campaign. We show that while we have achieved high inter-annotator agreement for simpler tasks such as identification of events and time expressions, temporal relation annotation proves to be much more challenging. We show that in order to improve the inter-annotator agreement it is important to strategically select the annotation targets, and the selection of annotation targets should be subject to syntactic, semantic and discourse constraints.

1 Introduction

Event-based temporal inference is a fundamental natural language technology that attempts to determine the temporal location of an event as well as the temporal ordering between events. It supports a wide range of natural language applications such as Information Extraction, Question Answering and Text Summarization. For some genres of text (such as news), a temporal ordering of events can be the most informative summarization of a document (Mani and Wilson, 2000; Filatova and Hovy, 2001). Temporal inference is especially important for multi-document summarization where events extracted from multiple documents need to be put in a chronological order (Lin and Hovy, 2001; Barzilay et al., 2002) to make logical sense. Event-based temporal inference is also necessary for Question Answering (Harabagiu and Bejan, 2005; Harabagiu and Bejan, 2006). For example, to answer "When was Beijing Olympics held?", events extracted from natural language text have to be associated with a temporal location, whereas to answer "how many terrorists have been caught since 9/11?", temporal ordering of multiple events is the prerequisite. Event-based temporal inference has also been studied extensively in the context of Information Extraction, which typically involves extracting unstructured information from natural language sources and putting them into a structured database for querying or other forms of information access. For event extraction, this means extracting the event participants as well as its temporal location. Generally, an event has to occur in a specific time and space, and the temporal location of an event provides the necessary context for accurately understanding that event.

Being able to infer the temporal location of an event in Chinese text has many additional applications. Besides Information Extraction, Question Answering and Text Summarization, knowing the temporal location of an event is also highly valuable to Machine Translation. To translate a language like Chinese into a language like English in which tense is grammatically marked with inflectional morphemes, an MT system will have to infer the necessary temporal information to determine the correct tense for verbs. Statistical MT systems, the currently dominant research paradigm, typically do not address this issue directly or even indirectly.

As machine learning approaches are gaining dominance in computational linguistics and producing state-of-the-art results in many areas, they have in turn fueled the demand for large quantities of human-annotated data of various types

Coling 2008: Poster Volume, pages 1363–1372,
Beijing, August 2010

that machine learning algorithms can be trained on and evaluated against. In the temporal inference domain, this has led to the creation of TimeBank (Pustejovsky et al., 2003), which is annotated based on the TimeML language (Pustejovsky et al., 2005). TimeML is becoming an ISO standard for annotating events and time expressions (ISO/TC 37/SC 4/WG 2, 2007). A version of the TimeBank has been provided as a shared public resource for TempEval-2007, the first temporal evaluation campaign aimed at automatically identifying temporal relations between events and time expressions as well the temporal ordering between events.

In this paper, we report work for a Chinese temporal annotation project as part of the 2010 multilingual temporal evaluation campaign (TempEval-2)[1]. Besides Chinese, TempEval-2 also includes English, French, Italian, Korean and Spanish. Our temporal annotation project is set up within the confines of BAT[2], a database-driven multilingual temporal annotation tool that is also used to support other TempEval-2 languages. The TempEval-2 evaluation framework takes a divide-and-conquer approach to temporal annotation. With the eventual goal being the annotation of temporal relations between events and between events and time expressions, the TempEval-2 annotation consists of a series of event and temporal annotation subtasks. The idea is that each of these subtasks will be easier to annotate than the larger task as a whole and is less demanding on the annotators. The hope is that this will lead to more consistent annotation that will be easier to learn for automatic systems as well.

The rest of the paper will be organized as follows. In Section 2, we briefly describe the seven layers of annotation. In Section 3, we describe our annotation procedure. In Section 4, we address a major issue that arises from our annotation effort, which is the question of how to select annotation targets. Our experience, some positive and some negative, shows that temporal annotation can be carried out much more smoothly and with higher quality when the right annotation targets are presented to the annotators. This is especially true during the annotation of temporal relations between events and between events and time expressions, which are more complex than simpler annotation tasks such as identifying the events and time expressions. Section 5 concludes our paper.

[1] http://www.timeml.org/tempeval2/

[2] http://www.timeml.org/site/bat

2 Layers of annotation

2.1 Events and time expressions

The ultimate goal for a temporal annotation project is to determine the temporal relationship between events, and between events and time expressions. In order to achieve that objective, events and time expressions must be first identified. Specifically, this means marking up text spans in a document that can be used to represent the events and time expressions. Events in particular are abstract objects and a full description of an event would include its participants and temporal and spatial location. The TempEval annotation framework simplifies this by just marking a verb or a noun that best represents an event. The verb or noun can be considered as an "event anchor" that represents the most important aspect of the event. This is illustrated in (1), where the verbs 参加 ("attend"), 举行 ("hold") and the noun 仪式 ("ceremony") are marked as event anchors.

(1) 国务院 副总理 邹家华
State Council Vice Premier Zou Jiahua
参加 了 今天 举行 的 投产
attend ASP today hold DE commissioning
剪彩 仪式 。
ribbon-cutting ceremony .

"Vice Premier Zou Jiahua of the State Council attended today's commissioning ribbon-cutting ceremony".

Once the text spans of event anchors are annotated, these events are then annotated with a set of attributes. The TempEval annotation framework allows variations across languages in the number of attributes one can define as well as the values for these attributes. For example, in the English annotation, one of the event attributes is grammatical *tense* which can be read off the morphological inflections of a verb. Chinese verbs, on the other hand, are not inflected for tense. Instead, in the

Chinese annotation, we have a more fully developed *aspect* attribute that has eight possible values: *Actual*, *Experiential*, *Complementive*, *Delimitative*, *Progressive*, *Durative*, *Inceptive*, and *Continuative*, largely based on the theoretical work of Xiao and McEnery (2004).

The most important attribute for both English and Chinese, however, is the *Class* attribute. The values for this attribute include *Reporting*, *Aspectual*, *Perception*, *I-Action*, *I-State*, *State*, and *Occurrence*. The different values of the *Class* attribute effectively constitute a classification of events, and they are defined in the TimeML specification language (Pustejovsky et al., 2005).

The other building block in the TempEval annotation framework is time expressions. Like events, time expressions are marked with both text spans and a set of attributes. The annotation of time expressions is relatively straightforward, and we follow the TimeML standards in our annotation study. In TimeML, time expressions are formally called TIMEX3s, and they have two obligatory attributes: *Type* and *Value*. The value of *Type* is one of *time*, *date*, *duration* or *set*. The *Value* attribute is essentially a normalized time value based on the TIDES standard for annotating time expressions (Ferro et al., 2004). The normalization allows easy comparison of time expression. For example, there are three time expressions in (2), 一九九二年("1992"), 一九九六年 ("1996") and 今年 ("this year"). Note that even though 一九九二年 至 一九九六年 (" 1992 to 1996") forms one duration, it is annotated as two time expressions. All three time expressions in the sentence are dates, and their normalized values are *1992*, *1996*, and *1997* respectively. To determine the normalized value for 今年 ("this year"), we need to know the document creation time, and fortunately this information is available in the metadata for the Chinese Treebank documents.

(2) 一九九二年 至 一九九六年 上海
1992 to 1996 Shanghai
国内 生产 总值 年均
GDP per year on average
增长 百分之十四点二 ， 今年 的
grow 14.2% , this year DE
增长 速度 也 将 达到 百分之十三
growth speed also will reach 13%
以上 。
above

"From 1992 to 1996, Shanghai's GDP on average grows at14.2% per year. This year the (GDP) growth will also reach above 13%."

2.2 Temporal relations

Once the events and time expressions are in place, we are in a position to annotate various temporal relations that are defined over them. (Since events and time expressions are entities that temporal relation is defined upon, we will subsume them under the cover term "temporal entity" when convenient.) The ultimate goal of temporal annotation is to identify all temporal relations in text. This goal cannot be achieved by manually annotating temporal relation of all temporal entities for three reasons. First, it is infeasible, given the number of temporal entities in a typical document. Second, it is unnecessary due to the transitive property of certain types of temporal relation. For example, if *e1*, *e2* and *e3* are all events, and if *e1* is before *e2*, and *e2* is before *e3*, there is no need to also annotate the relation between *e1* and *e3*. Third, the result of annotating all temporal entity pairs does not reflect the natural temporal relations that exist in text. Verhagen et al. (2009) found that a major contributor to high inter-annotator disagreement was hard-to-classify cases that annotators were instructed not to avoid. If a temporal relation is not made clear in text, then it should not be present in annotation.

Since it is infeasible, unnecessary and even detrimental to manually annotate all possible relations between temporal entities, the question then becomes one of selecting which temporal relations to annotate. The TempEval-2 evaluation starts by annotating the following temporal relations, which it considers to be a priority:

1. between an event and a time expression
2. between an event and the document creation time
3. between a subordinating event and its corresponding subordinated event

4. between a main event and its immediately preceding main event

The TempEval-2 annotation uses six values for all temporal relations, and they are *Before*, *Before-or-Overlap*, *Overlap*, *Overlap-or-After*, *After* and *Vague*. The *Vague* value is only used as the last resort when the annotator really cannot determine the temporal relationship between a pair of temporal entities. In the meantime, the TempEval-2 also allows variations from language to language regarding specific annotation strategies for each subtask. For Chinese temporal annotation, most of the decisions we have to make revolve around one central question, and that is which temporal entity pair to annotate.

2.2.1 Relation between events and time expressions

The annotation of the relationship between events and time expressions involves i) determining which event is related to which time expression, and ii) what is the nature of this relationship. In (3), for example, there are three events and three time expressions that enter into the temporal relation annotation. If the annotator is required to annotate all possible event/time combinations, there will be nine possible pairs. There are at least three possible strategies to go about selecting event/time pairs to annotate. The first strategy is to annotate all possible pairs. This seems to add unnecessary burden to the annotator because if we know that *e1* overlaps *t1*, we can infer the temporal relationship between *e1* and *t3* by virtue of the fact that *t1* occurs before *t3*. The second strategy is to allow the annotator to freely choose which event/time pair to annotate based on whether there is a clear temporal relation between them. This eliminates the possibility that the annotator is forced to annotate hard-to-classify and inconsequential relations, but leaving this decision to the annotator entirely might lead to low inter-annotator agreement where annotators choose to annotate different event/time pairs.

(3) 国际 货币 基金 组织 [**t1** ２１日]
International Monetary Fund 21st
在 此间 [**e1** 发表] 一 份 临时
at here publish one CL preliminary
评估 报告 ， 再次 [**e2** 调低] 了
assessment report , again lower AS
它 对 [**t2** 今] [**t3** 明] 两 年
its regarding this next two year
全球 经济 增长 速度 的 [**e3**
global economic growth speed DE
预测] 。
forecast .

"The International Monetary Fund on 21 published a preliminary assessment report, again lowering its forecast of the global economic growth for this year and next year."

In our annotation, we adopt a third strategy. Instead of simply asking which event bears a temporal relation to which temporal expression in the same sentence, we ask annotators to judge *which event(s) a given temporal expression is **intended** to modify*. In essence, this amounts to asking the annotator to first make a syntactic decision about which events fall within the scope of a time expression. In (3), all three events *e1*, *e2* and *e3* fall within the scope of *t1*, and none of them are in the scope of *t2* and *t3*. This approach reduces the number of fuzzy temporal relations that annotators might disagree on due to preference for thoroughness vs. accuracy.

2.2.2 Temporal relation between subordinating event and subordinated event

The two tasks in the TempEval framework that deal with event pairs are to annotate temporal relation between the subordinating event and the subordinated event, as well as the relation in main event pairs. The division of labor between them is quite clear: the former deals with intra-sentential temporal relations whereas the latter handles inter-sentential relations. It is not immediately clear, however, how each of the two types of relations should be defined.

Unlike in the event/time annotation where syntactic notions are invoked in selecting event/time pairs to annotate, our definitions of subordinating and subordinated events are primarily based on semantic criteria. The subordinating event is roughly the predicate while the subordinated event is one of its arguments, provided that both the

predicate and the argument are anchors of events. For example, in (4), there are two subordinating and subordinated event pairs. *e2* is a subordinated event of *e1*, and *e4* is a subordinated event of *e3*.

(4) 广东 [**e1** 举行] [**e2** 研讨会] [**e3** 介绍] [**e4** 税改] 及 加工 贸易 台帐 制度
Guangdong hold symposium introduce tax reform and processing trade accounting regulation

"Guangdong held a symposium introducing the tax reform and the accounting regulations on processing trade."

An alternative to using the notion of predicate-argument structure in determining the subordinating/subordinated events is to resort to syntactic relations such as the verb and its object. The net result would be the same for Example (4). However, the same argument that motivates the annotation of the predicate-argument structures in the Propbank (Palmer et al., 2005) and the Chinese Propbank (Xue and Palmer, 2009) also applies to temporal annotation. That is, the predicate-argument structure and temporal relations tend to hold constant in spite of the syntactic alternations and variations. For example, the temporal relation between the noun 研讨会 ("symposium") event and the verb 举行 ("hold") event remains the same in (5) in spite of the change in the syntactic relation between them. If only event pairs in a verb-object relation are annotated, the temporal relation between *e2* and *e1* in (5) would be lost.

(5) [**e2** 研讨会] 在 广东 [**e1** 举行]
symposium PREP Guangdong hold

"The symposium was held in Guangdong."

2.2.3 Temporal relations between main events

The purpose of annotating the temporal relation between main events is to capture the temporal ordering of events scattered in different sentences that constitute the main chain of events covered in the article. Annotation of the temporal relation between main events is further divided into two steps. In the first step, main events are first identified among all events in a sentence, and then the temporal relation between the main events in adjacent pairs of sentences is annotated. As a first approximation, we define "main event" as follows: a main event is the event expressed by the main verb of the top-most level clause of a sentence. The underlying assumption is that good writing would place words representing important events in prominent positions of a sentence and the first choice of a prominent position in a sentence is probably the main verb. An additional stipulation is that in case of a co-ordinated construction involving two or more main verbs at the top-most level, the event represented by the first is the main event of the sentence. This is to ensure that each sentence has only one main event. As we shall see in Section 3, this seemingly simple turns out to be surprisingly difficult, as reflected in the low inter-annotator agreement.

2.2.4 Temporal relation between events and the document creation time

In this layer, all the events identified in a document are annotated according to their temporal relation to the document creation time. This task is particularly challenging and intellectually interesting for Chinese. As an isolating language (Li and Thompson, 1981), Chinese has a small word to morpheme ratio. That is, the majority of its words consist of single morphemes. As a result, it lacks the inflectional morphology that grammatically marks tense. Tense directly encodes the temporal location of an event in natural language text and the lack of observable grammatical tense makes it that much harder to determine the temporal location of an event in Chinese text. This is not to say, however, that Chinese speakers do not attempt to convey the temporal location of events when they speak or write, or that they cannot interpret the temporal location when they read Chinese text, or even that they have a different way of representing the temporal location of events. In fact, there is evidence that the temporal location is represented in Chinese in exactly the same way as it is represented in English and most world languages: in relation to the moment of speech. One piece of evidence to support this claim is that Chinese temporal expressions like 今天 ("today"), 明天 ("tomorrow") and 昨天 ("yesterday") all assume a

temporal deixis that is the moment of speech in relation to which all temporal locations are defined. Annotating the temporal relation between events and document creation time would then directly capture the temporal location of events.

3 Annotation procedure and annotation consistency

The data set consists of 60 files taken from the Chinese Treebank (Xue et al., 2005). The source of these files is Xinhua newswire. It goes through a two-phase double blind and adjudication process. The first phase involves three annotators, with each file annotated by two annotators; the second phase involves two judges, with each double annotated document assigned to a single judge for disagreement resolution. The inter-annotator agreement between the two annotators (A and B) as the agreement between each annotator and the judge (J) are presented in Table 1. The agreement is measured in terms of F1-score[3], which is a weighted average between precision and recall. The F1-score is calculated as follows:

$$F = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \quad (1)$$

The agreement statistics in Table 1 clearly show that event and time expression annotations are easier but temporal relations are harder as reflected in the lower inter-annotator agreement scores. This is somewhat expected because relations involve two temporal entities while we are only dealing with one temporal entity with event and time expression annotations. The figures also show the seemingly simple task of main event annotation (which only involves picking one event per sentence as the main event) has a surprisingly low inter-annotator agreement score. One reason might be that in a less grammaticalized language like Chinese, it is not always clear which verb is the main verb when the syntactic tree information is not displayed in the annotation interface. Another reason is that annotators sometimes disregard the syntax-based rule when it runs too much afoul to their intuition, a point that we will come back to and discuss in greater detail in Section 4.

[3]For a subset of the tasks, the total number of annotated instances for the two annotators is the same. This subset includes identification of main events, the temporal relation between the main events in two adjacent sentences, and the temporal relation between an event and the document creation time.

Layer	f(A, B)	f(A, J)	f(B, J)
event-extent	0.90	0.93	0.94
timex-extent	0.86	0.88	0.93
main-events	0.74	0.90	0.82
tlinks-main-events	0.65	0.70	0.75
tlinks-dct-events	0.77	0.86	0.90
tlinks-e-t	0.75	0.88	0.83
tlinks-sub-e	0.53	0.74	0.70

Table 1: Inter-annotator agreement for the subtasks: *event-extent*, the textual extent of an event anchor; *timex-extent*, the textual span of a time expression; *tlinks-main-event*, the temporal relation between the main events; *tlinks-dct-events*, the temporal link between an event and the document creation time; *tlinks-e-t*, the temporal relation between an event and a time expression; *tlinks-sub-e*, the temporal relation between a subordinating event and a subordinated event.

It is worth noting that the annotation of the temporal relation between an event and a time expression, and between a subordinating event and a subordinated event involves two decisions. The annotator needs to first decide which pairs of temporal entities to annotate, and then decide what temporal relation should be assigned to each temporal entity pair. To take a closer look at which of these two decisions creates more of a problem for the annotator, we computed the agreement figures for these two steps respectively. In Table 2, Column 3 presents the figure for just identifying which pair to annotate, and Column 4 is the agreement for just assigning the temporal relation, assuming the same pair of temporal entities are found by both annotators.

Layer	all	identification f	relation
tlinks-e-t	0.75	0.86	0.89
tlinks-sub-e	0.53	0.60	0.87

Table 2: Detailed agreement for event-time and subordinating-subordinated events

From Table 2, it is clear that for both tasks,

there is lower agreement between the annotators in deciding which pair to annotate. Once the two annotators agree on which pair to annotate, determining the temporal relation is relatively easier, as reflected in higher agreement.

4 Detailed discussion

As described in Section 2, when annotating the temporal relation between an event and a time expression, the annotators are instructed to annotate an event-time pair if the event is falling within the syntactic scope of the time expression. When annotating the relation between subordinating and subordinated events, the annotators are instructed to select event pairs based on the semantic notion of predicate-argument structure. This assumes a certain level of linguistic sophistication on the part of the annotators. From the lower agreement score in identifying event-time pairs (Table 2), it is clear that our annotators, who are not trained linguists, lack in this type of specialized knowledge. They are better at making the more intuitive judgment regarding the temporal relation between two temporal entities. One solution is obviously to find better trained linguists to perform these tasks, but it may not always be feasible. Since our data is taking from the Chinese Treebank and has already been annotated with syntactic structures and predicate-argument structures (from the Chinese Propbank annotation (Xue and Palmer, 2009)), an alternative is to extract the event-time or event-event pairs using the syntactic and predicate-argument structures as constraints.[4]

The annotation of main events and their relations presents a different challenge. Our first approximation is to select main events based on syntactic considerations. A main event is equated with the matrix verb in a sentence. In many cases this turns out to be unintuitive. Two of the recurring counter-intuitive cases involve directly quoted speech and coordination structures.

Directly quoted speech In Chinese newswire text, it is often the case that the source of information is explicitly cited in the form of direct quotations. (6) is such an example:

(6) 宋健 说：“如今 ，中国
Song-Jian say , “ nowadays , China
已 能 生产 上万 门
already can produce tens-of-thousands CL
数字 电话 程控交换机。”
digital telephone PBX

“Song Jian said, ‘nowadays, China is capable of producing tens of thousands of digital telephone PBX.’ ”

While the event represented by the underlined verb 说 (“say”) may very well be important in some natural language processing applications (for example, sometimes the source of the target information is crucial), it is not normally part of the intended information being covered by a news article. And it does not make much sense to annotate its temporal relation to adjacent main events that are on a par with what was said, not the saying event itself. The point would be even clearer when such a case is contrasted with a case in which a similar semantic relation is formulated in a different syntactic structure, as shown in (7):

(7) **据** 官方 权威 人士
according to official authority source
透露 ，今年 中国 政府
divulge , this-year China government
确定 的 经济 增长率 为
determine DE economic growth rate be
百分之八。
8%

“According to some official sources in position of authority, the economic growth rate determined by the Chinese government is 8%.”

Because of the presence of the preposition 据 (“according to”), the underlined reporting verb 透露 (“divulge”), similar to 说 (“say”) in (6) with respect to its semantic relation to the following material, would not be annotated as representing the main event of the sentence. The difference in the annotation of the main event between (7) and (6) seems to be an undesirable artifact of the purely syntax-based annotation rule for identifying main events.

[4] See a similar approach in Bethard et al. (2007).

Co-ordination structure Co-ordination by no means is a rare occurrence in the data, and often times, all events within a co-ordination structure, taken together, represent the main event of the sentence. For example, in (8), both events represented by the underlined verbs seem to be equally significant and should be included in the same chain of events. Given the prevalence of co-ordination between verbs, the stipulation that only the first one counts significantly undermines the coverage of the task and goes against the annotator's intuitions.

(8) 今年 9月 ，多 家 外国
This year September , many CL foreign
石油公司 与 哈 国家 石油
oil company with Kazakstan national oil
公司 签署 了 一揽子 “世纪
company sign LE a series of “century
合同” ，这些 合同 将 在今后
contract” , these contract will in future
４０年 内 产生 ７０００亿
40 years within generate 700-billion
美元 的 巨额 利润 。
dollar DE enormous profit

“In September of this year, many foreign oil companies signed a series of ‘century contract’ with Kazakstan National Oil Company. These contracts will generate an enormous profit of 700-billion dollars.”

The issue in the annotation of the temporal relation between main events seem to be more in the selection of main event pairs than in the determination of the nature of their relationship. Our current rule states that any two main events in consecutive sentences form a pair for annotation. This task suffers a low level of inter-annotator agreement partly because many main events identified by syntactic criteria are not actually main events in our intended sense. Often times, two consecutive main events come from different levels of the discourse structure or different chains of events, which puts annotators in a hard-to-classify situation.

To achieve high inter-annotator consistency when annotating the temporal relation between events from different sentences, we believe the selection of event pairs has to be informed by the discourse structure of the document. This only makes sense given that the annotation of temporal relation between events and time expressions within one sentence is informed by the syntactic structure, and the temporal relation between subordination and subordinating events benefits from an understanding of the predicate-argument structure.

The specific type of discourse structure we have in mind is the kind represented in the Penn Discourse Treebank (Miltsakaki et al., 2004). The Penn Discourse Treebank-style of annotation can inform temporal relation annotation in at least two ways. First, the Penn Discourse Treebank annotates the discourse relation between two adjacent sentences. The discourse relation holds between two abstract objects such as events or propositions. If a discourse relation holds between two events, the temporal relation between those two events might also be what we are interested in for temporal annotation. The implicit assumption is that the discourse structure of a document represents the important temporal relations within that document as well. (9) is an example taken from the Penn Discourse Treebank. The discourse relation, characterized by the discourse connective “in particular”, holds between the events anchored by“dropped” and “fell”. The temporal relation between these events also happens to be what we would be interested in if we are to annotate the main events between two adjacent sentences. Notice that in (9), material that is irrelevant to the discourse relation is taken out of the two arguments of this discourse relation, which are marked in italics and bold face respectively.

(9) *Meanwhile, the average yield on taxable funds dropped nearly a tenth of a percentage point, the largest drop since midsummer.* implicit = in particular **The average seven-day compound yield**, which assumes that dividends are reinvested and that current rates continue for a year, **fell to 8.47%, its lowest since late last year, from 8.55% the week before, according to Donoghue’ s.**

The Penn Discourse Treebank also marks attributions when annotating discourse relations. In

(10), for example, "he says" will be marked as a case of attribution and the "say" verb would be marked as the main event of the sentence if syntactic criteria are followed. Having attributions identified would directly help with the temporal annotation of examples like (6), where the main event is embedded in direct quoted speech.

(10) When *Mr. Green won a $240,000 verdict in a land condemnation case against the State in June 1983*, [he says] **Judge O' Kicki unexpectedly awarded him an additional $100,000.**

As of now, the data we use for our temporal annotation experiment have not yet been annotated with discourse structures. In order to make our temporal annotation sensitive to the discourse structure, we either have to annotate the discourse structure in a separate pass, or to incorporate the key elements of the discourse structure when developing guidelines for temporal annotation.

5 Conclusion

We described a Chinese temporal annotation experiment that produced a sizable data set for the TempEval-2 annotation campaign. We show that while we have achieved high inter-annotator agreement for simpler tasks such as identification of events and time expressions, temporal relation annotation proves to be much more challenging. We show that in order to improve annotation consistency it is important to strategically select the annotation targets, and this selection process should be subject to syntactic, semantic and discourse constraints.

Acknowledgements

This work is supported by the National Science Foundation via Grant No. 0855184 entitled "Building a community resource for temporal inference in Chinese". All views expressed in this paper are those of the authors and do not necessarily represent the view of the National Science Foundation.

References

Regina Barzilay, Noemie Elhadad, and Kathleen McKeown. 2002. Inferring strategies for sentence ordering in multidocument news summarization. *Journal of Artificial Intelligence Research*, 17:35–55.

Steven Bethard, James H. Martin, and Sara Klingenstein. 2007. Finding Temporal Structure in Text: Machine Learning of Syntactic Temporal Relations. *International Journal of Semantic Computing*, 11(4).

Lisa Ferro, Laurie Gerber, Inderjeet Mani, Beth Sundheim, and George Wilson. 2004. TIDES 2003 Standard for the Annotation of Temporal Expressions.

Elena Filatova and Eduard Hovy. 2001. Assigning Time-Stamps to Event Clauses. In *Proceedings of the Workshop on Temporal and Spatial Information Processing*, Toulouse.

Sanda Harabagiu and Cosmin Adrian Bejan. 2005. Question Answering Based on Temporal Inference. In *Proceedings of the AAAI-2005 Workshop on Inference for Textual Question Answering*, Pittsburgh, Pennsylvania.

Sanda Harabagiu and Cosmin Adrian Bejan. 2006. An Answer Bank for Temporal Inference. In *Proceedings of LREC 2006*, Genoa, Italy.

ISO/TC 37/SC 4/WG 2. 2007. Language Resource Management – Semantic Annotation Framework (SemAF) – Part 1: Time and events.

Charles Li and Sandra Thompson. 1981. *Mandarin Chinese: A Functional Reference Grammar*. Berkeley, Los Angeles, London: University of California Press.

Chin-Yew Lin and Eduard Hovy. 2001. Neats: A multidocument summarizer. In *Proceedings of the Document Understanding Workshop*.

Inderjeet Mani and George Wilson. 2000. Robust temporal processing of news. In *Proceedings of the ACL'2000*, Hong Kong, China.

Eleni Miltsakaki, Rashmi Prasad, Aravind Joshi, and Bonnie Webber. 2004. The Penn Discourse TreeBank. In *Proceedings of the Language Resources and Evaluation Conference*, Lisbon, Portugal.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The Proposition Bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106.

James Pustejovsky, Patrick Hanks, Roser Sauri, Andrew See, David Day, Lisa Ferro, Robert Gaizauskas, Marcia Lazo, Andrea Setzer, and Beth

Sundheim. 2003. The TimeBank Corpus. *Corpus Linguistics*, pages 647–656.

James Pustejovsky, Bob Ingria, Roser Sauri, Jose Castano, Jessica Littman, Rob Gaizauskas, Andrea Setzer, G. Katz, and I. Mani. 2005. The specification language TimeML. In I. Mani, J. Pustejovsky, and R. Gaizauskas, editors, *The Language of Time: a Reader*. Oxford University Press.

Marc Verhagen, Robert Gaizauskas, Frank Schilder, Mark Hepple, Jessica Moszkowicz, and James Pustejovsky. 2009. The TempEval Challenge: Identifying Temporal Relation in Text. *Language Resources and Evaluation*, 43(1):161–179.

Richard Xiao and Tony McEnery. 2004. *Aspect in Mandarin Chinese: A Corpus-based Study*. Amsterdam: John Benjamins.

Nianwen Xue and Martha Palmer. 2009. Adding semantic roles to the Chinese Treebank. *Natural Language Engineering*, 15(1):143–172.

Nianwen Xue, Fei Xia, Fu dong Chiou, and Martha Palmer. 2005. The Penn Chinese TreeBank: Phrase Structure Annotation of a Large Corpus. *Natural Language Engineering*, 11(2):207–238.

Syntax-Driven Machine Translation as a Model of ESL Revision

Huichao Xue and **Rebecca Hwa**
Department of Computer Science
University of Pittsburgh
{hux10,hwa}@cs.pitt.edu

Abstract

In this work, we model the writing revision process of English as a Second Language (ESL) students with syntax-driven machine translation methods. We compare two approaches: tree-to-string transformations (Yamada and Knight, 2001) and tree-to-tree transformations (Smith and Eisner, 2006). Results suggest that while the tree-to-tree model provides a greater coverage, the tree-to-string approach offers a more plausible model of ESL learners' revision writing process.

1 Introduction

When learning a second language, students make mistakes along the way. While some mistakes are idiosyncratic and individual, many are systematic and common to people who share the same primary language. There has been extensive research on grammar error detection. Most previous efforts focus on identifying specific types of problems commonly encountered by English as a Second Language (ESL) learners. Some examples include the proper usage of determiners (Yi et al., 2008; Gamon et al., 2008), prepositions (Chodorow et al., 2007; Gamon et al., 2008; Hermet et al., 2008), and mass versus count nouns (Nagata et al., 2006). However, previous work suggests that *grammar error correction* is considerably more challenging than detection (Han et al., 2010). Furthermore, an ESL learner's writing may contain multiple interacting errors that are difficult to detect and correct in isolation.

A promising research direction is to tackle automatic grammar error correction as a machine translation (MT) problem. The disfluent sentences produced by an ESL learner can be seen as the input source language, and the corrected revision is the result of the translation. Brockett et al. (2006) showed that phrase-based statistical MT can help to correct mistakes made on mass nouns. To our knowledge, phrase-based MT techniques have not been applied for rewriting entire sentences. One major challenge is the lack of appropriate training data such as a sizable parallel corpus. Another concern is that phrase-based MT may not be similar enough to the problem of correcting ESL learner mistakes. While MT rewrites an entire source sentence into the target language, not every word written by an ESL learner needs to be modified.

Another alternative that may afford a more general model of ESL error corrections is to consider syntax-driven MT approaches. We argue that syntax-based approaches can overcome the expected challenges in applying MT to this domain. First, it can be less data-intensive because the mapping is formed at a structural level rather than the surface word level. While it does require a robust parser, a syntax-driven MT model may not need to train on a very large parallel corpus. Second, syntactic transformations provide an intuitive description of how second language learners revise their writings: they are transforming structures in their primary language to those in the new language.

In this paper, we conduct a first inquiry into the applicability of syntax-driven MT methods to automatic grammar error correction. In particular, we investigate whether a syntax-driven model can capture ESL students' process of writing revisions. We compare two approaches: a tree-to-string mapping proposed by Yamada & Knight (2001) and a tree-to-tree mapping using the Quasi-Synchronous

Coling 2008: Poster Volume, pages 1373–1381,
Beijing, August 2010

Grammar (QG) formalism (Smith and Eisner, 2006). We train both models on a parallel corpus consisting of multiple drafts of essays by ESL students. The approaches are evaluated on how well they model the revision pairs in an unseen test corpus. Experimental results suggest that 1) the QG model has more flexibility and is able to describe more types of transformations; but 2) the YK model is better at capturing the incremental improvements in the ESL learners' revision writing process.

2 Problem Description

This paper explores the research question: can ESL learners' process of revising their writings be described by a computational model? A successful model of the revision process has several potential applications. In addition to automatic grammar error detection and correction, it may also be useful as an automatic metric in an intelligent tutoring system to evaluate how well the students are learning to make their own revisions.

Revising an ESL student's writing bears some resemblance to translating. The student's first draft is likely to contain disfluent expressions that arose from translation divergences between English and the student's primary language. In the revised draft, the divergences should be resolved so that the text becomes fluent English. We investigate to what extent are formalisms used for machine translation applicable to model writing revision. We hypothesize that ESL students typically modify sentences to make them sound more fluent rather than to drastically change the meanings of what they are trying to convey. Thus, our work focuses on syntax-driven MT models.

One challenge of applying MT methods to model grammar error correction is the lack of appropriate training data. The equivalence to the bilingual parallel corpus used for developing MT systems would be a corpus in which each student sentence is paired with a fluent version re-written by an instructor. Unlike bilingual text, however, there is not much data of this type in practice because there are typically too many students for the teachers to provide detailed manual inspection and correction at a large scale. More commonly, students are asked to revise their previously written essays as they learn more about the English language. Here is an example of a student sentence from a first-draft essay:

> The problem here is that they come to the US like illegal.

In a later draft, it has been revised into:

> The problem here is that they come to the US illegally.

Although the students are not able to create "gold standard revisions" due to their still imperfect understanding of English, a corpus that pairs the students' earlier and later drafts still offers us an opportunity to model how ESL speakers make mistakes.

More formally, the corpus $\mathcal{C}$ consists of a set of sentence pairs (O, R), where O represents the student's original draft and R represents the revised draft. Note that while R is assumed to be an improvement upon O, its quality may fall short of the gold standard revision, G. To train the syntax-driven MT models, we optimize the joint probability of observing the sentence pair, $\Pr(O, R)$, through some form of mapping between their parse trees, τ_O and τ_R.

An added wrinkle to our problem is that it might not always be possible to assign a sensible syntactic structure to an ungrammatical sentence. It is well-known that an English parser trained on the Penn Treebank is bad at handling disfluent sentences (Charniak et al., 2003; Foster et al., 2008). In our domain, since O (and perhaps also R) might be disfluent, an important question that a translation model must address is: how should the mapping between the trees τ_O and τ_R be handled?

3 Syntax-Driven Models for Essay Revisions

There is extensive literature on syntax-driven approaches to MT (cf. a recent survey by

Lopez (2008)); we focus on two particular formalisms that reflects different perspectives on the role of syntax. Our goal is to assess which formalism is a better fit with the domain of essay revision modeling, in which the data largely consist of imperfect sentences that may not support a plausible syntactic interpretation.

3.1 Tree-to-String Model

The Yamada & Knight (henceforth, YK) tree-to-string model is an instance of noisy channel translation systems, which assumes that the observed source sentence is the result of transformation performed on the parse tree of the intended target sentence due to a noisy communication channel. Given a parallel corpus, and a parser for the the target side, the parameters of this model can be estimated using EM(Expectation Maximization). The trained model's job is to recover the target sentence (and tree) through decoding.

While the noisy channel generation story may sound somewhat counter-intuitive for translation, it gives a plausible account of ESL learner's writing process. The student really wants to convey a fluent English sentence with a well-formed structure, but due to an imperfect understanding of the language, writes down an ungrammatical sentence, O, as a first draft. The student serves as the noisy channel. The YK model describes this as a stochastic process that performs three operations on τ_G, the parse of the intended sentence, G:

1. Each node in τ_G may have its children **reordered** with some probability.

2. Each node in τ_G may have a child node **inserted** to its left or right with some probability.

3. Each leaf node (i.e., surface word) in τ_G is **replaced** by some (possibly empty) string according to its lexical translation distribution.

The resulting sentence, O, is the concatenation of the leaf nodes of the transformed τ_G.

Common mistakes made by ESL learners, such as misuses of determiners and prepositions, word choice errors, and incorrect constituency orderings, can be modeled by a combination of the **insert**, **replace**, and **reorder** operators. The YK model allows us to perform transformations on a higher syntactic level. Another potential benefit is that the model does not attempt to assign syntactic interpretations over the source sentences (i.e., the less fluent original draft).

3.2 Tree-to-Tree Model

The Quasi-Synchronous Grammar formalism (Smith and Eisner, 2006) is a generative model that aims to produce the most likely target tree for a given source tree. It differs from the more strict synchronous grammar formalisms (Wu, 1995; Melamed et al., 2004) because it does not try to perform simultaneous parsing on parallel grammars; instead, the model learns an augmented target-language grammar whose rules make "soft alignments" with a given source tree.

QG has been applied to some NLP tasks other than MT, including answer selection for question-answering (Wang et al., 2007), paraphrase identification (Das and Smith, 2009), and parser adaptation and projection (Smith and Eisner, 2009). In this work we use an instantiation of QG that largely follows the model described by Smith and Eisner (2006). The model is trained on a parallel corpus in which both the first-draft and revised sentences have been parsed. Using EM to estimate its parameters, it learns an augmented target PCFG grammar[1] whose production rules form associations with the given source trees.

Consider the scenario in Figure 1. Given a source tree τ_O, the trained model generates a target tree by expanding the production rules in the augmented target PCFG. To apply a

[1]For expository purposes, we illustrate the model using a PCFG production rule. In the experiment, a statistical English *dependency* parser (Klein and Manning, 2004) was used.

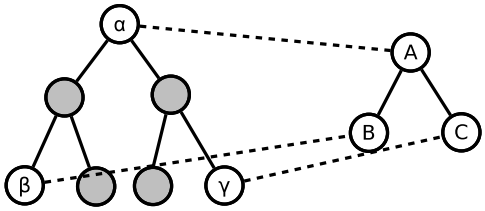


Figure 1: An example of QG's soft alignments between a given source tree and a possible target rule expansion.

target-side production rule such as

$$A \rightarrow BC,$$

the model considers which source tree nodes might be associated with each target-side nonterminals:

$$(\alpha, A) \rightarrow (\beta, B)(\gamma, C)$$

where α, β, γ are nodes in τ_O. Thus, assuming that the target symbol A has already been aligned to source node α from an earlier derivation step, the likelihood of expanding (α, A) with the above production rule depends on three factors:

1. the likelihood of the **monolingual target rule**, $\Pr(A \rightarrow BC)$
2. the likelihood of **alignments** between B and β as well as C and γ.
3. the likelihood that the source nodes form some expected **configuration** (i.e., between α and β as well as between α and γ). In this work, we distinguish between two configuration types: *parent-child* and *other*. This restriction doesn't reduce the explanatory power of the resulting QG model, though it may not be as fine-tuned as some models in (Smith and Eisner, 2006).

Under QG, the ESL students' first drafts are seen as text in a different language that has its own syntactic constructions. QG explains the grammar rules that govern the revised text in terms of how different components map to structures in the original draft. It makes explicit the representation of divergences between the students' original mental model and the expected structure.

3.3 Method of Model Comparison

Cross entropy can be used as a metric that measures the distance between the learned probabilistic model and the real data. It can be interpreted as measuring the amount of information that is needed in addition to the model to accurately recover the observed data. In language modeling, cross entropy is widely used in showing a given model's prediction power.

To determine how well the two syntax-driven MT models capture the ESL student revision generation process, we measure the cross entropy of each trained model on an unseen test corpus. This quantity measures how surprised a model is about relating an initial sentence, O, to its corresponding revision, R. Specifically, the cross entropy for some model M on a test corpus $\mathcal{C}$ of original and revised sentence pairs (O, R) is:

$$-\frac{1}{|\mathcal{C}|} \sum_{(O,R)\in\mathcal{C}} \log \Pr_M(O, R)$$

Because neither model computes the joint probability of the sentence pair, we need to make additional computations so that the models can be compared directly.

The YK model computes the likelihood of the first-draft sentence O given an assumed gold parse τ_R of the revised sentence: $\Pr_{YK}(O \mid \tau_R)$. To determine the joint probability, we would need to compute:

$$\begin{aligned} \Pr_{YK}(O, R) &= \sum_{\tau_R \in \Lambda_R} \Pr_{YK}(O, \tau_R) \\ &= \sum_{\tau_R \in \Lambda_R} \Pr_{YK}(O \mid \tau_R) \Pr(\tau_R) \end{aligned}$$

where Λ_R represents the set of possible parse trees for sentence R. Practically, performing tree-to-string mapping over the entire set of trees in Λ_R is computationally intractable. Moreover, the motivation behind the YK

	mean	stdev
percentage of $O = R$	54.11%	N/A
O's length	12.95	4.87
R's length	12.74	4.20
edit distance	1.88	3.58

Table 1: This table summarizes some statistics of the dataset.

model is to trust the given τ_R. Thus, we made a Viterbi approximation:

$$\begin{aligned}\Pr_{YK}(O, R) &= \sum_{\tau_R \in \Lambda_R} \Pr_{YK}(O \mid \tau_R) \Pr(\tau_R) \\ &\approx \Pr_{YK}(O \mid \hat{\tau_R}) \Pr(\hat{\tau_R})\end{aligned}$$

where $\Pr(\hat{\tau_R})$ is the probability of the single best parse tree according to a standard English parser.

Similarly, to compute the joint sentence pair probability under the QG model would require summing over both sets of trees because the model computes $\Pr_{QG}(\tau_R \mid \tau_O)$. Here, we make the Viterbi approximation on both trees.

$$\begin{aligned}\Pr_{QG}(O, R) &= \sum_{\tau_R \in \Lambda_R} \sum_{\tau_O \in \Lambda_O} \Pr_{QG}(\tau_O, \tau_R) \\ &= \sum_{\tau_R \in \Lambda_R} \sum_{\tau_O \in \Lambda_O} \Pr_{QG}(\tau_R \mid \tau_O) \Pr(\tau_O) \\ &\approx \Pr_{QG}(\hat{\tau_R} \mid \hat{\tau_O}) \Pr(\hat{\tau_O})\end{aligned}$$

where $\hat{\tau_O}$ and $\hat{\tau_R}$ are the best parses for sentences O and R according to the underlying English dependency parser, respectively.

4 Experiments

4.1 Data

Our experiments are conducted using a collection of ESL students' writing samples[2]. These are short essays of approximately 30 sentences on topics such as "a letter to your parents." The students are asked to revise their essays at least once. From the dataset, we extracted 358 article pairs.

[2]The dataset is made available by the Pittsburgh Science of Learning Center English as a Second Language Course Committee, supported by NSF Award SBE-0354420.

Typically, the changes between the drafts are incremental. Approximately half of the sentences are not changed at all. These sentences are considered useful because this phenomenon strongly implies that the original version is good enough to the best of the author's knowledge. In a few rare cases, students may write an entirely different essay. We applied TF-IDF to automatically align the sentences between essay drafts. Any sentence pair with a cosine similarity score of less than 0.3 is filtered. This resulted in a parallel corpus of 7580 sentence pairs.

Because both models are computational intensive, we further restricted our experiments to sentence pairs for which the revised sentence has no more than 20 words. This reduces our corpus to 4666 sentence pairs. Some statistics of the sentence pairs are shown in Table 1.

4.2 Experimental Setup

We randomly split the resulting dataset into a training corpus of 4566 sentence pairs and a test corpus of 100 pairs.

The training of both models involve an EM algorithm. We initialize the model parameters with some reasonable values. Then, in each iteration of training, the model parameters are re-estimated by collecting the expected counts across possible alignments between each sentence pair in the training corpus. In out experiments, both models had two iterations of training. Below, we highlight our initialization procedure for each model.

In the YK model, the initial **reordering** probability distribution is set to prefer no change 50% of the time. The remaining probability mass is distributed evenly over all of the other permutations. For the **insertion** operation, for each node, the YK model first chooses whether to insert a new string to its left, to its right, or not at all, conditioned on the node's label and its parent's label. These distributions are initialized uniformly ($\frac{1}{3}$). If a new string should be inserted, the model then makes that choice with some probability. The insertion probability of each string in the

dictionary is assigned evenly with $\frac{1}{N}$, where N is the number of words in the dictionary. Finally, the **replace** probability distribution is initialized uniformly with the same value ($\frac{1}{N+1}$) across all words in the dictionary, including the empty string.

For the QG model, the initial parameters are determined as follows: For the **monolingual target parsing model parameters**, we first parse the target side of the corpus (i.e., the revised sentences) with the Stanford parser; we then use the maximum likelihood estimates based on these parse trees to initialize the parameters of the target parser, Dependency Model with Valence (DMV). We uniformly initialized the **configuration parameters**; the *parent-child* configuration and *other* configuration each has 0.5 probability. For the **alignment parameters**, we ran the GIZA++ implementation of the IBM word alignment model (Och and Ney, 2003) on the sentence pairs, and used the resulting translation table as our initial estimation. There may be better initialization setups, but the difference between those setups will become small after a few rounds of EM.

Once trained, the two models compute the joint probability of every sentence pair in the test corpus as described in Section 3.3.

4.3 Experiment I

To evaluate how well the models describe the ESL revision domain, we want to see which model is less "surprised" by the test data. We expected that the better model should be able to transform more sentence pair in the test corpus; we also expect that the better model should have a lower cross entropy with respect to the test corpus.

Applying both YK and QG to the test corpus, we find that neither model is able to transform all the test sentence pairs. Of the two, QG had the better coverage; it successfully modeled 59 pairs out of 100 (we denote this subset as D_{QG}). In contrast, YK modeled 36 pairs (this subset is denoted as D_{YK}).

To determine whether there were some characteristics of the data that made one model better at performing transformations for certain sentence pairs, we compare corpus statistics for different test subsets. Based on the results summarized in Table 2, we make a few observations.

First, the sentence pairs that neither model could transform seem, as a whole, more difficult. Their average lengths are longer, and the average per word Levenshtein edit distance is bigger. The differences between *Neither* and the other subsets are statistically significant with 90% confidence. For the length difference, we applied standard two-sample t-test. For the edit distance difference, we applied hypothesis testing with the null-hypothesis that "longer sentence pairs are as likely to be covered by our model as shorter ones."

Second, both models sometimes have trouble with sentence pairs that require no change. This may be due to out-of-vocabulary words in the test corpus. A more aggressive smoothing strategy could improve the coverage for both models.

Third, comparing the subset of sentence pairs that only QG could transform ($D_{QG} - D_{YK}$) against the subset of sentences that both models could transform ($D_{QG} \cap D_{YK}$), the former has slightly higher average edit distance and length, but the difference is not statistically significant. Although QG could transform more sentence pairs, the cross entropy of $D_{QG} - D_{YK}$ is higher than QG's estimate for the $D_{QG} \cap D_{YK}$ subset. QG's soft alignment property allows it to model more complex transformations with greater flexibility.

Finally, while the YK model has a more limited coverage, it models those transformations with a greater certainty. For the common subset of sentence pairs that both models could transform, YK has a much lower cross entropy than QG. Table 3 further breaks down the common subset. It is not surprising that both models have low entropy for identical sentence pairs. For modeling sentence pairs that contain revisions, YK is more efficient than QG.

	Neither	$D_{QG} \cap D_{YK}$	$D_{QG} - D_{YK}$	$D_{YK} - D_{QG}$
number of instances	38	33	26	3
average edit distance	2.42	1.88	2.08	1
% of identical pairs	53%	48%	58%	67%
average O length	14.63	12.36	12.58	6.67
average R length	13.87	12.06	12.62	6.67
QG cross entropy	N/A	127.95	138.9	N/A
YK cross entropy	N/A	78.76	N/A	43.84

Table 2: A comparison of the two models based on their coverage of the test corpus. Some relevant statistics on the sentence subsets are also summarized in the table.

	YK	QG
overall entropy	78.76	127.95
on identical pairs	52.59	85.40
on non-identical pairs	103.99	168.00

Table 3: A further comparison of the two models on $D_{QG} \cap D_{YK}$, the sentence pairs in the test corpus that both could transform.

4.4 Experiment II

The results of the previous experiment raises the possibility that QG might have a greater coverage because it is too flexible. However, an appropriate model should not only assign large probability mass to positive examples, but it should also have a low chance of choosing negative examples. In this next experiment, we construct a "negative" test corpus to see how it affects the models.

To construct a negative scenario, we still use the same test corpus as before, but we *reverse* the sentence pairs. That is, we use the revised sentences as "originals" and the original sentences as "revisions." We would expect a good model to have a raised cross entropy values along with a drop in coverage on the new dataset because the "revisions" should be more disfluent than the "original" sentences.

Table 4 summarizes the results. We observe that the number of instances that can be transformed has dropped for both models: from 59 to 49 pairs for QG, and from 36 to 20 pairs for YK; also, the proportion of identical instances in each set has raised. This means that both models are more surprised by the reverse test corpus, suggesting that both models have, to some extent, succeeded in modeling the ESL revision domain. However, QG still allows for many more transformations. Moreover, 16 out of the 49 instances are non-identical pairs. In contrast, YK modeled only 1 non-identical sentence pair. The results from these two experiments suggest that YK is more suited for modeling the ESL revision domain than QG. One possible explanation is that QG allows more flexibility and would require more training. Another possible explanation is that because YK assumes well-formed syntax structure for only the target side, the philosophy behind its design is a better fit with the ESL revision problem.

5 Related Work

There are many research directions in the field of ESL error correction. A great deal of the work focuses on the lexical or shallow syntactic level. Typically, local features such as word identity and POS tagging information are combined to deal with some specific kind of error. Among them, (Burstein et al., 2004) developed a tool called Critique that detects collocation errors and word choice errors. Nagata et al. (2006) uses a rule-based approach in distinguishing mass and count nouns. Knight and Chander (1994) and Han et al. (2006) both addressed the misuse of articles. Chodorow et al. (2007), Gamon et al. (2008), Hermet et al. (2008) proposed several techniques in detecting and correcting proposition errors. In detecting errors and giving suggestions, Liu et al. (2000), Gamon et al. (2008) and Hermet et al. (2008) make use of

	Neither	$D_{QG} \cap D_{YK}$	$D_{QG} - D_{YK}$	$D_{YK} - D_{QG}$
number of instances	50	19	30	1
average edit distance	2.88	0.05	2.17	1
percentage of identical pairs	0.40	0.95	0.5	0
average O length	14.18	9.00	12.53	17
average R length	14.98	9.05	12.47	16
QG cross entropy	N/A	81.85	139.36	N/A
YK cross entropy	N/A	51.2	N/A	103.75

Table 4: This table compares the two models on a "trick" test corpus in which the earlier and later drafts are reversed. If a model is trained to prefer more fluent English sentences are the revision, it should be perplexed on this corpus.

information retrieval techniques. Chodorow et al. (2007) instead treat it as a classification problem and employed a maximum entropy classifier. Similar to our approach, Brockett et al. (2006) view error correction as a Machine Translation problem. But their translation system is built on phrase level, with the purpose of correcting local errors such as mass noun errors.

The problem of error correction at a syntactic level is less explored. Lee and Seneff (2008) examined the task of correcting verb form misuse by applying tree template matching rules. The parse tree transformation rules are learned from synthesized training data.

6 Conclusion

This paper investigates the suitability of syntax-driven MT approaches for modeling the revision writing process of ESL learners. We have considered both the Yamada & Knight tree-to-string model, which only considers syntactic information from the typically more fluent revised text, as well as Quasi-Synchronous Grammar, a tree-to-tree model that attempts to learn syntactic transformation patterns between the students' original and revised texts. Our results suggests that while QG offers a greater degree of freedom, thus allowing for a better coverage of the transformations, YK has a lower entropy on the test corpus. Moreover, when presented with an alternative "trick" corpus in which the "revision" is in fact the earlier draft, YK was more perplexed than QG. These results suggest that the YK model may be a promising approach for automatic grammar error correction.

Acknowledgments

This work has been supported by NSF Grant IIS-0745914. We thank Joel Tetreault and the anonymous reviewers for their helpful comments and suggestions.

References

Brockett, Chris, William B. Dolan, and Michael Gamon. 2006. Correcting esl errors using phrasal smt techniques. In *Proceedings of COLING-ACL 2006*, Sydney, Australia, July.

Burstein, Jill, Martin Chodorow, and Claudia Leacock. 2004. Automated essay evaluation: The criterion online writing service. *AI Magazine*, 25(3).

Charniak, Eugene, Kevin Knight, and Kenji Yamada. 2003. Syntax-based language models for machine translation. In *Proc. MT Summit IX*, New Orleans, Louisiana, USA.

Chodorow, Martin, Joel Tetreault, and Na-Rae Han. 2007. Detection of grammatical errors involving prepositions. In *Proceedings of the 4th ACL-SIGSEM Workshop on Prepositions*, Prague, Czech Republic.

Das, Dipanjan and Noah A. Smith. 2009. Paraphrase identification as probabilistic quasi-synchronous recognition. In *Proceedings of ACL-IJCNLP 2009*, Suntec, Singapore, August.

Foster, Jennifer, Joachim Wagner, and Josef van Genabith. 2008. Adapting a WSJ-trained

parser to grammatically noisy text. In *Proceedings of the 46th ACL on Human Language Technologies: Short Papers*, Columbus, Ohio.

Gamon, Michael, Jianfeng Gao, Chris Brockett, Alexandre Klementiev, William B. Dolan, Dmitriy Belenko, and Lucy Vanderwende. 2008. Using contextual speller techniques and language modeling for ESL error correction. In *Proceedings of IJCNLP*, Hyderabad, India.

Han, Na-Rae, Martin Chodorow, and Claudia Leacock. 2006. Detecting errors in English article usage by non-native speakers. *Natural Language Engineering*, 12(02).

Han, Na-Rae, Joel Tetreault, Soo-Hwa Lee, and Jin-Young Han. 2010. Using an error-annotated learner corpus to develop and ESL/EFL error correction system. In *Proceedings of LREC 2010*, Valletta, Malta.

Hermet, Matthieu, Alain Désilets, and Stan Szpakowicz. 2008. Using the web as a linguistic resource to automatically correct Lexico-Syntactic errors. In *Proceedings of the LREC*, volume 8.

Klein, Dan and Christopher Manning. 2004. Corpus-based induction of syntactic structure: Models of dependency and constituency. In *Proceedings of ACL 2004*, Barcelona, Spain.

Knight, Kevin and Ishwar Chander. 1994. Automated postediting of documents. In *Proceedings of AAAI-94*, Seattle, Washington.

Lee, John and Stephanie Seneff. 2008. Correcting misuse of verb forms. *Proceedings of the 46th ACL, Columbus*.

Liu, Ting, Ming Zhou, Jianfeng Gao, Endong Xun, and Changning Huang. 2000. PENS: a machine-aided english writing system for chinese users. In *Proceedings of the 38th ACL*, Hong Kong, China.

Lopez, Adam. 2008. Statistical machine translation. *ACM Computing Surveys*, 40(3), September.

Melamed, I. Dan, Giorgio Satta, and Ben Wellington. 2004. Generalized multitext grammars. In *Proceedings of the 42nd ACL*, Barcelona, Spain.

Nagata, Ryo, Atsuo Kawai, Koichiro Morihiro, and Naoki Isu. 2006. A feedback-augmented method for detecting errors in the writing of learners of english. In *Proceedings of COLING-ACL 2006*, Sydney, Australia, July.

Och, Franz Josef and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1).

Smith, David A. and Jason Eisner. 2006. Quasi-synchronous grammars: Alignment by soft projection of syntactic dependencies. In *Proceedings on the Workshop on Statistical Machine Translation*, New York City, June.

Smith, David A. and Jason Eisner. 2009. Parser adaptation and projection with quasi-synchronous grammar features. In *Proceedings of EMNLP 2009*, Singapore, August.

Wang, Mengqiu, Noah A. Smith, and Teruko Mitamura. 2007. What is the Jeopardy model? a quasi-synchronous grammar for QA. In *Proceedings of EMNLP-CoNLL 2007*, Prague, Czech Republic, June.

Wu, Dekai. 1995. Stochastic inversion transduction grammars, with application to segmentation, bracketing, and alignment of parallel corpora. In *Proc. of the 14th Intl. Joint Conf. on Artificial Intelligence*, Montreal, Aug.

Yamada, Kenji and Kevin Knight. 2001. A syntax-based statistical translation model. In *Proceedings of the 39th ACL*, Toulouse, France.

Yi, Xing, Jianfeng Gao, and William B Dolan. 2008. A web-based english proofing system for english as a second language users. In *Proceedings of IJCNLP*, Hyderabad, India.

Chasing the ghost: recovering empty categories in the Chinese Treebank

Yaqin Yang
Computer Science Department
Brandeis University
yaqin@cs.brandeis.edu

Nianwen Xue
Computer Science Department
Brandeis University
xuen@cs.brandeis.edu

Abstract

Empty categories represent an important source of information in syntactic parses annotated in the generative linguistic tradition, but empty category recovery has only started to receive serious attention until very recently, after substantial progress in statistical parsing. This paper describes a unified framework in recovering empty categories in the Chinese Treebank. Our results show that given skeletal gold standard parses, the empty categories can be detected with very high accuracy. We report very promising results for empty category recovery for automatic parses as well.

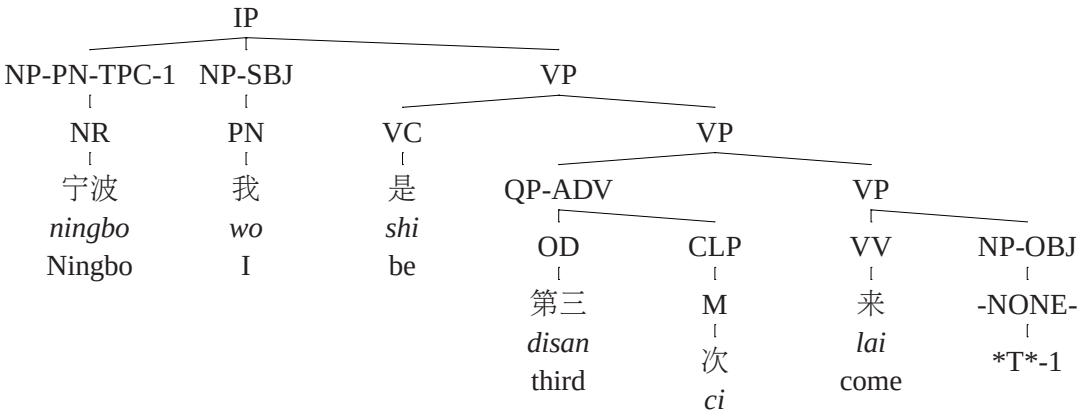
1 Introduction

The use of empty categories to represent the syntactic structure of a sentence is the hallmark of the generative linguistics and they represent an important source of information in treebanks annotated in this linguistic tradition. The use of empty categories in the annotation of treebanks started with the Penn Treebank (Marcus et al., 1993), and this practice is continued in the Chinese Treebank (CTB) (Xue et al., 2005) and the Arabic Treebank, the Penn series of treebanks. Empty categories come in a few different varieties, serving different purposes. One use of empty categories is to mark the extraction site of an dislocated phrase, thus effectively reconstructing the canonical structure of a sentence, allowing easy extraction of its predicate-argument structure. For example, in Figure 1, the empty category ***T*-1** is coindexed with the dislocated topic NP 宁波 ("Ningbo"), indicating that the canonical position of this NP is next to the verb 来 ("come"). The empty category effectively localizes the syntactic dependency between the verb and this NP, making it easier to detect and extract this relation.

Marking the extraction site of a dislocated item is not the only use of empty categories. For languages like Chinese, empty categories are also used to represent dropped pronouns. Chinese is a pro-drop language (Huang, 1989) and subject pronouns are routinely dropped. Recovering these elliptical elements is important to many natural language applications. When translated into another language, for example, these dropped pronouns may have to be made explicit and replaced with overt pronouns or noun phrases if the target language does not allow dropped pronouns.

Although empty categories have been an integral part of the syntactic representation of a sentence ever since the Penn Treebank was first constructed, it is only recently that they are starting to receive the attention they deserve. Works on automatic detection of empty categories started to emerge (Johnson, 2002; Dienes and Dubey, 2003; Campbell, 2004; Gabbard et al., 2006) after substantial progress has been made in statistical syntactic parsing. This progress has been achieved after over a decade of intensive research on syntactic parsing that has essentially left the empty categories behind (Collins, 1999; Charniak, 2000). Empty categories were and still are routinely pruned out in parser evaluations (Black et al., 1991). They have been excluded from the parser development and evaluation cycle not so much because their importance was not understood, but because researchers haven't figured out

Coling 2008: Poster Volume, pages 1382–1390,
Beijing, August 2010



"Ningbo, this is the third time I came here."

Figure 1: A CTB tree with empty categories

a way to incorporate the empty category detection in the parsing process. In fact, the detection of empty categories relies heavily on the other components of the syntactic representation, and as a result, empty category recovery is often formulated as postprocessing problem after the skeletal structure of a syntactic parse has been determined. As work on English has demonstrated, empty category detection can be performed with high accuracy given high-quality skeletal syntactic parses as input.

Because Chinese allows dropped pronouns and thus has more varieties of empty categories than languages like English, it can be argued that there is added importance in Chinese empty category detection. However, to our knowledge, there has been little work in this area, and the work we report here represents the first effort in Chinese empty category detection. Our results are promising, but they also show that Chinese empty category detection is a very challenging problem mostly because Chinese syntactic parsing is difficult and still lags significantly behind the state of the art in English parsing. We show that given skeletal gold-standard parses (with empty categories pruned out), the empty detection can be performed with a fairly high accuracy of almost 89%. The performance drops significantly, to 63%, when the output of an automatic parser is used.

The rest of the paper is organized as follows. In Section 2, we formulate the empty category detection as a binary classification problem where each word is labeled as either having a empty category before it or not. This makes it possible to use any standard machine learning technique to solve this problem. The key is to find the appropriate set of features. Section 3 describes the features we use in our experiments. We present our experimental results in Section 4. There are two experimental conditions, one with gold standard treebank parses (stripped of empty categories) as input and the other with automatic parses. Section 5 describes related work and Section 6 conclude our paper.

2 Formulating the empty category detection as a tagging problem

In the CTB, empty categories are marked in a parse tree which represents the hierarchical structure of a sentence, as illustrated in Figure 1. There are eight types of empty categories annotated in the CTB, and they are listed in Table 1. Among them, *pro* and *PRO* are used to represent nominal empty categories, *T* and *NP* are used to represent traces of dislocated items, *OP* is used to represent empty relative pronouns in relative clauses, and *RNR* is used to represent pseudo attachment. The reader is referred to the CTB bracketing manual (Xue and Xia, 2000) for detailed descriptions and examples. As can be seen from Table 1, the distribution of these empty categories is very uneven, and many of these empty categories do not occur very often.

EC Type	count	Description
pro	2024	small pro
PRO	2856	big pro
T	4486	trace for extraction
RNR	217	right node raising
OP	879	operator
*	132	trace for raising

Table 1: Empty categories in CTB.

As a first step of learning an empty category model, we treat all the empty categories as a unified type, and for each word in the sentence, we only try to decide if there is an empty category before it. This amounts to an empty category detection task, and the objective is to first locate the empty categories without attempting to determine the specific empty category type. Instead of predicting the locations of the empty categories in a parse tree and having a separate classifier for each syntactic construction where an empty category is likely to occur, we adopt a linear view of the parse tree and treat empty categories, along with overt word tokens, as leaves in the tree. This allows us to identify the location of the empty categories in relation to overt word tokens in the same sentence, as illustrated in Example (1):

(1) 宁波 我 是 第三 次 来 *T* 。

In this representation, the position of the empty category can be defined either in relation to the previous or the next word, or both. To make this even more amenable to machine learning approaches, we further reformulate the problem as a tagging problem so that each overt word is labeled either with EC, indicating there is an empty category *before* this word, or NEC, indicating there is no empty category. This reformulated representation is illustrated in Example (2):

(2) 宁/NEC 波/NEC 我/NEC 是/NEC 第/NEC 三/NEC 次/NEC 来/NEC 。/EC

In (2), the EC label attached to the final period indicates that there is an empty category before this punctuation mark. There is a small price to pay with this representation: when there is more than one empty category before a word, it is indistinguishable from cases where there is only one empty category. What we have gained is a simple unified representation for all empty categories that lend itself naturally to machine learning approaches. Another advantage is that for natural language applications that do not need the full parse trees but only need the empty categories, this representation provides an easy-to-use representation for those applications. Since this linearlized representation is still aligned with its parse tree, we still have easy access to the full hierarchical structure of this tree from which useful features can be extracted.

3 Features

Having modeled empty category detection as a machine learning task, feature selection is crucial to successfully finding a solution to this problem. The machine learning algorithm scans the words in a sentence from left to right one by one and determine if there is an empty category before it. When the sentence is paired with its parse tree, the feature space is all the surrounding words of the target word as well as the syntactic parse for the sentence. The machine learning algorithm also has access to the empty category labels (EC or NEC) of all the words before the current word. Figure 2 illustrates the feature space for the last word (a period) in the sentence.

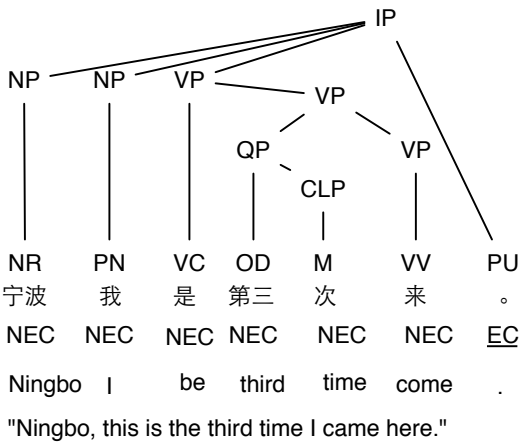


Figure 2: Feature space of empty category detection

For purposes of presentation, we divide our features into lexical and syntactic features. The

lexical features are different combinations of the words and their parts of speech (POS), while syntactic features are the structural information gathered from the nonterminal phrasal labels and their syntactic relations.

3.1 Lexical features

The lexical features are collected from a narrow window of five words and their POS tags. If the target word is a verb, the lexical features also include transitivity information of this verb, which is gathered from the CTB. A transitivity lexicon is induced from the CTB by checking whether a verb has a right NP or IP sibling. Each time a verb is used as a transitive verb (having a right NP or IP sibling), its transitive count is incremented by one. Conversely, each time a verb is used as an intransitive verb (not having a right NP or IP sibling), its intransitive use is incremented by one. The resulting transitivity lexicon after running through the entire Chinese Treebank consists of a list of verbs with frequencies of their transitive and intransitive uses. A verb is considered to be transitive if its intransitive count in this lexicon is zero or if its transitive use is more than three times as frequent as its intransitive use. Similarly, a verb is considered to be intransitive if its transitive count is zero or if its intransitive use is at least three times as frequent as its transitive use. The full list of lexical features is presented in Table 2.

Feature Names	Description
word(0)	Current word
word(-1)	Previous word
pos(0)	POS of current word
pos(-1,0)	POS of previous and current word
pos(0, 1)	POS of current and next word
pos(0, 1, 2)	POS of current & next word, & word 2 after
pos(-2, -1)	POS of previous word & word 2 before
word(-1), pos(0)	Previous word & POS of current word
pos(-1),word(0)	POS of previous word& current word
trans(0)	current word is transitive or intransitive verb
prep(0)	true if POS of current word is a preposition

Table 2: Feature set.

3.2 Syntactic features

Syntactic features are gathered from the CTB parses stripped of function tags and empty categories when the gold standard trees are used as input. The automatic parses used as input to our system are produced by the Berkeley parser. Like most parsers, the Berkeley parser does not reproduce the function tags and empty categories in the original trees in the CTB. Syntactic features capture the syntactic context of the target word, and as we shall show in Section 4, the syntactic features are crucial to the success of empty category detection. The list of syntactic features we use in our system include:

1. **1st-IP-child**: True if the current word is the first word in the lowest IP dominating this word.
2. **1st-word-in-subjectless-IP**: True if the current word starts an IP with no subject. Subject is detected heuristically by looking at left sisters of a VP node. Figure 3 illustrates this feature for the first word in a sentence where the subject is a dropped pronoun.
3. **1st-word-in-subjectless-IP+POS**: POS of the current word if it starts an IP with no subject.
4. **1st-VP-child-after-PU**: True if the current word is the first terminal child of a VP following a punctuation mark.
5. **NT-in-IP**: True if POS of current word is NT, and it heads an NP that does not have a subject NP as its right sister.
6. **verb-in-NP/VP**: True if the current word is a verb in an NP/VP.
7. **parent-label**: Phrasal label of the parent of the current node, with the current node always corresponding to a terminal node in the parse tree.
8. **has-no-object**: True If the previous word is a transitive verb and this verb does not take an object.

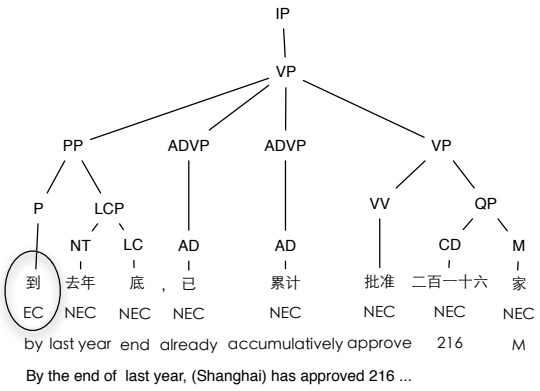


Figure 3: First word in a subject-less IP

Empty categories generally occur in clausal or phrasal boundaries, and most of the features are designed to capture such information. For example, the five feature types, *1st-IP-child*, *1st-word-in-subjectless-IP*, *1st-word-in-subjectless-IP*, *1st-VP-child-after-PU* and *NT-in-IP* all represent the left edge of a clause (IP) with some level of granularity. *parent label* and *verb-in-NP/VP* represent phrases within which empty categories typically occur do not occur. The *has-no-object* feature is intended to capture transitive uses of a verb when the object is missing.

4 Experiments

Given that our approach is independent of specific machine learning techniques, many standard machine learning algorithms can be applied to this task. For our experiment we built a Maximum Entropy classifier with the Mallet toolkit[1].

4.1 Data

In our experiments, we use a subset of the CTB 6.0. This subset is further divided into training (files chtb_0081 thorough chtb_0900), development (files chtb_0041 through chtb_0080) and test sets (files chtb_0001 through chtb_0040, files chtb_0901 through chtb_0931). The reason for not using the entire Chinese Treebank is that the data in the CTB is from a variety of different sources and the automatic parsing accuracy is very uneven across these different sources.

[1]http://mallet.cs.umass.edu

4.2 Experimental conditions

Two different kinds of data sets were used in the evaluation of our method: 1) gold standard parse trees from the CTB; and 2) automatic parses produced by the Berkeley parser[2] .

4.2.1 Gold standard parses

There are two experimental conditions. In our first experiment, we use the gold standard parse trees from the CTB as input to our classifier. The version of the parse tree that we use as input to our classifier is stripped of the empty category information. What our system effectively does is to restore the empty categories given a skeletal syntactic parse. The purpose of this experiment is to establish a topline and see how accurately the empty categories can be restored given a "correct" parse.

4.2.2 Automatic parses

To be used in realistic scenarios, the parse trees need to be produced automatically from raw text using an automatic parser. In our experiments we use the Berkeley Parser as a representative of the state-of-the-art automatic parsers. The input to the Berkeley parser is words that have already been segmented in the CTB. Obviously, to achieve fully automatic parsing, the raw text should be automatically segmented as well. The Berkeley parser comes with a fully trained model, and to make sure that none of our test and development data is included in the training data in the original model, we retrained the parser with our training set and used the resulting model to parse the documents in the development and test sets.

When training our empty category model using automatic parses, it is important that the quality of the parses match between the training and test sets. So the automatic parses in the training set are acquired by first training the parser with 4/5 of the data and using the resulting model to parse the remaining 1/5 of the data that has been held out. Measured by the ParsEval metric (Black et al., 1991), the parser accuracy stands at 80.3% (F-score), with a precision of 81.8% and a recall of 78.8% (recall).

[2]http://code.google.com/p/berkeleyparser

4.3 Evaluation metrics

We use precision, recall and F-measure as our evaluation metrics for empty category detection. Precision is defined as the number of correctly identified Empty Categories (ECs) divided by the total number of ECs that our system produced. Recall is defined as the number of correctly identified ECs divided by the total number of EC labels in the CTB gold standard data. F-measure is defined as the geometric mean of precision and recall.

$$R = \frac{\text{\# of correctly detected EC}}{\text{\# of EC tagged in corpus}} \quad (1)$$

$$P = \frac{\text{\# of correctly detected EC}}{\text{\# of EC reported by the system}} \quad (2)$$

$$F = \frac{2}{1/R + 1/P} \quad (3)$$

4.4 Overall EC detection performance

We report our best result for the gold standard trees and the automatic parses produced by the Berkeley parser in Table 3. These results are achieved by using all lexical and syntactic features presented in Section 3.

Data	Prec.(%)	Rec.(%)	F(%)
Gold	95.9 (75.3)	83.0 (70.5)	89.0 (72.8)
Auto	80.3 (57.9)	52.1 (50.2)	63.2 (53.8)

Table 3: Best results on the gold tree.

As shown in Table 3, our feature set works well for the gold standard trees. Not surprisingly, the accuracy when using the automatic parses is lower, with the performance gap between using the gold standard trees and the Berkeley parser at 25.8% (F-score). When the automatic parser is used, although the precision is 80.3%, the recall is only 52.1%. As there is no similar work in Chinese empty category detection using the same data set, for comparison purposes we established a baseline using a rule-based approach. The rule-based algorithm captures two most frequent locations of empty categories: the subject and the object positions. Our algorithm labels the first word within a VP with EC if the VP does not have a subject NP. Similarly, it assigns the EC label to the word immediately following a transitive verb if it does not have an NP or IP object. Since the missing subjects and objects account for most of the empty categories in Chinese, this baseline covers most of the empty categories. The baseline results are also presented in Table 3 (in brackets). The baseline results using the gold standard trees are 75.3% (precision), 70.5% (recall), and 72.8% (F-score). Using the automatic parses, the results are 57.9% (precision), 50.2% (recall), and 53.8% (F-score) respectively. It is clear from our results that our machine learning model beats the rule-based baseline by a comfortable margin in both experimental conditions. Table 4 breaks down our results by empty category types. Notice that we did not attempt to predict the specific empty category type. This only shows the percentage of empty categories our model is able to recover (recall) for each type. As our model does not predict the specific empty category type, only whether there is an empty category before a particular word, we cannot compute the precision for each empty category type. Nevertheless, this breakdown gives us a sense of which empty category is easier to recover. For both experimental conditions, the empty category that can be recovered with the highest accuracy is ***PRO***, an empty category often used in subject/object control constructions. ***pro*** seems to be the category that is most affected by parsing accuracy. It has the widest gap between the two experimental conditions, at more than 50%.

EC Type	Total	Correct	Recall(%)
pro	290	274/125	94.5/43.1
PRO	299	298/196	99.7/65.6
T	578	466/338	80.6/58.5
RNR	32	22/20	68.8/62.5
OP	134	53/20	40.0/14.9
*	19	9/5	47.4/26.3

Table 4: Results of different types of empty categories.

4.5 Comparison of feature types

To investigate the relative importance of lexical and syntactic features, we experimented with using just the lexical or syntactic features under both experimental conditions. The results are pre-

sented in Table 5. Our results show that when using only the lexical features, the drop in accuracy is small when automatic parses are used in place of gold standard trees. However, when using only the syntactic features, the drop in accuracy is much more dramatic. In both experimental conditions, however, syntactic features are more effective than the lexical features, indicating the crucial importance of high-quality parses to successful empty category detection. This makes intuitive sense, given that all empty categories occupy clausal and phrasal boundaries that can only defined in syntactic terms.

Data	Prec.(%)	Rec.(%)	F(%)
Lexical	79.7/77.3	47.6/39.9	59.6/52.7
Syntactic	95.9/78.0	70.0/44.5	81.0/56.7

Table 5: Comparison of lexical and syntactic features.

4.6 Comparison of individual features

Given the importance of syntactic features, we conducted an experiment trying to evaluate the impact of each individual syntactic feature on the overall empty category detection performance. In this experiment, we kept the lexical feature set constant, and switched off the syntactic features one at a time. The performance of the different syntactic features is shown in Table 6. The results here assume that automatic parses are used. The first row is the result of using all features (both syntactic and lexical) while the last row is the result of using only the lexical features. It can be seen that syntactic features contribute more than 10% to the overall accuracy. The results also show that features (e.g., *1st-IP-child*) that capture clause boundary information tend to be more discriminative and they occupy the first few rows of a table that sorted based on feature performance.

5 Related work

The problem of empty category detection has been studied both in the context of reference resolution and syntactic parsing. In the reference resolution literature, empty category detection manifests itself in the form of zero anaphora (or zero pronoun)

Feature Name	Prec.(%)	Rec.(%)	F(%)
all	80.3	52.1	63.2
1st-IP-child	79.8	49.2	60.8
1st-VP-child-after-PU	79.7	50.5	61.8
NT-in-IP	79.4	50.8	61.9
1st-word-in-subjectless-IP+Pos	79.5	51.1	62.2
has-no-object	80.0	51.1	62.4
1st-word-in-subjectless-IP	79.4	51.5	62.5
verb-in-NP/VP	79.9	52.0	63.0
parent-label	79.4	52.4	63.1
only lexical	77.3	39.9	52.7

Table 6: Performance for individual syntactic features with automatic parses.

detection and resolution. Zero anaphora resolution has been studied as a computational problem for many different languages. For example, (Ferrández and Peral, 2000) describes an algorithm for detecting and resolving zero pronouns in Spanish texts. (Seki et al., 2002) and (Lida et al., 2007) reported work on zero pronoun detection and resolution in Japanese.

Zero anaphora detection and resolution for Chinese has been studied as well. Converse (2006) studied Chinese pronominal anaphora resolution, including zero anaphora resolution, although there is no attempt to automatically detect the zero anaphors in text. Her work only deals with anaphora resolution, assuming the zero anaphors have already been detected. Chinese zero anaphora identification and resolution have been studied in a machine learning frameworking in (Zhao and Ng, 2007) and (Peng and Araki, 2007).

The present work studies empty category recovery as part of the effort to fully parse natural language text and as such our work is not limited to just recovering zero anaphors. We are also interested in other types of empty categories such as traces. Our work is thus more closely related to the work of (Johnson, 2002), (Dienes and Dubey, 2003), (Campbell, 2004) and (Gabbard et

al., 2006).

Johnson (2002) describes a pattern-matching algorithm for recovering empty nodes from phrase structure trees. The idea was to extract minimal connected tree fragments that contain an empty node and its antecedent(s), and to match the extracted fragments against an input tree. He evaluated his approach both on Penn Treebank gold standard trees stripped of the empty categories and on the output of the Charniak parser (Charniak, 2000).

(Dienes and Dubey, 2003) describes an empty detection method that is similar to ours in that it treats empty detection as a tagging problem. The difference is that the tagging is done without access to any syntactic information so that the identified empty categories along with word tokens in the sentence can then be fed into a parser. The success of this approach depends on strong local cues such as infinitive markers and participles, which are non-existent in Chinese. Not surprisingly, our model yields low accuracy if only lexical features are used.

Cambell (2004) proposes an algorithm that uses linguistic principles in empty category recovery. He argues that a rule-based approach might perform well for this problem because the locations of the empty categories, at least in English, are inserted by annotators who follow explicit linguistic principles.

Yuqing(2007) extends (Cahill et al., 2004) 's approach for recovering English non-local dependencies and applies it to Chinese. This paper proposes a method based on the Lexical-Functional Grammar f-structures, which differs from our approach. Based on parser output trees including 610 files from the CTB, the authors of this paper claimed they have achieved 64.71% f-score for trace insertion and 54.71% for antecedent recovery.

(Gabbard et al., 2006) describes a more recent effort to fully parse the Penn Treebank, recovering both the function tags and the empty categories. Their approach is similar to ours in that they treat empty category recovery as a post-processing process and use a machine learning algorithm that has access to the skeletal information in the parse tree. Their approach is different from ours in that they have different classifiers for different types of empty categories.

Although generally higher accuracies are reported in works on English empty category recovery, cross-linguistic comparison is difficult because both the types of empty categories and the linguistic cues that are accessible to machine learning algorithms are different. For example, there are no empty complementizers annotated in the CTB while English does not allow dropped pronouns.

6 Conclusion and future work

We describe a unified framework to recover empty categories for Chinese given skeletal parse trees as input. In this framework, empty detection is formulated as a tagging problem where each word in the sentence receives a tag indicating whether there is an empty category before it. This advantage of this approach is that it is amenable to learning-based approaches and can be addressed with a variety of machine learning algorithms. Our results based on a Maximum Entropy model show that given skeletal gold standard parses, empty categories can be recovered with very high accuracy (close to 90%). We also report promising results (over 63%). when automatic parses produced by an off-the-shelf parser is used as input.

Detecting empty categories is only the first step towards fully reproducing the syntactic representation in the CTB, and the obvious next step is to also classify these empty categories into different types and wherever applicable, link the empty categories to their antecedent. This is the line of research we intend to pursue in our future work.

Acknowledgment

This work is supported by the National Science Foundation via Grant No. 0910532 entitled "Richer Representations for Machine Translation". All views expressed in this paper are those of the authors and do not necessarily represent the view of the National Science Foundation.

References

Black, E., S. Abney, D. Flickinger, C. Gdaniec, R. Grishman, P. Harrison, D. Hindle, R. Ingria, F. Jelinek, J. Klavans, M. Liberman, M. Marcus, S. Roukos, B. Santorini, and T. Strzalkowski. 1991. A procedure for quantitively comparing the syntactic coverage of English grammars. In *Proceedings of the DARPA Speech and Natural Language Workshop*, pages 306–311.

Cahill, Aoife, Michael Burke, Ruth O' Donovan, Josef van Genabith, and Andy Way. 2004. Long-Distance Dependency Resolution in Automatically Acquired Wide-Coverage PCFG-Based LFG Approximations. In *In Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*.

Campbell, Richard. 2004. Using linguistic principles to recover empty categories. In *Proceedings of the 42nd Annual Meeting on Association For Computational Linguistics*.

Charniak, E. 2000. A Maximum-Entropy-Inspired Parser. In *Proceedings of NAACL-2000*, pages 132–139, Seattle, Washington.

Collins, Michael. 1999. *Head-driven Statistical Models for Natural Language Parsing*. Ph.D. thesis, University of Pennsylvania.

Converse, Susan. 2006. *Pronominal anaphora resolution for Chinese*. Ph.D. thesis.

Dienes, Péter and Amit Dubey. 2003. Deep syntactic processing by combining shallow methods. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, volume 1.

Ferrández, Antonio and Jesús Peral. 2000. A computational approach to zero-pronouns in Spanish. In *Proceedings of the 38th Annual Meeting on Association For Computational Linguistics*.

Gabbard, Ryan, Seth Kulick, and Mitchell Marcus. 2006. Fully parsing the penn treebank. In *Proceedings of HLT-NAACL 2006*, pages 184–191, New York City.

Guo, Yuqing, Haifeng Wang, and Josef van Genabith. 2007. Recovering Non-Local Dependencies for Chinese. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*.

Huang, James C.-T. 1989. Pro drop in Chinese, a generalized control approach. In O, Jaeggli and K. Safir, editors, *The Null Subject Parameter*. D. Reidel Dordrecht.

Johnson, Mark. 2002. A simple pattern-matching algorithm for recovering empty nodes and their antecedents. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*.

Lida, Ryu, Kentaro Inui, and Yuji Matsumoto. 2007. Zero-anaphora resolution by learning rich syntactic pattern features. *ACM Transactions on Asian Language Information Processing*, pages 1–22.

Marcus, Mitchell P., Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of english: The penn treebank. *Computational Linguistics*, 19(2):313–330.

Peng, Jing and Kenji Araki. 2007. Zero-anaphora resolution in chinese using maximum entropy. *IEICE - Trans. Inf. Syst.*, E90-D(7):1092–1102.

Seki, Kazuhiro, Atsushi Fujii, and Tetsuya Ishikawa. 2002. A probabilistic method for analyzing Japanese anaphora integrating zero pronoun detection and resolution. In *Proceedings of the 19th international Conference on Computational Linguistics*, volume 1.

Xue, Nianwen and Fei Xia. 2000. The Bracketing Guidelines for Penn Chinese Treebank Project. Technical Report IRCS 00-08, University of Pennsylvania.

Xue, Nianwen, Fei Xia, Fu dong Chiou, and Martha Palmer. 2005. The Penn Chinese TreeBank: Phrase Structure Annotation of a Large Corpus. *Natural Language Engineering*, 11(2):207–238.

Zhao, Shanheng and Hwee Tou Ng. 2007. Identification and Resolution of Chinese Zero Pronouns: A Machine Learning Approach. In *Proceedings of EMNLP-CoNLL Joint Conference*, Prague, Czech Republic.

Unsupervised Part of Speech Tagging Using Unambiguous Substitutes from a Statistical Language Model

Mehmet Ali Yatbaz
Dept. of Computer Engineering
Koç University
myatbaz@ku.edu.tr

Deniz Yuret
Dept. of Computer Engineering
Koç University
dyuret@ku.edu.tr

Abstract

We show that unsupervised part of speech tagging performance can be significantly improved using likely substitutes for target words given by a statistical language model. We choose unambiguous substitutes for each occurrence of an ambiguous target word based on its context. The part of speech tags for the unambiguous substitutes are then used to filter the entry for the target word in the word–tag dictionary. A standard HMM model trained using the filtered dictionary achieves 92.25% accuracy on a standard 24,000 word corpus.

1 Introduction

We define the unsupervised part-of-speech (POS) tagging problem as predicting the correct part-of-speech tag of a word in a given context using an unlabeled corpus and a dictionary with possible word–tag pairs[0] The performance of an unsupervised POS tagging system depends highly on the quality of the word–tag dictionary (Banko and Moore, 2004). We propose a dictionary filtering procedure based on likely substitutes suggested by a statistical language model. The procedure reduces the word–tag dictionary size and leads to significant improvement in the accuracy of the POS models.

Probabilistic models such as the hidden Markov model (HMM) trained by expectation maximization (EM), maximum a posteriori (MAP) estimation, and Bayesian methods have been used to solve the unsupervised POS tagging problem (Merialdo, 1994; Goldwater and Griffiths, 2007). All of these approaches first learn the parameters connecting the hidden structure to the observed sequence of variables and then identify the most probable values of the hidden structure for a given observed sequence. They differ in the way they estimate the model parameters. HMM-EM estimates model parameters by using the maximum likelihood estimation (MLE), MAP defines a prior distribution over parameters and finds the parameter values that maximize the posterior distribution given data, and Bayesian methods integrate over the posterior of the parameters to incorporate all possible parameter settings into the estimation process. Some baseline results and performance reports from the literature are presented in Table 1.

(Johnson, 2007) criticizes the standard EM based HMM approaches because of their poor performance on the unsupervised POS tagging and their tendency to assign equal number of words to each hidden state. (Mitzenmacher, 2004) further claims that words have skewed POS tag distributions, and a Bayesian method with sparse priors over the POS tags may perform better than HMM estimated with EM. (Goldwater and Griffiths, 2007) uses a fully Bayesian HMM model that averages over all possible parameter values. Their model achieves 86.8% tagging accuracy with sparse POS priors and outperforms 74.50% accuracy of the standard second order HMM-EM (3-gram tag model) on a 24K word subset of the Penn Treebank corpus. (Smith and Eisner, 2005) take a different approach and use the conditional random fields estimated using contrastive estimation which outperforms the HMM-EM and

[0]In the POS literature the term "unsupervised" is typically used to describe systems that do not directly use the tagged data. However, many of the unsupervised systems, including ours, uses the tag–word dictionary.

Coling 2008: Poster Volume, pages 1391–1398,
Beijing, August 2010

Accuracy	System
64.2	Random baseline
74.4	Second order HMM
82.0	First order HMM
86.8	Fully Bayesian approach with sparse priors (Goldwater and Griffiths, 2007)
88.6	CRF/CE (Smith and Eisner, 2005)
91.4	EM-HMM with language specific information, good initialization and manual adjustments to standard dictionary (Goldberg et al., 2008)
91.8	Minimized models for EM-HMM with 100 random restarts (Ravi and Knight, 2009).
94.0	Most frequent tag baseline

Table 1: Tagging accuracy on a 24K-word corpus. All the systems – except (Goldwater and Griffiths, 2007) – use the same 45 tag dictionary that is constructed from the Penn Treebank.

Bayesian methods by achieving 88.6% accuracy on the same 24K corpus.

Despite the fact that HMM-EM has a poor reputation in POS literature (Goldberg et al., 2008) has shown that with good initialization together with some language specific features and language dependent constraints HMM-EM achieves 91.4% accuracy. Aside from the language specific information and the good initialization, they also manually reduce the noise in the word–tag dictionary.

(Ravi and Knight, 2009) focus on the POS tag collection to find the smallest POS model that explain the data. They apply integer programming to construct a minimal bi-gram POS tag set and use this set to constrain the training phase of the EM algorithm. The model trained by EM is used to reduce the dictionary and these steps are iteratively repeated until no further improvement is observed. Their model achieves 91.6% accuracy on the 24K word corpus (with 100 random starts this goes up to 91.8%). The main advantage of this model is the restriction of the tag set so that rare POS tags or the noise in the corpus do not get incorporated into the estimation process.

Language models for disambiguation: Recent work has shown that statistical language models trained on large amounts of unlabeled text can be used to improve the performance on various disambiguation problems. The language model is used to generate likely substitutes for the target word in the given context and these benefit the disambiguation process to the extent that the likely substitutes are unambiguous or have different ambiguities compared to the target word. Using statistical language models based on large corpora for unsupervised word sense disambiguation and lexical substitution has been explored in (Yuret, 2007; Hawker, 2007; Yuret and Yatbaz, 2010). Unsupervised morphological disambiguation in agglutinative languages using likely substitutes has been shown to improve on standard methods in (Yatbaz and Yuret, 2009).

In this paper we use the statistical language model to reduce the possible number of tags per word to help the disambiguation process. Specifically we assume that the same hidden tag sequence that has generated a particular test sentence can also generate artificial sentences where one of the words has been replaced with a likely substitute. POS tags of the likely substitutes can then be used to reduce the tag set of the target word. Thus, the substitutes are implicitly incorporated into the disambiguation process for reducing the noise and the rare tags in the dictionary.

Currency gyrations can **whipsaw(VB/NN)** the funds .
Currency gyrations can **withdraw(VB)** the funds .
Currency gyrations can **restore(VB)** the funds .
Currency gyrations can **modify(VB)** the funds .
Currency gyrations can **justify(VB)** the funds .
Currency gyrations can **regulate(VB)** the funds .

Table 2: Sample artificial sentences generated for a test sentence from the Penn Treebank.

Table 2 presents an example where the likely unambiguous replacements of the target word "whipsaw" for a given sentence taken from the Penn Treebank (Marcus et al., 1994) are listed. In this example each substitute is an unambiguous verb (VB), confirming our assumption that each artificial sentence comes from the same hidden sequence. For all occurrences of the word "whipsaw", our reduction algorithm will count the POS tags of the likely substitutes and remove the tags

that have not been observed from the dictionary. Assuming that the first sentence in Table 2 is the only sentence in which we observe "whipsaw", the "NN" tag of "whipsaw" will be removed.

The next section describes the details of our dictionary reduction method. Section 3 explains the details of statistical language model. We experimentally demonstrate that the word–tag dictionary reduced by the substitutes improve the performance by constraining the unsupervised model in Section 4. Finally, Section 5 comments on the results and discusses the possible extensions of our method.

2 Dictionary Reduction

Our main assumption is that likely replacements of a target word should have the same POS tag as the target word in a given context. Motivated by this idea we propose a new procedure that automatically reduces the dictionary size by using the unambiguous replacements of the target word. For all occurrences of the target word the procedure counts the POS tags of the replacement words and removes the unobserved POS tags of the target word from the dictionary.

Our approach is based on the idea that similar words in a given context should have the same tag sequence. To reduce the dictionary with the help of the replacement words similar to a target word w, we follow three rules:

1. Choose the replacement word from unambiguous substitutes that are likely to appear in the target word context.

2. Substitutes must be observed in the training corpus.

3. Count the tags of the replacement for all occurrences of the target word.

4. Remove the tags that are not observed as the tag of replacements in any occurrences of the target word.

The first rule is used to increase the likelihood of getting a replacement word with the same POS tag. The second rule makes sure that the size of the vocabulary does not change. The third rule determines the unused POS tags in all occurrences of w and finally, last rule removes the unobserved tags of w from the dictionary.

We use the standard first order HMM to test the performance of our method. In a standard n^{th} order HMM each hidden state is conditioned by its n preceding hidden states and each observation is conditioned by its corresponding hidden state. In POS tagging, the observed variable sequence is a sentence s and the hidden variables t_i are the POS tags of the words w_i in s. The HMM parameters θ can be estimated by using Baum-Welch EM algorithm on an unlabeled training corpus D (Baum, 1972). The tag sequence that maximizes $\Pr(t|s,\hat{\theta})$ can be identified by the Viterbi search algorithm.

3 Statistical Language Modeling

In order to estimate highly probable replacement words for a given word w in the context c_w, we use an n-gram language model. The context is defined as the $2n-1$ word window $w_{-n+1}\dots w_0\dots w_{n-1}$ and it is centered at the target word position. The probability of a word in a given context can be estimated as:

$$\begin{aligned} P(w_0 = w|c_w) &\propto P(w_{-n+1}\dots w_0\dots w_{n-1}) && (1)\\ &= P(w_{-n+1})P(w_{-n+2}|w_{-n+1}) \\ &\quad \dots P(w_{n-1}|w_{-n+1}^{n-2}) && (2)\\ &\propto P(w_0|w_{-n+1}^{-1})P(w_1|w_{-n+2}^{0}) \\ &\quad \dots P(w_{n-1}|w_0^{n-2}) && (3)\end{aligned}$$

where w_i^j represents the sequence of words $w_i w_{i+1}\dots w_j$. In Equation 1, $\Pr(w|c_w)$ is proportional to $\Pr(w_{-n+1}\dots w_0\dots w_{n+1})$ since the context of the target word replacements is fixed. Terms without w_0 are common for every replacement in Equation 2 therefore they have been dropped in Equation 3. Finally, because of the Markov property of n-gram language model, only $n-1$ words are used as a conditional context.

The probabilities in Equation 3 are estimated using a 4 gram language model for all the words in the vocabulary of D that are unambiguous and have a common tag with the target word w. The words with the highest $\Pr(r|c_w)$ where $r \in D$ are selected as the replacement words of w in c_w.

To get accurate domain independent probability estimates we used the Web 1T data-set (Brants and Franz, 2006), which contains the counts of word sequences up to length five in a 10^{12} word corpus derived from publicly accessible Web pages. The SRILM toolkit is used to train 5-gram language model (Stolcke, 2002). The language model parameters are optimized by using a randomly selected 24K words corpus from Penn Treebank. In order to efficiently apply the language model to a given test corpus, the vocabulary size is limited to the words seen in the test corpus.

4 Experiments

In this section we present a number of experiments measuring the performance of several variants of our algorithm. The models in this section are trained[1] and tested on the same unlabeled data therefore there aren't any out-of-vocabulary words. The experiments in this section focus on: (1) the analysis of the dictionary reduction (2) the number of the substitutes used for each ambiguous word and (3) the size of the word–tag dictionary.

4.1 Dataset

We trained HMM-EM models on a corpus that consists of the first 24K words of the Penn Treebank corpus. To be consistent with the POS tagging literature, the tag dictionary is constructed by listing all observed tags for each word in the entire Penn Treebank. Nearly 55% of the words in Penn Treebank corpus are ambiguous and the average number of tags is 2.3.

Groups	Member POS tags	Count	%
Noun	NN/NNP/NNS/NNPS	7511	31.30
Verb	VBD/VB/VBZ/VBN/VBG/VBP	3285	13.69
Adj	JJ/JJR/JJS	1718	7.16
Adv	RB/RBR	742	3.09
Pronoun	CD/PRP/PRP$	1397	5.82
Content	Noun/Verb/Adj/Adv/Pronoun	14653	61.05
Function	Other	9347	38.95
Total	All 45 POS tags	24K	100.00

Table 3: Group names, members, number and percentage of the words according to their gold POS tags.

[1]The GMTK tool is used to train HMM-EM model on an unlabeled corpus (Bilmes and Zweig, 2002).

Table 3 shows the POS speech groups and their distributions in the 24K word corpus. We report the model accuracy on several POS groups. Our motivation is to determine HMM-EM model accuracies on the subgroups before and after implementing the dictionary reduction procedure.

4.2 Baseline

Table 4 presents some standard baselines for comparison. We define a random and a most frequent tag (MFT) baseline on the 24K corpus. The random baseline is calculated by randomly picking one of the tags of each word and it also represents the amount of ambiguity in the corpus. The MFT baseline simply selects the most frequent POS tag of each word from the 1M word Penn Treebank corpus (counts of the first 24K words is not included in the 1M word corpus). If the target word does not exist in the training set, the MFT baseline randomly picks one of the possible tags of the missing word.

The first and second order HMMs can be treated as the unsupervised baselines. These unsupervised baselines are calculated by training uniformly initialized first and second order HMMs on the target corpus without any smoothing. All the initial parameters of HMM-EM are uniformly initialized to observe only the effects of the artificial sentences on the performance of HMM-EM.

The success of the MFT baseline on the $Noun$, Adj, $Pronoun$ and function word groups shows that tag distributions of the words in these groups are more skewed towards to one of the available tags. The MFT baseline performs poorly, compared to the above groups, on $Verb$, and Adv which is due to the less skewed POS tag behavior of these tags.

The POS tagging literature widely uses the second order HMM as the baseline model; however, the performance of this model can be outperformed by an unsupervised first order HMM model or a simple MFT baseline as presented in Table 4. A point worth noting is that although the first order HMM and the MFT baseline have similar content word accuracies, the MFT baseline is significantly better on the function words. This is expected since EM tends to assign words uniformly to the available POS tags. Thus EM can

	Noun	Verb	Adj	Adv	Pronoun	Content	Function	Total(%)
Random Baseline	76.98	53.87	68.46	72.98	87.64	71.59	52.64	64.21
3-gram HMM	77.43	68.16	78.06	73.32	94.85	76.88	70.45	74.38
2-gram HMM	92.22	83.84	85.22	83.96	95.56	89.42	70.49	82.05
MFT Baseline	96.11	80.30	88.56	83.15	98.75	91.28	98.25	93.99

Table 4: Percentages of words tagged correctly by different models using standard dictionary.

not capture the highly skewed behavior of function words. Moreover the amount of skewness affects the accuracy of the EM such that the performance gain of the MFT baseline over the first order HMM on function words is around 28%-30% while the performance gain on $Noun$, Adj and $Pronoun$ is around 3%-4%.

4.3 Reduced Dictionary

EM can not capture the sparse structure of the word distributions therefore it tends to assign equal number of words to each POS tag. Together with the noisy word–tag dictionary great portion of the function words are tagged with very rare POS tags. The abuse of the rare tags is presented in Table 5 in a similar fashion with (Ravi and Knight, 2009). The count of replacement word POS tags and the removed rare POS tags of 2 erroneous function words are also shown in Table 5.

Word	Tag dictionary	Gold tagging	EM tagging	Replacement POS counts
of	{RB,RP,IN}	IN(632)	IN(0)	IN(2377)
		RP(0)	RP(632)	**RP(0)**
		RB(0)	RB(0)	RB(850)
a	{LS,SYM,NNP,	DT(458)	DT(0)	DT(513)
	FW,JJ,IN,DT}	IN(1)	IN(0)	IN(317)
		JJ(2)	JJ(0)	JJ(1329)
		SYM(1)	SYM(258)	**SYM(0)**
		LS(0)	LS(230)	**LS(0)**

Table 5: Removed POS tags of the given words are shown in bold.

The results obtained with the dictionary that is reduced by using 5 replacements are presented in Table 6. Note that with reduced dictionary the uniformly initialized first order HMM-EM achieves 91.85% accuracy. Dictionary reduction also removes some of the useful tags therefore the upper–bound (oracle score) of the 24K dataset becomes 98.15% after the dictionary reduction. We execute 100 random restarts of the EM algorithm and select the model with the highest corpus likelihood, our model achieves 92.25% accuracy which is the highest accuracy reported for the 24K corpus so far.

As Table 6 shows, the effect of the dictionary reduction on the function words is higher than the effect on the content words. The main reason for this situation is, function words are frequently tagged with one of its tags which is also the reason for the high accuracy of the majority voting based baseline on the function words.

The reduced dictionary (RD) removes the rare problematic POS tags of the words as a result the accuracy on the content and function words shows a drastic improvement compared to HMM models trained with the original dictionary.

Pos groups	2-gram HMM accuracy(%)	2-gram HMM RD accuracy(%)
Noun	92.22	94.01
Verb	83.84	84.90
Adj	85.22	89.52
Adv	83.96	85.18
Pronoun	95.56	95.92
Content	89.42	91.18
Function	70.49	92.92
All	82.05	91.85

Table 6: Percentages of the correctly tagged words by different models with modified dictionary. The dictionary size is reduced by using the top 5 replacements of each target word.

4.4 More Data

In this set of experiments we doubled the size of the data and trained HMM-EM models on a corpus that consists of the first 48K words of the Penn Treebank corpus. Our aim is to observe the effect of more data on our dictionary reduction proce-

dure. Using the 5 replacements of each ambiguous word we reduce the dictionary and train a new HMM-EM model using this dictionary. The additional data together with 100 random starts increases the model accuracy to 92.47% on the 48K corpus.

Pos groups	3-gram HMM RD accuracy(%)	2-gram HMM RD accuracy(%)
Noun	89.45	93.47
Verb	85.56	88.99
Adj	86.02	87.53
Adv	94.44	95.92
Pronoun	94.08	94.04
Content	88.91	91.97
Function	92.44	92.26
All	90.31	92.09

Table 7: Percentages of the correctly tagged words by the first and second order HMM-EM model trained on the 48K corpus with reduced dictionary. The dictionary size is reduced by using the top 5 replacements of each target word.

As we mentioned before, when the model is trained using the original dictionary, the performance gap between the first order HMM the second order HMM is around 8% as presented in Table 4. On the other hand, when we use the reduced dictionary together with more data the accuracy gap between the second order and the first order HMM-EM becomes less than 2% as shown in Table 7. This confirms the hypothesis that the low performance of the second order HMM is due to data sparsity in the 24K-word dataset, and better results may be achieved with the second order HMM in larger datasets.

4.5 Number of Replacements

In this set of experiments we vary the number of artificial replacement words per each ambiguous word in s. We run our method on the 24K corpus with 1, 5, 10, 25 and 50 replacement words per ambiguous word and we present the results in Table 8. The performance of our method affected by the the number of replacements and highest score is achieved when 5 replacements are used. Incorporating the probability of the substitutes into the model rather than using a hard cutoff may be a better solution.

Number of replacements	2-gram HMM RD accuracy(%)
none	82.05
1	89.65
5	**91.85**
10	90.09
25	89.97
50	89.83

Table 8: Percentages of the correctly tagged words by the models trained on the 24K corpus with different reduced dictionaries. The dictionary size is reduced by using different number replacements.

4.6 17-Tagset

To observe the effect our method on a model with coarse grained dictionary, we collapsed the 45–tagset treebank dictionary to a 17–tagset coarse dictionary (Smith and Eisner, 2005). The POS literature after the work of Smith and Eisner follows this tradition and also tests the models on this 17–tagset. Table 9 summarizes the previously reported results on coarse grained POS tagging. Our system achieves 92.9% accuracy where the oracle accuracy of 24K dataset with the reduced 17–tagset dictionary is 98.3% and the state-of-the-art system IP+EM scores 96.8%.

Model	Accuracy	Data Size
BHMM	87.3	24K
CE+spl	88.7	24K
RD	**92.9**	24K
LDA+AC	93.4	1M
InitEM-HMM	93.8	1M
IP+EM	96.8	24K

Table 9: Performance of different systems using the coarse grained dictionary.

The IP+EM system constructs a model that describes the data by using minimum number of bigram POS tags then uses this model to reduce the dictionary size (Ravi and Knight, 2009). InitEM-HMM uses the language specific information together with good initialization and it achieves 93.8% accuracy on the 1M word treebank corpus. LDA+AC semi-supervised Bayesian model with strong ambiguity class component given the morphological features of words and scores 93.4% on the 1M word treebank corpus. (Toutanova and Johnson, 2007). CE+spl is HMM model estimated

by contrastive estimation method and achieves 88.7% accuracy (Smith and Eisner, 2005). Finally, BHMM is a fully Bayesian approach that uses sparse POS priors and scores 87.3% (Goldwater and Griffiths, 2007).

5 Contributions

In this paper we proposed a dictionary reduction method that can be applied to unsupervised tagging problems. With the help of a statistical language model, our system creates artificial replacements that are assumed to have the same POS tag as the target word and use them to reduce the size of the word–tag dictionary. To test our method we used HMM-EM as the unsupervised model. Our method significantly improves the prediction accuracy of the unsupervised first order HMM-EM system in all of the POS groups and achieves 92.25% and 92.47% word tagging accuracy on the 24K and 48K word corpora respectively. We also tested our model on a coarse grained dictionary with 17 tags and achieved an accuracy of 92.8%.

In this work, we show that unambiguous replacements of an ambiguous word can reduce the amount of the ambiguity thus replacement words might also be incorporated into the other unsupervised disambiguation problems.

Acknowledgments

This work was supported in part by the Scientific and Technical Research Council of Turkey (TÜBİTAK Project 108E228).

References

Banko, Michele and Robert C. Moore. 2004. Part of speech tagging in context. In *COLING '04: Proceedings of the 20th international conference on Computational Linguistics*, page 556, Morristown, NJ, USA. Association for Computational Linguistics.

Baum, L.E. 1972. An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes. *Inequalities*, 3(1):1–8.

Bilmes, J. and G. Zweig. 2002. The Graphical Models Toolkit: An open source software system for speech and time-series processing. In *IEEE International Conference On Acoustics Speech and Signal Processing*, volume 4, pages 3916–3919.

Brants, T. and A. Franz. 2006. Web 1T 5-gram Version 1. *Linguistic Data Consortium, Philadelphia*.

Goldberg, Y., M. Adler, and M. Elhadad. 2008. Em can find pretty good hmm pos-taggers (when given a good start). *Proceedings of ACL-08. Columbus, OH*, pages 746–754.

Goldwater, S. and T. Griffiths. 2007. A fully Bayesian approach to unsupervised part-of-speech tagging. In *Annual Meeting-Assosiation for Computational Linguistics*, volume 45, page 744.

Hawker, Tobias. 2007. Usyd: Wsd and lexical substitution using the web1t corpus. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 446–453, Prague, Czech Republic, June. Association for Computational Linguistics.

Johnson, M. 2007. Why doesnt EM find good HMM POS-taggers. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 296–305.

Marcus, M.P., B. Santorini, and M.A. Marcinkiewicz. 1994. Building a large annotated corpus of English: The Penn Treebank. *Computational linguistics*, 19(2):313–330.

Merialdo, B. 1994. Tagging english text with a probabilistic model. *Computational linguistics*, 20(2):155–171.

Mitzenmacher, M. 2004. A brief history of generative models for power law and lognormal distributions. *Internet mathematics*, 1(2):226–251.

Ravi, Sujith and Kevin Knight. 2009. Minimized models for unsupervised part-of-speech tagging. In *ACL-IJCNLP '09: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1*, pages 504–512, Morristown, NJ, USA. Association for Computational Linguistics.

Smith, Noah A. and Jason Eisner. 2005. Contrastive estimation: training log-linear models on unlabeled data. In *ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 354–362, Morristown, NJ, USA. Association for Computational Linguistics.

Stolcke, A. 2002. SRILM-an extensible language modeling toolkit. In *Seventh International Conference on Spoken Language Processing*, volume 3.

Toutanova, K. and M. Johnson. 2007. A Bayesian LDA-based model for semi-supervised part-of-speech tagging. In *Proceedings of NIPS*, volume 20.

Yatbaz, Mehmet Ali and Deniz Yuret. 2009. Unsupervised morphological disambiguation using statistical language models. In *NIPS 2009 Workshop on Grammar Induction, Representation of Language and Language Learning.*

Yuret, Deniz and Mehmet Ali Yatbaz. 2010. The noisy channel model for unsupervised word sense disambiguation. *Computational Linguistics*, 36(1), March.

Yuret, Deniz. 2007. KU: Word sense disambiguation by substitution. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 207–214, Prague, Czech Republic, June. Association for Computational Linguistics.

Jointly Identifying Entities and Extracting Relations in Encyclopedia Text via A Graphical Model Approach*

Xiaofeng YU **Wai LAM**

Information Systems Laboratory
Department of Systems Engineering & Engineering Management
The Chinese University of Hong Kong
{xfyu,wlam}@se.cuhk.edu.hk

Abstract

In this paper, we investigate the problem of entity identification and relation extraction from encyclopedia articles, and we propose a joint discriminative probabilistic model with arbitrary graphical structure to optimize all relevant subtasks simultaneously. This modeling offers a natural formalism for exploiting rich dependencies and interactions between relevant subtasks to capture mutual benefits, as well as a great flexibility to incorporate a large collection of arbitrary, overlapping and non-independent features. We show the parameter estimation algorithm of this model. Moreover, we propose a new inference method, namely collective iterative classification (CIC), to find the most likely assignments for both entities and relations. We evaluate our model on real-world data from Wikipedia for this task, and compare with current state-of-the-art pipeline and joint models, demonstrating the effectiveness and feasibility of our approach.

1 Introduction

We investigate a compound information extraction (IE) problem from encyclopedia articles, which consists of two subtasks — recognizing structured information about entities and extracting the relationships between entities. The most common approach to this problem is a pipeline architecture: attempting to perform different subtasks, namely, named entity recognition and relation extraction between recognized entities in several separate, and independent stages. Such kind of design is widely adopted in NLP.

The work described in this paper is substantially supported by grants from the Research Grant Council of the Hong Kong Special Administrative Region, China (Project No: CUHK4128/07) and the Direct Grant of the Faculty of Engineering, CUHK (Project Codes: 2050442 and 2050476). This work is also affiliated with the Microsoft-CUHK Joint Laboratory for Human-centric Computing and Interface Technologies.

The most common and simplest approach to performing compound NLP tasks is the 1-best pipeline architecture, which only takes the 1-best hypothesis of each stage and pass it to the next one. Although it is comparatively easy to build and efficient to run, this pipeline approach is highly ineffective and suffers from serious problems such as error propagation (Finkel *et al.*, 2006; Yu, 2007; Yu *et al.*, 2008). It is not surprising that, the end-to-end performance will be restricted and upper-bounded.

Usually, one can pass N-best lists between different stages in pipeline architectures, and this often gives useful improvements (Hollingshead and Roark, 2007). However, effectively making use of N-best lists often requires lots of engineering and human effort (Toutanova, 2005). On the other hand, one can record the complete distribution at each stage in a pipeline, to compute or approximate the complete distribution at the next stage. Doing this is generally infeasible, and this solution is rarely adopted in practice.

One promising way to tackle the problem of error propagation is to explore joint learning which integrates evidences from multiple sources and captures mutual benefits across multiple components of a pipeline for all relevant subtasks simultaneously (e.g., (Toutanova *et al.*, 2005), (Poon and Domingos, 2007), (Singh *et al.*, 2009)). Joint learning aims to handle multiple hypotheses and uncertainty information and predict many variables at once such that subtasks can aid each other to boost the performance, and thus usually leads to complex model structure. However, it is typically intractable to run a joint model and they sometimes can hurt the performance, since they

Coling 2008: Poster Volume, pages 1399–1407,
Beijing, August 2010

increase the number of paths to propagate errors. Due to these difficulties, research on building joint approaches is still in the beginning stage.

A significant amount of recent work has shown the power of discriminatively-trained probabilistic graphical models for NLP tasks (Lafferty *et al.*, 2001; Sutton and McCallum, 2007; Wainwright and Jordan, 2008). The superiority of graphical model is its ability to represent a large number of random variables as a family of probability distributions that factorize according to an underlying graph, and capture complex dependencies between variables. And this progress has begun to make the joint learning approach possible.

In this paper we study and formally define the joint problem of entity identification and relation extraction from encyclopedia text, and we propose a joint paradigm in a single coherent framework to perform both subtasks simultaneously. This framework is based on undirected probabilistic graphical models with arbitrary graphical structure. We show how the parameters in this model can be estimated efficiently. More importantly, we propose a new inference method — collective iterative classification (CIC), to find the maximum a posteriori (MAP) assignments for both entities and relations. We perform extensive experiments on real-world data from Wikipedia for this task, and substantial gains are obtained over state-of-the-art probabilistic pipeline and joint models, illustrating the promise of our approach.

2 Problem Formulation

2.1 Problem Description

This problem involves identifying entities and discovering semantic relationships between entity pairs from English encyclopedic articles. The basic document is an article, which mainly defines and describes an entity (known as *principal entity*). This document mentions some other entities as *secondary entities* related to the principal entity. Clearly, our task consists of two subtasks — first, for entity identification, we need to recognize the secondary entities (both the boundaries and types of them) in the document [1]. Second, after all the secondary entities are identified, our goal for relation extraction is to predict what relation, if any, each secondary entity has to the principal entity. We assume that there is no relationship between any two secondary entities in one document.

As an illustrative example, Figure 1 shows the task of entity identification and relationship extraction from encyclopedic documents. Here, *Abraham Lincoln* is the principal entity. Our task consists of assigning a set of pre-defined entity types (e.g., PER, DATE, YEAR, and ORG) to segmentations in encyclopedic documents and assigning a set of pre-defined relations (e.g., birth_day, birth_year, and member_of) for each identified secondary entity to the principal entity.

2.2 Problem Formulation

Let $\mathbf{x}$ be an observation sequence of tokens in encyclopedic text and $\mathbf{x} = \{x_1, \cdots, x_N\}$. Let s_p be the principal entity (we assume that it is known or can be easily recognized), and let $\mathbf{s} = \{s_1, \cdots, s_L\}$ be a segmentation assignment of observation sequence $\mathbf{x}$. Each segment s_i is a triple $s_i = \{\alpha_i, \beta_i, y_i\}$, where α_i is a start position, β_i is an end position, and y_i is the label assigned to all tokens of this segment. The segment s_i satisfies $0 \leq \alpha_i < \beta_i \leq |\mathbf{x}|$ and $\alpha_{i+1} = \beta_i + 1$. Let r_{pn} be the relation assignment between principal entity s_p and secondary entity candidate s_n from the segmentation $\mathbf{s}$, and $\mathbf{r}$ be the set of relation assignments for sequence $\mathbf{x}$.

Let $\mathbf{y} = \{\mathbf{r}, \mathbf{s}\}$ be the pair of segmentation $\mathbf{s}$ and segment relations $\mathbf{r}$ for an observation sequence $\mathbf{x}$. A valid assignment $\mathbf{y}$ must satisfy the condition that the assignments of the segments and the assignments of the relations of segments are maximized simultaneously. We now formally define this joint optimization problem as follows:

Definition 1 ***(Joint Optimization of Entity Identification and Relation Extraction)****: Given an observation sequence* $\boldsymbol{x}$*, the goal of joint optimization of entity identification and relation extraction is to find the assignment* $\boldsymbol{y}^* = \{\boldsymbol{r}^*, \boldsymbol{s}^*\}$ *that has the maximum a posteriori (MAP) probability*

$$\boldsymbol{y}^* = \arg\max_{\boldsymbol{y}} P(\boldsymbol{y}|\boldsymbol{x}), \quad (1)$$

[1] Since the topic/title of an article usually defines a principal entity (e.g., a famous person) and it is easy to identify, in this paper we only focus on secondary entity identification.

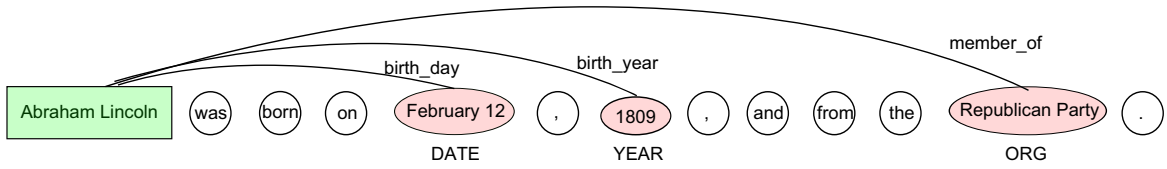


Figure 1: An example of entity identification and relation extraction excerpted from our dataset. The secondary entities are in pink color and labeled. The semantic relation of each secondary entity to the principal entity *Abraham Lincoln* (in green color and we assume that it is known or can be easily recognized) is also shown.

where $\mathbf{r}^*$ *and* $\mathbf{s}^*$ *denote the most likely relation assignment and segmentation assignment, respectively.*

Note that this problem is usually very challenging and offers new opportunities for information extraction, since complex dependencies between segmentations and relations should be exploited.

3 Our Proposed Model

3.1 Preliminaries

Conditional random fields (CRFs) (Lafferty *et al.*, 2001) are undirected graphical models trained to maximize the conditional probability of the desired outputs given the corresponding inputs. Let $\mathcal{G}$ be a factor graph (Kschischang *et al.*, 2001) defining a probability distribution over a set of output variables $\mathbf{o}$ conditioned on observation sequences $\mathbf{x}$. $C = \{\Phi_c(\mathbf{o}_c, \mathbf{x}_c)\}$ is a set of factors in $\mathcal{G}$, then the probability distribution over $\mathcal{G}$ can be written as

$$P(\mathbf{o}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{c \in C} \Phi_c(\mathbf{o}_c, \mathbf{x}_c) \qquad (2)$$

where Φ_c is a potential function and $Z(\mathbf{x}) = \sum_{\mathbf{o}} \prod_{c \in C} \Phi_c(\mathbf{o}_c, \mathbf{x}_c)$ is a normalization factor. We assume the potentials factorize according to a set of features $\{f_k(\mathbf{o}_c, \mathbf{x}_c)\}$ as $\Phi_c(\mathbf{o}_c, \mathbf{x}_c) = \exp(\sum_k \theta_k f_k(\mathbf{o}_c, \mathbf{x}_c))$ so that the family of distributions is an exponential family. The model parameters are a set of real-valued weights $\Theta = \{\theta_k\}$, one weight for each feature. Practical models rely extensively on parameter tying to use the same parameters for several factors.

However, the traditional fashion of CRFs can only deal with single task, they lack the capability to represent more complex interaction between multiple subtasks. In the following we will describe our joint model in detail for this problem.

3.2 A Joint Model for Entity Identification and Relation Extraction

Following the notations in Section 2.2, let L and M be the number of segments and number of relations for sequence $\mathbf{x}$, respectively. We define a joint conditional distribution for segmentation $\mathbf{s}$ in observation sequence $\mathbf{x}$ and segment relation $\mathbf{r}$ in undirected, probabilistic graphical models. The nature of our modeling enables us to partition the factors C of $\mathcal{G}$ into three groups $\{C_S, C_R, C_\nabla\}$=$\{\{\phi^S\}, \{\phi^R\}, \{\phi^\nabla\}\}$, namely the segmentation potential ϕ^S, the relation potential ϕ^R, and the segmentation-relation joint potential ϕ^∇, and each potential is a clique template whose parameters are tied. The potential function $\phi^S(i, \mathbf{s}, \mathbf{x})$ models segmentation $\mathbf{s}$ in $\mathbf{x}$, the potential function $\phi^R(r_{pm}, r_{pn}, \mathbf{r})$ $(m \neq n)$ represent dependencies (e.g., long-distance dependencies, relation transitivity, etc) between any two relations in the relation set $\mathbf{r}$, where r_{pm} is the relation assignment between the principal entity s_p and the secondary entity candidate s_m from $\mathbf{s}$, and similarly for r_{pn}. And the joint potential $\phi^\nabla(s_p, s_j, \mathbf{r})$ captures rich and complex interactions between segmentation $\mathbf{s}$ for secondary entity identification and relation $\mathbf{r}$ between each secondary entity candidate s_j to the principal entity s_p. According to the celebrated Hammersley-Clifford theorem (Besag, 1974), the joint conditional distribution $P(\mathbf{y}|\mathbf{x}) = P(\{\mathbf{r}, \mathbf{s}\}|\mathbf{x})$ is factorized as a product of potential functions over cliques in the graph $\mathcal{G}$ as the form of an exponential family:

$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \Big(\prod_{C_S} \phi^S(i, \mathbf{s}, \mathbf{x}) \Big) \Big(\prod_{C_R} \phi^R(r_{pm}, r_{pn}, \mathbf{r}) \Big) \Big(\prod_{C_\nabla} \phi^\nabla(s_p, s_j, \mathbf{r}) \Big) \qquad (3)$$

where $Z(\mathbf{x}) = \sum_{\mathbf{y}} \prod_{C_S} \phi^S(i, \mathbf{s}, \mathbf{x}) \prod_{C_R} \phi^R(r_{pm}, r_{pn}, \mathbf{r}) \prod_{C_\nabla} \phi^\nabla(s_p, s_j, \mathbf{r})$ is the normalization factor of the joint model.

We assume the potential functions ϕ^S, ϕ^R and ϕ^∇ factorize according to a set of features and a corresponding set of real-valued weights. More specifically, $\phi^S(i, \mathbf{s}, \mathbf{x}) = \exp(\sum_{i=1}^{|\mathbf{s}|} \sum_{k=1}^{K} \lambda_k g_k(i, \mathbf{s}, \mathbf{x}))$. To effectively capture properties of segmentation, we relax the first-order Markov assumption to semi-Markov such that each segment feature function $g_k(\cdot)$ depends on the current segment s_i, the previous segment s_{i-1}, and the whole observation sequence $\mathbf{x}$, that is, $g_k(i, \mathbf{s}, \mathbf{x}) = g_k(s_{i-1}, s_i, \mathbf{x}) = g_k(y_{i-1}, y_i, \alpha_i, \beta_i, \mathbf{x})$. And transitions within a segment can be non-Markovian.

Similarly, the potential $\phi^R(r_{pm}, r_{pn}, \mathbf{r}) = \exp(\sum_{m,n}^{M} \sum_{w=1}^{W} \mu_w q_w(r_{pm}, r_{pn}, \mathbf{r}))$ and $\phi^\nabla(s_p, s_j, \mathbf{r}) = \exp(\sum_{j=1}^{L} \sum_{t=1}^{T} \nu_t h_t(s_p, s_j, \mathbf{r}))$, where W and T are number of feature functions, $q_w(\cdot)$ and $h_t(\cdot)$ are feature functions, μ_w and ν_t are corresponding weights for them. The potential $\phi^R(r_{pm}, r_{pn}, \mathbf{r})$ allows long-range dependency representation between different relations r_{pm} and r_{pn}. For example, if the same secondary entity is mentioned more than once in an observation sequence, all mentions probably have the same relation to the principal entity. Using potential $\phi^R(r_{pm}, r_{pn}, \mathbf{r})$, evidences for the same entity segments to the principal entity are shared among all their occurrences within the document. The joint factor $\phi^\nabla(s_p, s_j, \mathbf{r})$ exploits tight dependencies between segmentations and relations. For example, if a segment is labeled as a *location* and the principal entity is *person*, the semantic relation between them can be *birth_place* or *visited*, but cannot be *employment*. Such dependencies are essential and modeling them often leads to improved performance. In summary, the probability distribution of the joint model can be rewritten as:

$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp\left\{ \sum_{i=1}^{|\mathbf{s}|} \sum_{k=1}^{K} \lambda_k g_k(i, \mathbf{s}, \mathbf{x}) + \sum_{m,n}^{M} \sum_{w=1}^{W} \mu_w q_w(r_{pm}, r_{pn}, \mathbf{r}) + \sum_{j=1}^{L} \sum_{t=1}^{T} \nu_t h_t(s_p, s_j, \mathbf{r}) \right\} \quad (4)$$

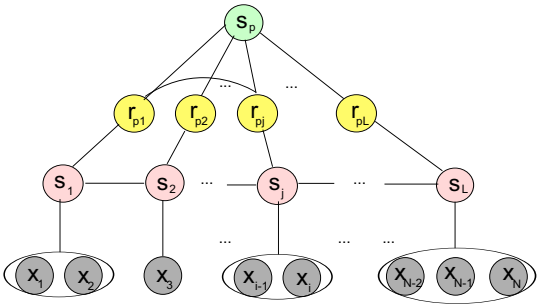


Figure 2: Graphical representation of the probabilistic joint model. The gray nodes represent sequence tokens $\{x_1, \cdots, x_N\}$. Each ellipse represents a segment consisting of several consecutive sequence tokens. The pink nodes represent segmentation assignment $\{s_1, \cdots, s_L\}$ of sequence. The yellow nodes represent relation assignment $\{r_{p1}, \cdots, r_{pL}\}$ between the principal entity s_p (in green color) and secondary entity segments.

As illustrated in Figure 2, our model consists of three sub-structures: a semi-Markov chain on the segmentations $\mathbf{s}$ conditioned on the observation sequences $\mathbf{x}$, represented by ϕ^S; potential ϕ^R measuring dependencies between different relations r_{pm} and r_{pn}; and a fully-connected graph on the principal entity s_p and each segment s_j for their relations, represented by ϕ^∇.

While several special cases of CRFs are of particular interest, and we emphasize on the differences and advantages of our model against others. Linear-chain CRFs (Lafferty *et al.*, 2001) can only perform single sequence labeling, they lack the ability to capture long-distance dependency and represent complex interactions between multiple subtasks. Skip-chain CRFs (Sutton and Mccallum, 2004) introduce skip edges to model long-distance dependencies to handle the label consistency problem in single sequence labeling and extraction. 2D CRFs (Zhu *et al.*, 2005) are two-dimensional conditional random fields incorporating the two-dimensional neighborhood dependencies in Web pages, and the graphical representation of this model is a 2D grid. Hierarchical CRFs (Liao *et al.*, 2007) are a class of CRFs with hierarchical tree structure. Our probabilistic model for joint entity identification and relation extraction has distinct graphical structure from 2D and hierarchical CRFs. And this modeling has sev-

eral advantages over previous probabilistic graphical models by using semi-Markov chains for efficient segmentation and labeling, by representing long-range dependencies between relations, and by capturing rich and complex interactions between relevant subtasks to exploit mutual benefits.

4 Learning the Parameters

Given independent and identically distributed (IID) training data $\mathcal{D} = \{\mathbf{x}^i, \mathbf{y}^i\}_{i=1}^{\mathbb{N}}$, where $\mathbf{x}^i$ is the i-th sequence instance, $\mathbf{y}^i = \{\mathbf{r}^i, \mathbf{s}^i\}$ is the corresponding segmentation and relation assignments. The objective of learning is to estimate $\Lambda = \{\lambda_k, \mu_w, \nu_t\}$ which is the vector of model's parameters. Under the IID assumption, we ignore the summation operator $\sum_{i=1}^{\mathbb{N}}$ in the log-likelihood during the following derivations. To reduce over-fitting, we use regularization and a common choice is a spherical Gaussian prior with zero mean and covariance $\sigma^2 I$. Then the regularized log-likelihood function $\mathcal{L}$ for the data is

$$\mathcal{L} = \log\big[\Phi(\mathbf{r}, \mathbf{s}, \mathbf{x})\big] - \log\big[Z(\mathbf{x})\big] - \sum_{k=1}^{K} \frac{\lambda_k^2}{2\sigma_\lambda^2} - \sum_{w=1}^{W} \frac{\mu_w^2}{2\sigma_\mu^2} - \sum_{t=1}^{T} \frac{\nu_t^2}{2\sigma_\nu^2} \quad (5)$$

where $\Phi(\mathbf{r}, \mathbf{s}, \mathbf{x}) = \exp\{\sum_{i=1}^{|\mathbf{s}|} \sum_{k=1}^{K} \lambda_k g_k(i, \mathbf{s}, \mathbf{x}) + \sum_{m,n}^{M} \sum_{w=1}^{W} \mu_w q_w(r_{pm}, r_{pn}, \mathbf{r}) + \sum_{j=1}^{L} \sum_{t=1}^{T} \nu_t h_t(s_p, s_j, \mathbf{r})\}$, $Z(\mathbf{x}) = \sum_{\mathbf{y}} \prod \Phi(\mathbf{r}, \mathbf{s}, \mathbf{x})$, and $1/2\sigma_\lambda^2$, $1/2\sigma_\mu^2$, $1/2\sigma_\nu^2$ are regularization parameters.

Taking derivatives of the function $\mathcal{L}$ over the parameter λ_k yields:

$$\frac{\partial \mathcal{L}}{\partial \lambda_k} = \sum_{i=1}^{|\mathbf{s}|} g_k(i, \mathbf{s}, \mathbf{x}) - \sum_{i=1}^{|\mathbf{s}|} g_k(i, \mathbf{s}, \mathbf{x}) P(\mathbf{y}|\mathbf{x}) - \sum_{k=1}^{K} \frac{\lambda_k}{\sigma_\lambda^2} \quad (6)$$

Similarly, the partial derivatives of the log-likelihood with respect to parameters μ_w and nu_t are as follows:

$$\frac{\partial \mathcal{L}}{\partial \mu_w} = \sum_{m,n}^{M} q_w(r_{pm}, r_{pn}, \mathbf{r}) - \sum_{m,n}^{M} q_w(r_{pm}, r_{pn}, \mathbf{r}) \times P(\mathbf{y}|\mathbf{x}) - \sum_{w=1}^{W} \frac{\mu_w}{\sigma_\mu^2} \quad (7)$$

$$\frac{\partial \mathcal{L}}{\partial \nu_t} = \sum_{j=1}^{L} h_t(s_p, s_j, \mathbf{r}) - \sum_{j=1}^{L} h_t(s_p, s_j, \mathbf{r}) P(\mathbf{y}|\mathbf{x}) - \sum_{t=1}^{T} \frac{\nu_t}{\sigma_\nu^2} \quad (8)$$

The function $\mathcal{L}$ is concave, and can be efficiently maximized by standard techniques such as stochastic gradient and limited memory quasi-Newton (L-BFGS) algorithms. The parameters λ_k μ_w and ν_t are optimized iteratively until converge.

5 Finding the Most Likely Assignments

The objective of inference is to find $\mathbf{y}^* = \{\mathbf{r}^*, \mathbf{s}^*\} = \arg\max_{\{\mathbf{r},\mathbf{s}\}} P(\mathbf{r}, \mathbf{s}|\mathbf{x})$ such that both $\mathbf{s}^*$ and $\mathbf{r}^*$ are optimized simultaneously. Unfortunately, exact inference to this problem is generally prohibitive, since it requires enumerating all possible segmentation and corresponding relation assignments. Consequently, approximate inference becomes an alternative.

We propose a new algorithm: collective iterative classification (CIC) to perform approximate inference to find the maximum a posteriori (MAP) segmentation and relation assignments of our model in an iterative fashion. The basic idea of CIC is to decode every target hidden variable based on the assigning labels of its sampled variables, where the labels might be dynamically updated throughout the iterative process. Collective classification refers to the classification of relational objects described as nodes in a graphical structure, as in our model.

The CIC algorithm performs inference in two steps, as shown in Algorithm 1. The first step, `bootstrapping`, predicts an initial labeling assignment for a unlabeled sequence $\mathbf{x}_i$, given the trained model $P(\mathbf{y}|\mathbf{x})$. The second step is the `iterative classification` process which re-estimates the labeling assignment of $\mathbf{x}_i$ several times, picking them in a sample set $\mathcal{S}$ based on initial assignment for $\mathbf{x}_i$. Here we exploit the sampling technique (Andrieu *et al.*, 2003). The advantages of sampling are summarized as follows. Sampling stochastically enables us to generate a wide range of inference situations, and the samples are likely to be in high probability areas, increasing our chances of finding the max-

imum, thus leading to more robust and accurate performance. The CIC algorithm may converge if none of the labeling assignments change during an iteration or a given number of iterations is reached.

Noticeably, this inference algorithm is also used to efficiently compute the marginal probability $P(\mathbf{y}|\mathbf{x})$ during parameter estimation (the normalization constant $Z(\mathbf{x})$ can also be calculated via approximation techniques). As can be seen, this algorithm is simple to design, efficient and scales well *w.r.t.* the size of data.

Algorithm 1: Collective Iterative Classification Inference

Input: A unlabeled sequence $\mathbf{x}_i$ and a trained model $P(\mathbf{y}|\mathbf{x})$
Output: The set of predicted assignment $y_i = \{r_i, s_i\}$
`// Bootstrapping`
foreach $y_i \in \mathcal{Y}$ **do**
  $\overline{y_i} \leftarrow \arg\max_{y_i} P(y_i|x_i)$;
end
`// Iterative Classification`
repeat
  Generate a sample set $\mathcal{S}$ based on initial label assignment $\overline{y_i}$ for sequence $\mathbf{x}_i$;
  foreach $\mathbb{s}_i \in \mathcal{S}$ **do**
    Assign new label assignment to sample $\mathbb{s}_i$;
  end
until *all labels have stabilized or a threshold number of iterations have elapsed* ;
return $y_i = \{r_i, s_i\}$

6 Experiments

6.1 Data

Our data comes from Wikipedia[2], the world's largest free online encyclopedia. This dataset consists of 1127 paragraphs from 441 pages from the online encyclopedia Wikipedia. We labeled 7740 entities into 8 categories, yielding 1243 *person*, 1085 *location*, 875 *organization*, 641 *date*, 1495 *year*, 38 *time*, 59 *number*, and 2304 *miscellaneous* names. This dataset also contains 4701 relation instances and 53 labeled relation types. The 10 most frequent relation types are *job_title*, *visited*, *birth_place*, *associate*, *birth_year*, *member_of*, *birth_day*, *opus*, *death_year*, and *death_day*. Note that this compound IE task involving entity identification and relation extraction is very challenging, and modeling tight interactions between entities and their relations is highly attractive.

6.2 Feature Set

Accurate entities enable features that are naturally expected to be useful to boost relation extraction. And a wide range of rich, overlapping features can be exploited in our model. These features include contextual features, part-of-speech (POS) tags, morphological features, entity-level dictionary features, clue word features. Feature conjunctions are also used. In leveraging relation extraction to improve entity identification, we use a combination of syntactic, entity, keyword, semantic, and Wikipedia characteristic features. More importantly, our model can incorporate multiple mention features $q_w(\cdot)$, which are used to collect evidences from other occurrences of the same secondary entities for consistent segmentation and relation labeling to the principal entity. The features $h_t(\cdot)$ capture deep dependencies between segmentations and relations, and they are natural and useful to enhance the performance.

[2]http://www.wikipedia.org/

6.3 Methodology

We perform four-fold cross-validation on this dataset, and take the average performance. For performance evaluation, we use the standard measures of Precision (P), Recall (R), and F-measure (the harmonic mean of P and R: $\frac{2PR}{P+R}$) for both entity identification and relation extraction. We conduct holdout methodology for parameter tuning and optimization of our model. We compare our approach with a series of linear-chain CRFs: **CRF+CRF** and a joint model **DCRF** (Sutton *et al.*, 2007): dynamic probabilistic models combined with factored approach to multiple sequence labeling. **CRF+CRF** perform entity identification and relation extraction separately. Relation extraction is viewed as a sequence labeling problem in the second CRF. All these models exploit standard parameter learning and inference algorithms

Table 1: Comparative performance of our model, CRF+CRF, and DCRF models for entity identification.

Entities	CRF+CRF			DCRF			Our model		
	P	R	F_1	P	R	F_1	P	R	F_1
person	75.33	83.22	79.08	75.96	83.82	79.70	82.91	84.26	**83.58**
location	77.03	69.45	73.04	77.68	70.13	73.71	82.94	80.52	**81.71**
organization	53.78	47.76	50.59	54.55	46.98	50.48	61.63	62.61	**62.12**
date	98.54	97.53	**98.03**	97.98	95.22	96.58	98.90	96.24	97.55
year	97.14	99.10	98.11	98.12	99.09	**98.60**	97.36	99.55	98.44
time	60.00	20.33	30.37	50.00	25.33	33.63	100.0	25.00	**40.00**
number	98.88	60.33	74.94	100.0	66.00	**79.52**	100.0	65.52	79.17
miscellaneous	77.42	80.56	78.96	79.81	83.14	81.44	82.69	85.16	**83.91**
Overall	89.55	88.70	89.12	90.98	90.37	90.67	93.35	93.37	**93.36**

in our experiments. To avoid over-fitting, penalization techniques on likelihood are performed. We also use the same set of features for all these models.

6.4 Experimental Results

Table 1 shows the performance of entity identification and Table 2 shows the overall performance of relation extraction [3], respectively. Our model substantially outperforms all baseline models on the overall F-measure for entity identification, resulting in an relative error reduction of up to 38.97% and 28.83% compared to **CRF+CRF** and **DCRF**, respectively. For relation extraction, the improvements on the F-measure over **CRF+CRF** and **DCRF** are 4.68% and 3.75%. McNemar's paired tests show that all improvements of our model over baseline models are statistically significant. These results demonstrate the merits of our approach by capturing tight interactions between entities and relations to explore mutual benefits. The pipeline model **CRF+CRF** performs entity identification and relation extraction independently, and suffers from problems such as error accumulation. For example, **CRF+CRF** cannot extract the *member_of* relation between the secondary entity *Republican* and the principal entity *George W. Bush*, since the organization name *Republican* is incorrectly labeled as a *miscellaneous*. By modeling interactions between two subtasks, enhanced performance is achieved, as illustrated by **DCRF**. Unfortunately, training a **DCRF** model with unobserved nodes (hidden variables) makes this approach difficult to optimize, as we will show below.

[3]Due to space limitation, we only present the overall performance and omit the performance for 53 relation types.

Table 2: Comparative performance of our model, CRF+CRF, and DCRF models for relation extraction.

Model	Precision	Recall	F-measure
CRF+CRF	70.40	57.85	63.51
DCRF	69.30	60.22	64.44
Our model	72.57	64.30	**68.19**

The efficiency of different models is summarized in Table 3. Compared to the pipeline model **CRF+CRF**, the learning time of our model is only a small constant factor slower. Notably, our model is over orders of magnitude (approximately 15.7 times) faster than the joint model **DCRF**. The **DCRF** model uses loopy belief propagation (LBP) for approximate learning and inference. When the graph has large tree-width as in our case, the LBP algorithm in **DCRF** is inefficient, and is slow to converge. Using L-BFGS and the CIC approximate inference algorithms, both learning and decoding can be carried out efficiently.

Table 3: Efficiency comparison of different models on learning time (sec.) and inference time (sec.).

Model	Learning time	Inference time
CRF+CRF	2822.55	6.20
DCRF	105993.00	127.50
Our model	6733.69	62.75

Table 4 compares our CIC inference with two state-of-the-art inference approaches: Gibbs sampling (GS) (Geman and Geman, 1984) and the iterative classification algorithm (ICA) (Neville and Jensen, 2000) for our model. The CIC inference is shown empirically to help improve classi-

Table 4: Comparative performance of different inference algorithms for our model on entity identification and relation extraction.

Entity	Precision	Recall	F-measure
GS	92.45	92.15	92.30
ICA	92.19	91.98	92.08
CIC	93.35	93.37	**93.36**
Relation	Precision	Recall	F-measure
GS	71.22	63.29	67.02
ICA	71.58	63.68	67.40
CIC	72.57	64.30	**68.19**

fication accuracy and robustness over these two algorithms. When probability distributions are very complex or even unknown, the GS algorithm cannot be applied. ICA iteratively infers the states of variables given the current predicted labeling assignments of neighboring variables as observed information. Prediction errors on labels may then propagate during the iterations and the algorithm will then have difficulties to generalize correctly.

We mention some recently published results related to Wikipedia datasets (Note that it is difficult to compare with them strictly, since these results can be based on different experimental settings). Culotta *et al.* (2006) used a data set with a 70/30 split for training/testing and Nguyen *et al.* (2007) used 5930 articles for training and 45 for testing, to perform relatione extraction from Wikipedia. And the obtained F-measures were 67.91 and 37.76, respectively. Yu *et al.* (2009) proposed an integrated approach incorporating probabilistic graphical models with first-order logic to perform relation extraction from encyclopedia articles, with a F-measure of 65.66. All these systems assume that the golden-standard entities are already known and they only perform relation extraction. However, such assumption is not valid in practice. Notably, our approach deals with a fairly more challenging problem involving both entity identification and relation extraction, and it is more applicable to real-world IE tasks.

7 Related Work

A number of previous researchers have taken steps toward joint models in NLP and information extraction, and we mention some recently proposed, closely related approaches here. Roth and Yih (2007) considered multiple constraints between variables from tasks such as named entities and relations, and developed a integer linear programming formulation to seek an optimal global assignment to these variables. Zhang and Clark (2008) employed the generalized perceptron algorithm to train a statistical model for joint segmentation and POS tagging, and applied multiple-beam search algorithm for fast decoding. Toutanova *et al.* (2008) presented a model capturing the linguistic intuition that a semantic argument frame is a joint structure, with strong dependencies among the arguments. Finkel and Manning (2009) proposed a discriminative feature-based constituency parser for joint named entity recognition and parsing. And Dahlmeier *et al.* (2009) proposed a joint model for word sense disambiguation of prepositions and semantic role labeling of prepositional phrases. However, most of the mentioned approaches are task-specific (e.g., (Toutanova *et al.*, 2008) for semantic role labeling, and (Finkel and Manning, 2009) for parsing and NER), and they can hardly be applicable to other NLP tasks. Since we capture rich and complex dependencies between subtasks via potential functions in probabilistic graphical models, our approach is general and can be easily applied to a variety of NLP and IE tasks.

8 Conclusion and Future Work

In this paper, we investigate the compound IE task of identifying entities and extracting relations between entities in encyclopedia text. And we propose a unified framework based on undirected, conditionally-trained probabilistic graphical models to perform all relevant subtasks jointly. More importantly, we propose a new algorithm: CIC, to enable approximate inference to find the MAP assignments for both segmentations and relations. As we shown, our modeling offers several advantages over previous models and provides a natural formalism for this compound task. Experimental study exhibits that our model significantly outperforms state-of-the-art models while also running much faster than the joint models. In addition, the superiority of the CIC algorithm is also discussed and compared. We plan to improve the scalability of our approach and apply it to other real-world problems in the future.

References

Christophe Andrieu, Nando de Freitas, Arnaud Doucet, and Michael I. Jordan. An introduction to MCMC for machine learning. *Machine Learning*, 50(1):5–43, 2003.

Julian Besag. Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society*, 36:192–236, 1974.

Aron Culotta, Andrew McCallum, and Jonathan Betz. Integrating probabilistic extraction models and data mining to discover relations and patterns in text. In *Proceedings of HLT/NAACL-06*, pages 296–303, New York, 2006.

Daniel Dahlmeier, Hwee Tou Ng, and Tanja Schultz. Joint learning of preposition senses and semantic roles of prepositional phrases. In *Proceedings of EMNLP-09*, pages 450–458, Singapore, 2009.

Jenny Rose Finkel and Christopher D. Manning. Joint parsing and named entity recognition. In *Proceedings of HLT/NAACL-09*, pages 326–334, Boulder, Colorado, 2009.

Jenny Rose Finkel, Christopher D. Manning, and Andrew Y. Ng. Solving the problem of cascading errors: Approximate Bayesian inference for linguistic annotation pipelines. In *Proceedings of EMNLP-06*, pages 618–626, Sydney, Australia, 2006.

Stuart Geman and Donald Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transitions on Pattern Analysis and Machine Intelligence*, 6:721–741, 1984.

Kristy Hollingshead and Brian Roark. Pipeline iteration. In *Proceedings of ACL-07*, pages 952–959, Prague, Czech Republic, 2007.

Frank R. Kschischang, Brendan J. Frey, and Hans-Andrea Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47:498–519, 2001.

John Lafferty, Andrew McCallum, and Fernando Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of ICML-01*, pages 282–289, 2001.

Lin Liao, Dieter Fox, and Henry Kautz. Extracting places and activities from GPS traces using hierarchical conditional random fields. *International Journal of Robotics Research*, 26:119–134, 2007.

Jennifer Neville and David Jensen. Iterative classification in relational data. In *Proceedings of the AAAI-2000 Workshop on Learning Statistical Models from Relational Data*, pages 42–49, 2000.

Dat P. T. Nguyen, Yutaka Matsuo, and Mitsuru Ishizuka. Relation extraction from Wikipedia using subtree mining. In *Proceedings of AAAI-07*, pages 1414–1420, Vancouver, British Columbia, Canada, 2007.

Hoifung Poon and Pedro Domingos. Joint inference in information extraction. In *Proceedings of AAAI-07*, pages 913–918, Vancouver, British Columbia, Canada, 2007.

Dan Roth and Wentau Yih. Global inference for entity and relation identification via a linear programming formulation. In Lise Getoor and Ben Taskar, editors, *Introduction to Statistical Relational Learning*. MIT Press, 2007.

Sameer Singh, Karl Schultz, and Andrew Mccallum. Bidirectional joint inference for entity resolution and segmentation using imperatively-defined factor graphs. In *Proceedings of ECML/PKDD-09*, pages 414–429, Bled, Slovenia, 2009.

Charles Sutton and Andrew Mccallum. Collective segmentation and labeling of distant entities in information extraction. In *Proceedings of ICML Workshop on Statistical Relational Learning and Its Connections to Other Fields*, 2004.

Charles Sutton and Andrew McCallum. An introduction to conditional random fields for relational learning. In Lise Getoor and Ben Taskar, editors, *Introduction to Statistical Relational Learning*. MIT Press, 2007.

Charles Sutton, Andrew McCallum, and Khashayar Rohanimanesh. Dynamic conditional random fields: Factorized probabilistic models for labeling and segmenting sequence data. *Journal of Machine Learning Research*, 8:693–723, 2007.

Kristina Toutanova, Aria Haghighi, and Christopher D. Manning. Joint learning improves semantic role labeling. In *Proceedings of ACL-05*, pages 589–596, 2005.

Kristina Toutanova, Aria Haghighi, and Christopher D. Manning. A global joint model for semantic role labeling. *Computational Linguistics*, 34:161–191, 2008.

Kristina Toutanova. *Effective statistical models for syntactic and semantic disambiguation*. PhD thesis, Stanford University, 2005.

Martin J. Wainwright and Michael I. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1:1–305, 2008.

Xiaofeng Yu, Wai Lam, and Shing-Kit Chan. A framework based on graphical models with logic for chinese named entity recognition. In *Proceedings of the Third International Joint Conference on Natural Language Processing (IJCNLP-08)*, pages 335–342, Hyderabad, India, 2008.

Xiaofeng Yu, Wai Lam, and Bo Chen. An integrated discriminative probabilistic approach to information extraction. In *Proceedings of CIKM-09*, pages 325–334, Hong Kong, China, 2009.

Xiaofeng Yu. Chinese named entity recognition with cascaded hybrid model. In *Proceedings of HLT/NAACL-07*, pages 197–200, Rochester, New York, 2007.

Yue Zhang and Stephen Clark. Joint word segmentation and POS tagging using a single perceptron. In *Proceedings of ACL-08*, pages 888–896, Ohio, USA, 2008.

Jun Zhu, Zaiqing Nie, Ji-Rong Wen, Bo Zhang, and Wei-Ying Ma. 2D conditional random fields for Web information extraction. In *Proceedings of ICML-05*, pages 1044–1051, Bonn, Germany, 2005.

Accelerated Training of Maximum Margin Markov Models for Sequence Labeling: A Case Study of NP Chunking*

Xiaofeng YU **Wai LAM**

Information Systems Laboratory
Department of Systems Engineering & Engineering Management
The Chinese University of Hong Kong
{xfyu,wlam}@se.cuhk.edu.hk

Abstract

We present the first known empirical results on sequence labeling based on maximum margin Markov networks (M^3N), which incorporate both kernel methods to efficiently deal with high-dimensional feature spaces, and probabilistic graphical models to capture correlations in structured data. We provide an efficient algorithm, the stochastic gradient descent (SGD), to speedup the training procedure of M^3N. Using official dataset for noun phrase (NP) chunking as a case study, the resulting optimizer converges to the same quality of solution over an order of magnitude faster than the structured sequential minimal optimization (structured SMO). Our model compares favorably with current state-of-the-art sequence labeling approaches. More importantly, our model can be easily applied to other sequence labeling tasks.

1 Introduction

The problem of annotating or labeling observation sequences arises in many applications across a variety of scientific disciplines, most prominently in natural language processing, speech recognition, information extraction, and bioinformatics. Recently, the predominant formalism for modeling and predicting label sequences has been based on discriminative graphical models and variants.

Among such models, maximum margin Markov networks (M^3N) and variants (Taskar *et al.* (2003); Taskar (2004); Taskar *et al.* (2005)) have recently gained popularity in the machine learning community. While the M^3N framework makes extensive use of many theoretical results available for Markov networks, it largely dispenses with the probabilistic interpretation. M^3N thus combines the advantages of both worlds, the possibility to have a concise model of the relationships present in the data via log-linear Markov networks over a set of label variables and the highly accurate predictions based on maximum margin estimation of the model parameters.

Traditionally, M^3N can be trained using the structured sequential minimal optimization (structured SMO), a coordinate descent method for solving quadratic programming (QP) problems (Taskar *et al.*, 2003). Clearly, however, the polynomial number of constraints in the QP problem associated with the M^3N can still be very large, making the structured SMO algorithm slow to converge over the training data. This currently limits the scalability and applicability of M^3N to large-scale real world problems.

Stochastic gradient methods (e.g., Lecun *et al.* (1998); Bottou (2004)), on the other hand, are online and scale sub-linearly with the amount of training data, making them very attractive for large-scale datasets. In stochastic (or online) gradient descent (SGD), the true gradient is approximated by the gradient of the cost function only evaluated on a single training example. The parameters are then adjusted by an amount proportional to this approximate gradient. Therefore, the parameters of the model are updated after each training example. For large-scale datasets, online gradient descent can be much faster than standard (or batch) gradient descent.

In this paper, we marry the above two techniques and show how SGD can be used to significantly accelerate the training of M^3N. And we

The work described in this paper is substantially supported by grants from the Research Grant Council of the Hong Kong Special Administrative Region, China (Project No: CUHK4128/07) and the Direct Grant of the Faculty of Engineering, CUHK (Project Codes: 2050442 and 2050476). This work is also affiliated with the Microsoft-CUHK Joint Laboratory for Human-centric Computing and Interface Technologies.

Coling 2008: Poster Volume, pages 1408–1416,
Beijing, August 2010

then apply our model to the well-established sequence labeling task: noun phrase (NP) chunking. Experimental results show the validity and effectiveness of our approach. We now summarize the primary contributions of this paper as follows:

- We exploit M^3N to NP chunking on the standard evaluation dataset, achieving favorable performance against recent top-performing systems. The M^3N framework allows arbitrary features of observation sequence, as well as the important benefits of kernels. To the best of our knowledge, this is the first known empirical study on NP chunking using M^3N in the NLP community.

- We provide the efficient SGD algorithm to accelerate the training procedure of M^3N, and experimental results show that it converges over an order of magnitude faster than the structured SMO without sacrificing performance.

- Our model is easily extendable to other sequence labeling tasks, such as part-of-speech tagging and named entity recognition. Based on the promising results on NP chunking, we believe that our model will significantly further the applicability of margin-based approaches to large-scale sequence labeling tasks.

2 Maximum Margin Markov Networks for Sequence Labeling

In sequence labeling, the output is a sequence of labels $\mathbf{y} = (y_1, \ldots, y_T)$ which corresponds to an observation sequence $\mathbf{x} = (x_1, \ldots, x_T)$. Suppose each individual label can take values from set Σ, then the problem can be considered as a multiclass classification problem with $|\Sigma|^T$ different classes.

In M^3N, a pairwise Markov network is defined as a graph $\mathcal{G} = (Y, E)$. Each edge $(i, j) \in E$ is associated with a potential function

$$\begin{aligned}\psi_{ij}(\mathbf{x}, y_i, y_j) &= \exp(\sum_{k=1}^{l} w_k \phi_k(\mathbf{x}, y_i, y_j)) \\ &= \exp(\mathbf{w}^\top \phi(\mathbf{x}, y_i, y_j)) \end{aligned} \quad (1)$$

where $\phi(\mathbf{x}, y_i, y_j)$ is a pairwise basis function. All edges in the graph denote the same type of interaction, so that we can define a feature map $\phi_k(\mathbf{x}, \mathbf{y}) = \sum_{(i,j)\in E} \phi_k(\mathbf{x}, y_i, y_j)$. The network encodes the following conditional probability distribution (Taskar *et al.*, 2003):

$$P(\mathbf{y}|\mathbf{x}) \propto \prod_{(i,j)\in E} \psi_{ij}(\mathbf{x}, y_i, y_j) = \exp(\mathbf{w}^\top \phi(\mathbf{x}, \mathbf{y})) \quad (2)$$

where $\phi(\mathbf{x}, \mathbf{y}) = [\phi_1 \phi_2 \ldots \phi_{|\Sigma|} \phi_{trans}]^\top$ is used to learn a weight vector $\mathbf{w}$. $\phi_k = \sum_{i=1}^{n} \phi_i(x) \mathcal{I}(y_i = k), \forall k \in \{1, 2, \ldots, |\Sigma|\}$ and $\phi_{trans} = [c_{11} c_{12} \ldots c_{TT}]^\top$ where c_{ij} is the number of observed transitions from the i^{th} alphabet to the j^{th} alphabet in Σ.

Similar to SVMs (Vapnik, 1995), M^3N tries to find a projection to maximize the margin γ. On the other hand, M^3N also attempts to minimize $\|\mathbf{w}\|$ to minimize the generalization error. Suppose $\Delta \mathbf{t}_{\mathbf{x}}(\mathbf{y}) = \sum_{i=1}^{n} \Delta \mathbf{t}_{\mathbf{x}}(y_i) = \sum_{i=1}^{n} I(y_i \neq (\mathbf{t}(\mathbf{x}))_i)$ where $\mathbf{t}((\mathbf{x}))_i$ is the true label of the i^{th} sequence x_i, and $\Delta \phi_{\mathbf{x}}(\mathbf{y}) = \phi(\mathbf{x}, \mathbf{t}(\mathbf{x})) - \phi(\mathbf{x}, \mathbf{y})$ where $\mathbf{t}(\mathbf{x})$ is the true label of the observation sequence $\mathbf{x}$. We can get a quadratic program (QP) using a standard transformation to eliminate γ as follows:

$$\begin{aligned} \min \quad & \frac{1}{2}\|\mathbf{w}\|^2; \\ \text{s.t.} \quad & \mathbf{w}^\top \Delta \phi_{\mathbf{x}}(\mathbf{y}) \geq \Delta \mathbf{t}_{\mathbf{x}}(\mathbf{y}), \forall \mathbf{x} \in S, \forall \mathbf{y} \in \Sigma. \end{aligned} \quad (3)$$

However, the sequence data is often not separable by the defined hyperplane. In such cases, we can introduce slack variables $\xi_{\mathbf{x}}$ which are guaranteed to be non-negative to allow some constraints. Thus the complete primal form of the optimization problem can be formulated by:

$$\begin{aligned} \min \quad & \frac{1}{2}\|\mathbf{w}\|^2 + \mathcal{C} \sum_{\mathbf{x}} \xi_{\mathbf{x}}; \\ \text{s.t.} \quad & \mathbf{w}^\top \Delta \phi_{\mathbf{x}}(\mathbf{y}) \geq \Delta \mathbf{t}_{\mathbf{x}}(\mathbf{y}) - \xi_{\mathbf{x}}, \forall \mathbf{x} \in S, \forall \mathbf{y} \in \Sigma. \end{aligned} \quad (4)$$

where $\mathcal{C}$ is called the capacity in the support vector literature and presents a way to trade-off the training error and margin size. One should note that the number of constraints is $\sum_{i=1}^{T} |\Sigma^i|$, an extremely large number. And the corresponding dual formu-

lation can be defined as:

$$\max \sum_{\mathbf{x},\mathbf{y}} \alpha_{\mathbf{x}}(\mathbf{y})\Delta\mathbf{t}_{\mathbf{x}}(\mathbf{y}) - \frac{1}{2}\|\sum_{\mathbf{x},\mathbf{y}} \alpha_{\mathbf{x}}(\mathbf{y})\Delta\phi_{\mathbf{x}}(\mathbf{y})\|^2;$$
$$\text{s.t.} \sum_{\mathbf{y}} \alpha_{\mathbf{x}}(\mathbf{y}) = \mathcal{C}, \forall\mathbf{x}; \alpha_{\mathbf{x}}(\mathbf{y}) \geq 0, \forall\mathbf{x},\mathbf{y}. \quad (5)$$

where $\alpha_{\mathbf{x}}(\mathbf{y})$ is a dual variable.

As well as loss functions, kernels might have substantial influence on the performance of a classification system. M^3N is capable of incorporating many different kinds of kernel functions to reduce computations in the high-dimensional feature space $\mathcal{H}$. This is sometimes referred to as the "kernel trick" (Schölkopf and Smola, 2002; Shawe-Taylor and Cristianini, 2004). A linear kernel can be defined as

$$\kappa((\mathbf{x},\mathbf{y}),(\mathbf{x}',\mathbf{y}')) = \langle\phi(\mathbf{x},\mathbf{y}),\phi(\mathbf{x}',\mathbf{y}')\rangle_{\mathcal{H}} \quad (6)$$

For a polynomial kernel,

$$\kappa((\mathbf{x},\mathbf{y}),(\mathbf{x}',\mathbf{y}')) = (s\cdot\langle\phi(\mathbf{x},\mathbf{y}),\phi(\mathbf{x}',\mathbf{y}')\rangle_{\mathcal{H}} + r)^d, \quad (7)$$

and for a neural kernel,

$$\kappa((\mathbf{x},\mathbf{y}),(\mathbf{x}',\mathbf{y}')) = \tanh(s\cdot\langle\phi(\mathbf{x},\mathbf{y}),\phi(\mathbf{x}',\mathbf{y}')\rangle_{\mathcal{H}} + r), \quad (8)$$

where s, d, and r are coefficients in kernel functions.

3 Stochastic Gradient Descent

For M^3N optimization, Taskar *et al.* (2003) has proposed a reparametrization of the dual variables to take advantage of the network structure of the labeling sequence problem. The dual QP is then solved using the structured sequential minimal optimization (structured SMO) analogous to the SMO used for SVMs (Platt, 1998). However, the resulting number of constraints in the QP make the structured SMO algorithm slow to converge, or even prohibitively expensive for large-scale real world problems. In this section we will present stochastic gradient descent (SGD) method, and show SGD can significantly speedup the training of M^3N.

3.1 Regularized Loss Minimization

Recall that for M^3N, the goal is to find a linear hypothesis $h_{\mathbf{w}}$ such that $h_{\mathbf{w}}(\mathbf{x}) = \arg\max_{\mathbf{y}\in\Sigma} \mathbf{w}^\top\phi(\mathbf{x},\mathbf{y})$. The parameters $\mathbf{w}$ are learned by minimizing a regularized loss

$$\mathcal{L}(\mathbf{w};\{(x_i,y_i)\}_{i=1}^T, C) = \sum_{i=1}^m \ell(\mathbf{w},x_i,y_i) + \frac{C}{2}\|\mathbf{w}\|^2. \quad (9)$$

The function ℓ measures the loss incurred in using $\mathbf{w}$ to predict the label of x_i. Following (Taskar *et al.*, 2003), $\ell(\mathbf{w},x_i,y_i)$ is a variant of the hinge loss, and can be defined as follows:

$$\ell(\mathbf{w},x_i,y_i) = \max_{\mathbf{y}\in\Sigma}[e(x_i,y_i,\mathbf{y}) - \mathbf{w}\cdot(\phi(x_i,y_i) - \phi(x_i,\mathbf{y}))], \quad (10)$$

where $e(x_i,y_i,\mathbf{y})$ is some non-negative measure of the error incurred in predicting y instead of y_i as the label of x_i. We assume that $e(x_i,y_i,\mathbf{y}) = 0$ for all i, so that no loss is incurred for correct prediction, and therefore $\ell(\mathbf{w},x_i,y_i)$ is always non-negative. This loss function corresponds to the M^3N approach, which explicitly penalizes training examples for which, for some $y \neq y_i$, $\mathbf{w}\cdot(\phi(x_i,y_i) - \phi(x_i,\mathbf{y})) < e(x_i,y_i,\mathbf{y})$. And the function $\mathcal{L}$ is convex in $\mathbf{w}$ for $\ell(\mathbf{w},x_i,y_i)$. Therefore, minimization of $\mathcal{L}$ can be re-cast as optimization of the following dual convex problem:

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \sum_i \max_{\mathbf{y}\in\Sigma}[e(x_i,y_i,\mathbf{y}) - \mathbf{w}\cdot(\phi(x_i,y_i) - \phi(x_i,\mathbf{y}))] + \frac{C}{2}\|\mathbf{w}\|^2. \quad (11)$$

3.2 The SGD Algorithm

To perform parameter estimation, we need to minimize $\mathcal{L}(\mathbf{w};\{(x_i,y_i)\}_{i=1}^T, C)$. For this purpose we compute its gradient $\mathbf{G}(\mathbf{w})$:

$$\mathbf{G}(\mathbf{w}) = \frac{\partial}{\partial\mathbf{w}}(\mathcal{L}(\mathbf{w};\{(x_i,y_i)\}_{i=1}^T, C)) = \frac{\partial}{\partial\mathbf{w}}(\sum_{i=1}^m \ell(\mathbf{w},x_i,y_i) + \frac{C}{2}\|\mathbf{w}\|^2) \quad (12)$$

In addition to the gradient, second-order methods based on Newton steps also require computation and inversion of the Hessian $\mathbf{H}(\mathbf{w})$. Taking

the gradient of Equation 12 *wrt.* $\mathbf{w}$ yields:

$$\mathbf{H}(\mathbf{w}) = \frac{\partial}{\partial \mathbf{w}}\mathbf{G}(\mathbf{w}) = \frac{\partial^2}{\partial \mathbf{w}^2}\mathcal{L} \tag{13}$$

Explicitly computing the full Hessian is time consuming. Instead we can make use of the differential

$$\mathrm{d}\mathbf{G}(\mathbf{w}) = \mathbf{H}(\mathbf{w})\mathrm{d}\mathbf{w} \tag{14}$$

to efficiently compute the product of the Hessian with a chosen vector $\mathbf{v} =: \mathrm{d}\mathbf{w}$ by forward-mode algorithmic differentiation (Pearlmutter, 1994). These Hessian-vector products can be computed along with the gradient at only 2-3 times the cost of the gradient computation alone. We denote $\mathbf{G}(\mathbf{w}) = \nabla_{\mathbf{w}}\mathcal{L}$, and each iteration of the SGD algorithm consists in drawing an example (x_i, y_i) at random and applying the parameter update rule (Robbins and Monroe, 1951):

$$\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t - \eta \cdot \nabla_{\mathbf{w}}\mathcal{L} \tag{15}$$

where η is the learning rate in the algorithm.

The SGD algorithm has been shown to be fast, reliable, and less prone to reach bad local minima. In this algorithm, the weights are updated after the presentation of each example, according to the gradient of the loss function (Lecun *et al.*, 1998). The convergence is very fast when the training examples are redundant since only a few examples are needed to perform. This algorithm can get a good estimation after considerably few iterations.

3.3 Choosing Learning Rate η

The learning rate η is crucial to the speed of SGD algorithm. Ideally, each parameter weight w_i should have its own learning rate η_i. Because of possible correlations between input variables, the learning rate of a unit should be inversely proportional to the square root of the number of inputs to the unit. If shared weights are used, the learning rate of a weight should be inversely proportional to the square root of the number of connection sharing that weight.

For one-dimensional sequence labeling task, the optimal learning rate yields the fastest convergence in the direction of highest curvature is (Bottou, 2004):

$$\eta_{\text{opt}} = (\frac{\partial^2 \mathcal{L}}{\partial \mathbf{w}^2})^{-1} = (\mathbf{H}(\mathbf{w}))^{-1}, \tag{16}$$

and the maximum learning rate is $\eta_{\text{max}} = 2\eta_{\text{opt}}$.

The simple SGD update offers lots of engineering opportunities. In practice, however, at any moment during the training procedure, we can select a small subset of training examples and try various learning rates on the subset, then pick the one that most reduces the cost and use it on the full dataset.

3.4 The SGD Convergence

The convergence of stochastic algorithms actually has been studied for a long time in adaptive signal processing. Given a suitable choice of the learning rate η_t, the standard (batch) gradient descent algorithm is known to converge to a local minimum of the cost function. However, the random noise introduced by SGD disrupts this deterministic picture and the specific study of SGD convergence usually is fairly complex (Benveniste *et al.*, 1987).

It is reported that for the convex case, if several assumptions and conditions are valid, then the SGD algorithm converges almost surely to the optimum $\mathbf{w}^*$ [1]. For the general case where the cost function is non-convex and has both local and global minima, if four assumptions and two learning rate assumptions hold, it is guaranteed that the gradient $\nabla_{\mathbf{w}}\mathcal{L}$ converges almost surely to zero (Bottou, 2004). We omit the details of the convergence theorem and corresponding proofs due to space limitation.

3.5 SGD Speedup

Unfortunately, many of sophisticated gradient methods are not robust to noise, and scale badly with the number of parameters. The plain SGD algorithm can be very slow to converge. Inspired by stochastic meta-descent (SMD) (Schraudolph, 1999), the convergence speed of SGD can be further improved with gradient step size adaptation by using second-order information. SMD is a highly scalable local optimizer. It shines when gradients are stochastically approximated.

In SMD, the learning rate η is simultaneously

[1] One may argue that SGD on many architectures does not result in a global optima. However, our goal is to obtain good performance on future examples in learning rather than achieving a global optima on the training set.

INPUT: training set $S\,\{(x_1, y_1), \ldots, (x_{\mathbb{T}}, y_{\mathbb{T}})\}$;
factor λ; number of iterations N.
INITIALIZE: $\mathbf{w}_0$, $\mathbf{v}_0 = 0$, η_0.
FOR $t = 1, 2, \ldots, N$
Choose a random example $(x_i, y_i) \in S$
Compute the gradient $\nabla_t = \mathbf{G}_t$ and $\mathbf{H}_t\mathbf{v}_t$
Set $\mathbf{v}_{t+1} = \lambda\mathbf{v}_t - \eta_t \cdot (\mathbf{G}_t + \lambda\mathbf{H}_t\mathbf{v}_t)$
Update the parameter vector:
$\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t - \eta_t \cdot \nabla_t$
Adapt the gradient step size:
$\eta_{t+1} = \eta_t \cdot \max(\frac{1}{2}, 1 - \mu\mathbf{G}_{t+1} \cdot \mathbf{v}_{t+1})$
OUTPUT: $\mathbf{w}_{N+1}$

Figure 1: Pseudo-code for the SGD algorithm.

adapted via a multiplicative update with μ:

$$\eta_{t+1} = \eta_t \cdot \max(\frac{1}{2}, 1 - \mu\mathbf{G}_{t+1} \cdot \mathbf{v}_{t+1}), \quad (17)$$

where the vector $\mathbf{v}$ ($\mathbf{v} =: \mathrm{d}\mathbf{w}$) captures the long-term dependencies of parameters. $\mathbf{v}$ can be computed by the simple iterative update:

$$\mathbf{v}_{t+1} = \lambda\mathbf{v}_t - \eta_t \cdot (\mathbf{G}_t + \lambda\mathbf{H}_t\mathbf{v}_t), \quad (18)$$

where the factor $0 \leq \lambda \leq 1$ governs the time scale over which long-term dependencies are taken into account, and $\mathbf{H}_t\mathbf{v}_t$ can be calculated efficiently alongside the gradient by forward-mode algorithmic differentiation via Equation 14. This Hessian-vector product is computed implicitly and it is the key to SMD's efficiency. The pseudo-code for the SGD algorithm is shown in Figure 1.

4 Experiments: A Case Study of NP Chunking

4.1 Data

Our data comes from the CoNLL 2000 shared task (Sang and Buchholz, 2000). The dataset is divided into a standard training set of 8,936 sentences and a testing set of 2,012 sentences. This data consists of the same partitions of the Wall Street Journal corpus (WSJ) as the widely used data for NP chunking: sections 15-18 as training data (211,727 tokens) and section 20 as test data (47,377 tokens). And the annotation of the data has been derived from the WSJ corpus.

$w_{t-\delta} = w$
w_t matches `[A-Z]`
w_t matches `[A-Z]+`
w_t matches `[A-Z][a-z]+`
w_t matches `[A-Z]+[a-z]+[A-Z]+[a-z]`
w_t matches `.*[0-9].*`
w_t contains dash "-" or dash-based "-based"
w_t is capitalized, all-caps, single capital letter, or mixed capitalization
w_t contains years, year-spans or fractions
w_t is contained in a lexicon of words with POS p (from the Brill tagger)
$p_t = p$
$q_k(x, t + \delta)$ for all k and $\delta \in [-3, 3]$

Table 1: Input feature template $q_k(x, t)$ for NP chunking. In this table w_t is the token (word) at position t, p_t is the POS tag at position t, w ranges over all words in the training data, and p ranges over all POS tags.

4.2 Features

We follow some top-performing NP chunking systems and perform holdout methodology to design features for our model, resulting in a rich feature set including POS features provided in the official CoNLL 2000 dataset (generated by the Brill tagger (Brill, 1995), with labeling accuracy of around 95-97%), some contextual and morphological features. Table 1 lists our feature set for NP chunking.

4.3 Experimental Results

We trained linear-chain conditional random fields (CRFs) (Lafferty *et al.*, 2001) as the baseline. The well known limited memory quasi-Newton BFGS algorithm (L-BFGS) (Liu and Nocedal, 1989) was applied to learn the parameters for CRFs. To avoid over-fitting, we penalized the log-likelihood by the commonly used zero-mean Gaussian prior over the parameters. This gives us a competitive baseline CRF model for NP chunking. To make fair and accurate comparison, we used the same set of features listed in Table 1 for both M^3N and CRFs. All experiments were performed on the Linux platform, with a 3.2GHz Pentium 4 CPU and 4 GB of memory.

Model	Training Method	Kernel Function	Iteration	Training Time(s)	P(%)	R(%)	$F_{\beta=1}$
M^3N	structured SMO	linear kernel: $\langle a, b\rangle_{\mathcal{H}}$	100	1176	94.59	94.22	94.40
M^3N	structured SMO	polynomial(quadratic): $(\langle a, b\rangle_{\mathcal{H}} + 1)^2$	100	30792	94.88	94.49	**94.68**
M^3N	structured SMO	polynomial(cubic): $(\langle a, b\rangle_{\mathcal{H}} + 1)^3$	100	30889	94.47	94.01	94.24
M^3N	structured SMO	polynomial(biquadratic): $(\langle a, b\rangle_{\mathcal{H}} + 1)^4$	100	31556	93.90	93.77	93.83
M^3N	structured SMO	neural kernel: $\tanh(0.1 \cdot \langle a, b\rangle_{\mathcal{H}})$	20	7395	94.42	94.02	94.22
CRFs	L-BFGS	—	100	352	94.55	94.09	94.32

Table 2: M^3N vs. CRFs: Performance and training time comparison for NP chunking on the CoNLL 2000 official dataset. M^3N was trained using the structured SMO algorithm.

Model	Training Method	Kernel Function	Iteration	Training Time(s)	P(%)	R(%)	$F_{\beta=1}$
M^3N	SGD	linear kernel: $\langle a, b\rangle_{\mathcal{H}}$	100	89	94.58	94.21	94.39
M^3N	SGD	polynomial(quadratic): $(\langle a, b\rangle_{\mathcal{H}} + 1)^2$	100	1820	94.89	94.50	**94.69**
M^3N	SGD	polynomial(cubic): $(\langle a, b\rangle_{\mathcal{H}} + 1)^3$	100	1831	94.47	94.01	94.24
M^3N	SGD	polynomial(biquadratic): $(\langle a, b\rangle_{\mathcal{H}} + 1)^4$	100	1857	93.91	93.76	93.83
M^3N	SGD	neural kernel: $\tanh(0.1 \cdot \langle a, b\rangle_{\mathcal{H}})$	20	477	94.40	94.01	94.20
CRFs	L-BFGS	—	100	352	94.55	94.09	94.32

Table 3: M^3N vs. CRFs: Performance and training time comparison for NP chunking on the CoNLL 2000 official dataset. M^3N was trained using the SGD algorithm.

System	$F_{\beta=1}$
SVMs (polynomial kernel) (Kudo and Matsumoto, 2000)	93.79
SVM combination (Kudo and Matsumoto, 2001)	94.39
Generalized winnow (Zhang *et al.*, 2002)	94.38
Voted perceptron (Collins, 2002)	94.09
CRFs (Sha and Pereira, 2003)	94.38
Second order CRFs (McDonald *et al.*, 2005)	94.29
Chunks from the Charniak Parser (Hollingshead *et al.*, 2005)	94.20
Second order latent-dynamic CRFs + improved A* search based inference (Sun *et al.*, 2008)	94.34
Our approach	94.69

Table 4: NP chunking: Comparison with some existing state-of-the-art systems.

Similar to other discriminative graphical models such as CRFs, the modeling flexibility of M^3N permits the feature functions to be complex, arbitrary, nonindependent, and overlapping features, allowing the multiple features described in Table 1 to be directly exploited. Moreover, M^3N is capable of incorporating multiple kernel functions (see Section 2) which allow the efficient use of high-dimensional feature spaces during the experiments.

The resulting number of features is 7,835,439, and both M^3N and CRFs were trained to predict 47,366 tokens with 12,422 noun phrases in the testing set. For simplicity, we denote $a = \phi(\mathbf{x}, \mathbf{y})$, and $b = \phi(\mathbf{x}', \mathbf{y}')$, and the linear kernel can be rewritten as $\kappa(a, b) = \langle a, b\rangle_{\mathcal{H}}$. We performed holdout methodology to find optimal values for coefficients s, d, and r in M^3N kernel functions. For polynomial kernels, we varied d from 2 to 4, resulting in quadratic, cubic, and biquadratic kernels, respectively. Finally, we chose optimized values: $s = 1, r = 1$ for polynomial kernels, and $s = 0.1, r = 0$ for neural kernels. The capacity $\mathcal{C}$ for M^3N was set to 1 in our experiments.

Table 2 shows comparative performance and training time for M^3N (trained with structured SMO) and CRFs, while Table 3 shows comparative performance and training time for M^3N (trained with SGD) and CRFs [2]. For M^3N, when trained with quadratic kernel and structured SMO, the best F-measure of 94.68 was achieved, leading to an improvement of 0.36 compared to the CRF baseline. What follows is the linear kernel that obtained 94.40 F-measure. The cubic and neural kernels obtained close performance, while the biquadratic kernel led to the worst performance. However, the structured SMO is very computationally intensive, especially for polynomial kernels. For example, CRFs converged in 352 sec-

[2]We used Taku Kudo's CRF++ toolkit (available at http://crfpp.sourceforge.net/) in our experiments. The M^3N model, and the structured SMO and SGD training algorithms were also implemented using C++.

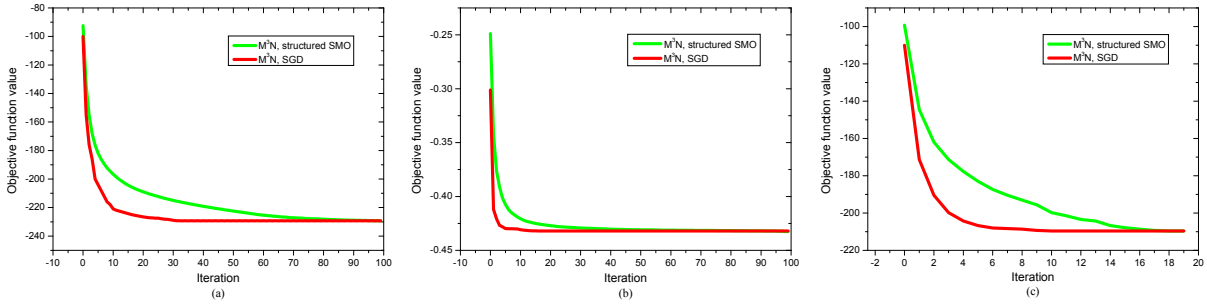


Figure 2: Convergence speed comparison for structured SMO and SGD algorithms. The X axis shows number of training iterations, and the Y axis shows objective function value. (a) The M^3N model was trained using linear kernel. (b) The M^3N model was trained using polynomial(quadratic) kernel. (c) The M^3N model was trained using neural kernel.

onds, while M^3N (polynomial kernels) took more than 8.5 hours to finish training.

As can be seen in Table 3, the SGD algorithm significantly accelerated the training procedure of M^3N without sacrificing performance. When the linear kernel was used, M^3N finished training in 89 seconds, more than 13 times faster than the model trained with structured SMO. And it is even much faster than the CRF model trained with L-BFGS. More importantly, SGD obtained almost the same performance as structured SMO with all M^3N kernel functions.

Table 4 gives some representative NP chunking results for previous work and for our best model on the same dataset. These results showed that our model compares favorably with existing state-of-the-art systems [3].

Figure 2 compares the convergence speed of structured SMO and SGD algorithms for the M^3N model. Linear (Figure 2 (a)), polynomial(quadratic) (Figure 2 (b)) and neural kernels (Figure 2 (c)) were used [4]. We calculated objective function values during effective training iterations. It can be seen that both structured SMO and SGD algorithms converge to the same objective function value for different kernels, but SGD converges considerably faster than the structured SMO.

Figure 3 (a) demonstrates the effect of training set size on performance for NP chunking. We increased the training set size from 1,000 sentences to 8,000 sentences, with an incremental step of 1,000. And the testing set was fixed to be 2,012 sentences. The M^3N models (with different kernels) were trained using the SGD algorithm. It is particularly interesting to know that the performance boosted for all the models when increasing the training set size. Using linear and quadratic kernels, M^3N model significantly and consistently outperforms the CRF model for different training set sizes. The cubic and neural kernels lead to almost the same performance for M^3N, which is slightly lower than the CRF baseline. As illustrated by the curves, M^3N (trained with quadratic kernel) achieved the best performance and larger training set size leads to better improvement for this model when compared to the CRF model, while M^3N (trained with biquadratic kernel) obtained the worst performance among all the models.

Accordingly, Figure 3 (b) shows the impact of increasing the training set size on training time for NP chunking. Increasing training set size leads to an increase in the computational complexity of training procedure for all models. For the M^3N model, it is faster when trained with linear kernel than the CRF model. And the three polynomial kernels (quadratic, cubic and biquadratic) have roughly the same training time. For CRFs and (M^3N, neural kernel), the training time is close to each other. For example, when the training set contains 1,000 sentences, the training time for CRFs, (M^3N, linear kernel), (M^3N, quadratic kernel), (M^3N, cubic kernel), (M^3N, biquadratic kernel), and (M^3N, neural kernel) is 24s, 7s, 72s,

[3] Note that it is difficult to compare strictly, since reported results sometimes leave out details (e.g., feature sets, significance tests, etc) needed for accurate comparison.

[4] For cubic and biquadratic kernels, the curves are very similar to that of quadratic kernel, and we omitted them for space.

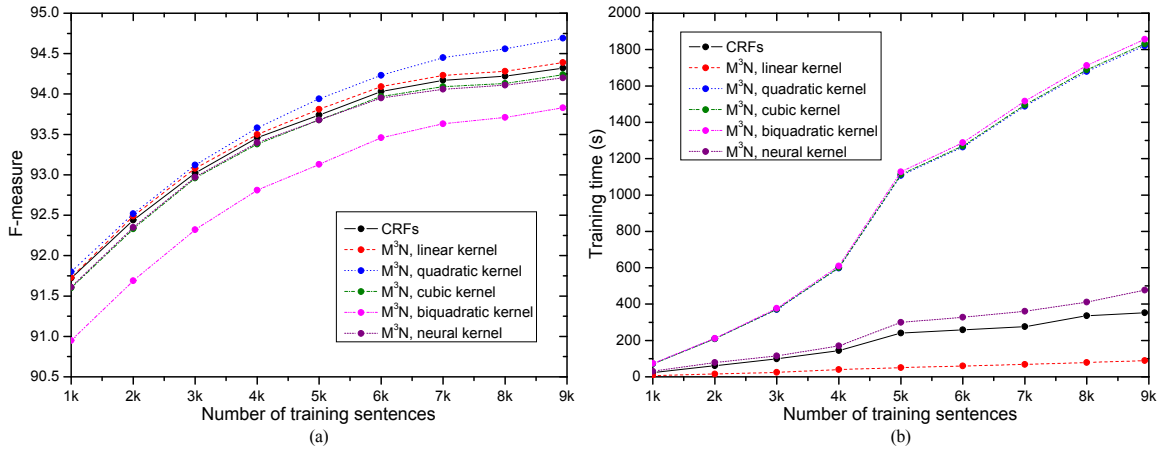


Figure 3: (a) Effect of training set size on performance for NP chunking. The training set size was increased from 1,000 sentences to 8,000 sentences, with an incremental step of 1,000. The testing set contains 2,012 sentences. All the M^3N models (with different kernels) were trained using the SGD algorithm. (b) Effect of training set size on training time for NP chunking.

72s, 74s, and 30s. When trained on 8,000 sentences, the numbers become 336s, 79s, 1679s, 1689s, 1712s, and 411s, respectively.

5 Related Work

The M^3N framework and its variants have generated much interest and great progress has been made, as evidenced by their promising results evaluated in handwritten character recognition, collective hypertext classification (Taskar *et al.*, 2003), parsing (Taskar *et al.*, 2004), and XML tag relabeling (Spengler, 2005). However, all the above mentioned research work used structured SMO algorithm for parameter learning, which can be computationally intensive, especially for very large datasets.

Recently, similar stochastic gradient methods have been applied to train log-linear models such as CRFs (Vishwanathan *et al.*, 2006). However, the maximum margin loss has a discontinuity in its derivative, making optimization of such models somewhat more involved than log-linear ones. We first exploit SGD method for fast parameter learning of M^3N and achieve state-of-the-art performance on the NP chunking task in the NLP community.

Several algorithms have been proposed to train max-margin models, including cutting plane SMO (Tsochantaridis *et al.*, 2005), exponentiated gradient (Bartlett *et al.*, 2004; Collins *et al.*, 2008), extragradient (Taskar *et al.*, 2006), and subgradient (Shalev-Shwartz *et al.*, 2007). Some methods are similar to SGD in that they all process a single training example at a time. The SGD methods directly optimize the primal problem, and at each update use a single example to approximate the gradient of the primal objective function. Some of the proposed algorithms, such as exponentiated gradient corresponds to block-coordinate descent in the dual, and uses the exact gradient with respect to the block being updated. We plan to implement and compare some of these algorithms with SGD for M^3N.

6 Conclusion and Future Work

We have presented the first known empirical study on sequence labeling based on M^3N. We have also provided the efficient SGD algorithm and shown how it can be applied to significantly speedup the training procedure of M^3N. As a case study, we performed extensive experiments on standard dataset for NP chunking, showing the promising and competitiveness of our approach. Several interesting issues, such as the convergence speed of the SGD algorithm, the effect of training set size on performance for NP chunking, and the effect of training set size on training time, were also investigated in our experiments. For the future work, we plan to further the scalability and applicability of our approach and evaluate it on other large-scale real world sequence labeling tasks, such as POS tagging and NER.

References

Peter L. Bartlett, Ben Taskar, Michael Collins, and David Mcallester. Exponentiated gradient algorithms for large-margin structured classification. In *Proceedings of NIPS-04*, pages 113–120. MIT Press, 2004.

A. Benveniste, M. Metivier, and P. Priouret. Algorithmes adaptatifs et approximations stochastiques. *Masson*, 1987.

Léon Bottou. Stochastic learning. In Olivier Bousquet and Ulrike von Luxburg, editors, *Advanced Lectures on Machine Learning*, Lecture Notes in Artificial Intelligence, LNAI 3176, pages 146–168. Springer Verlag, Berlin, 2004.

Eric Brill. Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging. *Computational Linguistics*, 21(4):543–565, 1995.

Michael Collins, Amir Globerson, Terry Koo, Xavier Carreras, and Peter L. Bartlett. Exponentiated gradient algorithms for conditional random fields and Max-margin Markov networks. *Journal of Machine Learning Research*, 9:1775–1822, 2008.

Michael Collins. Discriminative training methods for hidden Markov models: Theory and experiments with perceptron algorithms. In *Proceedings of HLT/EMNLP-02*, pages 1–8, 2002.

Kristy Hollingshead, Seeger Fisher, and Brian Roark. Comparing and combining finite-state and context-free parsers. In *Proceedings of HLT/EMNLP-05*, pages 787–794, Vancouver, British Columbia, Canada, 2005.

Taku Kudo and Yuji Matsumoto. Use of support vector learning for chunk identification. In *Proceedings of CoNLL-2000 and LLL-2000*, pages 142–144, Lisbon, Portugal, 2000.

Taku Kudo and Yuji Matsumoto. Chunking with support vector machines. In *Proceedings of HLT/NAACL-01*, pages 1–8, 2001.

John Lafferty, Andrew McCallum, and Fernando Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of ICML-01*, pages 282–289, 2001.

Yann Lecun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, Nov 1998.

Dong C. Liu and Jorge Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45:503–528, 1989.

Ryan McDonald, Koby Crammer, and Fernando Pereira. Flexible text segmentation with structured multilabel classification. In *Proceedings of HLT/EMNLP-05*, pages 987–994, Vancouver, British Columbia, Canada, 2005.

Barak A. Pearlmutter. Fast exact multiplication by the Hessian. *Neural Computation*, 6(1):147–160, 1994.

John C. Platt. Fast training of support vector machines using sequential minimal optimization. *Advances in Kernel Methods: Support Vector Learning*, pages 41–64, 1998.

H. Robbins and S. Monroe. A stochastic approximation method. *Annals of Mathematical Statistics*, 22:400–407, 1951.

Erik Tjong Kim Sang and Sabine Buchholz. Introduction to the CoNLL-2000 shared task: Chunking. In *Proceedings of CoNLL-2000*, pages 127–132, Lisbon, Portugal, 2000.

Bernhard Schölkopf and Alexander J. Smola. *Learning with Kernels*. MIT Press, Cambridge, MA, 2002.

Nicol N. Schraudolph. Local gain adaptation in stochastic gradient descent. In *Proceedings of the 9th International Conference on Artificial Neural Networks*, pages 569–574, 1999.

Fei Sha and Fernando Pereira. Shallow parsing with conditional random fields. In *Proceedings of HLT/NAACL-03*, pages 213–220, 2003.

Shai Shalev-Shwartz, Yoram Singer, and Nathan Srebro. Pegasos: Primal estimated sub-GrAdient SOlver for SVM. In *Proceedings of ICML-07*, pages 807–814, New York, NY, USA, 2007.

John Shawe-Taylor and Nello Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, Cambridge, UK, 2004.

Alex Spengler. Maximum margin Markov networks for XML tag relabelling. Master's thesis, University of Karlsruhe, 2005.

Xu Sun, Louis-Philippe Morency, Daisuke Okanohara, and Jun'ichi Tsujii. Modeling latent-dynamic in shallow parsing: A latent conditional model with improved inference. In *Proceedings of COLING-08*, pages 841–848, Manchester, UK, 2008.

Ben Taskar, Carlos Guestrin, and Daphne Koller. Max-margin Markov networks. In *Proceedings of NIPS-03*. MIT Press, 2003.

Ben Taskar, Dan Klein, Michael Collins, Daphne Koller, and Christopher Manning. Max-margin parsing. In *Proceedings of HLT/EMNLP-04*, pages 1–8, 2004.

Ben Taskar, Vassil Chatalbashev, Daphne Koller, and Carlos Guestrin. Learning structured prediction models: A large margin approach. In *Proceedings of ICML-05*, pages 896–903, Bonn, Germany, 2005.

Ben Taskar, Simon Lacoste-Julien, and Michael I. Jordan. Structured prediction via the extragradient method. In *Proceedings of NIPS-06*. MIT Press, 2006.

Ben Taskar. *Learning Structured Prediction Models: A Large Margin Approach*. PhD thesis, Stanford University, December 2004.

Ioannis Tsochantaridis, Thorsten Joachims, Thomas Hofmann, and Yasemin Altun. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, 6:1453–1484, 2005.

Vladimir N. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, Inc., New York, USA, 1995.

S. V. N. Vishwanathan, Nicol N. Schraudolph, Mark W. Schmidt, and Kevin P. Murphy. Accelerated training of conditional random fields with stochastic gradient methods. In *Proceedings of ICML-06*, pages 969–976, Pittsburgh, Pennsylvania, 2006.

Tong Zhang, Fred Damerau, and David Johnson. Text chunking based on a generalization of winnow. *Journal of Machine Learning Research*, 2:615–637, 2002.

Semi-automatically Developing Chinese HPSG Grammar from the Penn Chinese Treebank for Deep Parsing

Kun Yu[1] **Yusuke Miyao**[2] **Xiangli Wang**[1] **Takuya Matsuzaki**[1] **Junichi Tsujii**[1,3]

1. The University of Tokyo
{kunyu, xiangli, matuzaki, tsujii}
@is.s.u-tokyo.ac.jp

2. National Institute of Informatics
yusuke@nii.ac.jp

3. The University of Manchester

Abstract

In this paper, we introduce our recent work on Chinese HPSG grammar development through treebank conversion. By manually defining grammatical constraints and annotation rules, we convert the bracketing trees in the Penn Chinese Treebank (CTB) to be an HPSG treebank. Then, a large-scale lexicon is automatically extracted from the HPSG treebank. Experimental results on the CTB 6.0 show that a HPSG lexicon was successfully extracted with 97.24% accuracy; furthermore, the obtained lexicon achieved 98.51% lexical coverage and 76.51% sentential coverage for unseen text, which are comparable to the state-of-the-art works for English.

1 Introduction

Precise, in-depth syntactic and semantic analysis has become important in many NLP applications. Deep parsing provides a way of simultaneously obtaining both the semantic relation and syntactic structure. Thus, the method has become more popular among researchers recently (Miyao and Tsujii, 2006; Matsuzaki et al., 2007; Clark and Curran, 2004; Kaplan et al., 2004).

This paper introduces our recent work on deep parsing for Chinese, specifically focusing on the development of a large-scale grammar, based on the HPSG theory (Pollard and Sag, 1994). Because it takes a decade to manually develop an HPSG grammar that achieves sufficient coverage for real-world text, we use a semi-automatic approach, which has successfully been pursued for English (Miyao, 2006; Miyao et al., 2005; Xia, 1999; Hockenmaier and Steedman, 2002; Chen and Shanker, 2000; Chiang, 2000) and other languages (Guo et al., 2007; Cramer and Zhang, 2009; Hockenmaier, 2006; Rehbei and Genabith, 2009; Schluter and Genabith, 2009).

The following lists our method of approach: *(1) define a skeleton of the grammar (in this work, the structure of sign, grammatical principles and schemas)*, *(2) convert the CTB (Xue et al., 2002) into an HPSG-style treebank*, *(3) automatically extract a large-scale lexicon from the obtained treebank*.

Experiments were performed to evaluate the quality of the grammar developed from the CTB 6.0. More than 95% of the sentences in the CTB could be successfully converted, and the extracted lexicon was 97.24% accurate. The extracted lexicon achieved 98.51% lexical coverage and 76.51% sentential coverage for unseen text, which are comparable to the state-of-the-art works for English.

Since grammar engineering has many specific problems in each language, although we used the similar method applied in other languages to develop a Chinese HPSG grammar, it is very different from applying, such as statistical parsing models, to a new language. Lots of efforts have been done for the specific characteristics of Chinese. The contribution of our work is to describe these issues. As a result, a skeleton design of Chinese HPSG is proposed, and for the first time, a robust and wide-coverage Chinese HPSG grammar is developed from real-world text.

2 Design of Grammatical Constraints for Chinese HPSG

Because of the lack of a comprehensive HPSG-based syntactic theory for Chinese, we extended the original HPSG (Pollard and Sag, 1994) to analyze the specific linguistic phenomena in Chinese. Due to space limitations, we will provide a brief sampling of our extensions, and discuss several selected constructions.

2.1 Sign, Principles, and Schemas

Sign, which is a data structure to express grammatical constraints of words/phrases, is modified and extended for the analysis of Chinese specific constructions, as shown in Figure 1. *PHON, MOD, SPEC, SUBJ, MARKING,* and *SLASH* are

Coling 2008: Poster Volume, pages 1417–1425,
Beijing, August 2010

features defined in the original HPSG, and they represent the phonological information of a word, the constraints on the modifiee, the specifiee, the subject, the marker, and the long-distance dependency, respectively. *COMPS*, which represents the constraints on complements, is divided into *LCOMPS* and *RCOMPS*, to distinguish between left and right complements. Aspect, question, and negation particles are treated as markers as done in (Gao, 2000), which are distinguished by *ASPECT*, *QUESTION*, and *NEGATION*. *CONT* is also originated from Pollard and Sag (1994), although it is used to represent semantic structures with predicate-argument dependencies. *TOPIC* and *CONJ* are extended features that represent the constraints on the topic and the conjuncts of coordination. *FILLER* is another extended feature that records the grammatical function of the moved argument in a long-distance dependency.

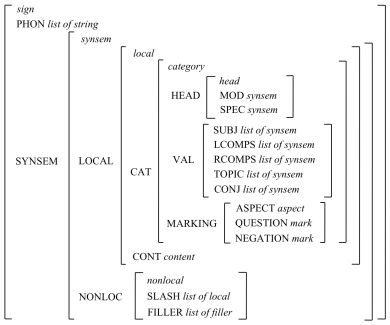


Figure 1. HPSG sign for Chinese.

The principles, including *Phonology Principle*, *Valence Principle*, *Head Feature Principle*, and *Nonlocal Feature Principle*, are implemented in our Chinese HPSG grammar as defined in (Pollard and Sag, 1994). *Semantic Principle* is slightly modified so that it composes predicate-argument structures.

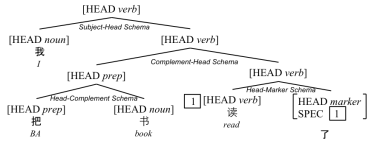
14 schemas are defined in our grammar, among which the *Coord-Empty-Conj Schema*, *Relative-Head Schema*, *Empty-Relativizer Schema*, and *Topic-Head Schema* are designed specifically for Chinese. The other 10 schemas are borrowed from the original HPSG theory.

15 Chinese constructions are considered in our current grammar (refer to Table 1). A detailed description of some particular constructions will be provided in the following subsection.

2.2 An HPSG Analysis for Chinese

2.2.1 BA Construction

The BA construction moves the object of a verb to the pre-verbal position. For example, the sentence in Figure 2 with the original word order is '我/I 读/read 了 书/book'. There were three popular ways to address the BA construction: as a verb (Huang, 1991; Bender, 2000), preposition (Gao, 1992), and case marker (Gao, 2000). Since the aspect markers, such as '了', cannot attach to BA, we exclude the analysis of treating BA as a verb. Because BA, like prepositions, always appears before a noun phrase, we therefore follow the analysis in Gao (1992), and treat BA as a preposition. As shown in Figure 2, BA takes a moved object as a complement, and attaches to the verb as a left-complement.

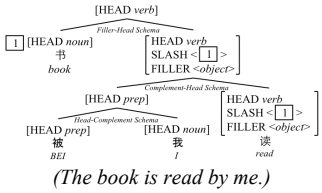


(I read the book.)

Figure 2[1]. Analysis of BA construction.

2.2.2 BEI Construction

The BEI construction is used to make the passive voice of a sentence. Because the aspect marker also cannot attach to BEI, we do not treat BEI as a verb, as done in the CTB. Similar to the analysis of BA construction, we regard BEI as a preposition that attaches to the verb as a left-complement. Additionally, because we can insert a clause '小李/Li 派/send 人/person' between the moved object '他/he' and the verb '打/beat', as is the case for '他/he 被/BEI 小李/Li 派/send 人/person 打/beat 了 (He was beaten by the person that is sent by Li)', we treat the relation between the moved object and the verb as a long-distance dependency. Figure 3 exemplifies our analysis of the BEI construction, in which the *Filler-Head Schema* is used to handle the long-distance dependency, and the *FILLER* feature is used to record that the role of the moved argument.



(The book is read by me.)

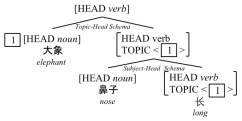
Figure 3. Analysis of BEI construction.

2.2.3 Topic Construction

As indicated in Li and Thompson (1989), a topic refers to the theme of a sentence, which always

[1] In the figures in this paper, we will show only selected features that are relevant to the explanation.

appears before the subject. The difference between the topic and subject is the subject must always have a direct semantic relationship with the verb in a sentence, whereas the topic does not. There are two types of topic constructions. In the first type, the topic does not fill any argument slots of the verb, such as the topic '大象/elephant' in Figure 4. In the second type, the topic has a semantic relationship with the verb. For example, in the sentence '他/he 我/I 喜欢/like (I like him)', the topic '他/he' is also an object of '喜欢/like'. For the first type, we define the *Topic-Head Schema* to describe the topic construction (refer to Figure 4). For the second type, we follow the same analysis as in English, and use the *Filler-Head Schema*.

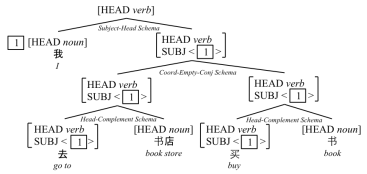


(The nose of an elephant is long.)

Figure 4. Analysis of topic construction.

2.2.4 Serial Verb Construction

In contrast to the definition of serial verb construction in Li and Thompson (1989), we specify a serial verb construction as a special type of verb phrase coordination, which describes several separate events with no conjunctions inside. Similar to ordinary coordination, the verb phrases in a serial verb construction share the same syntactic subject (Muller and Lipenkova, 2009), topic, and left-complement. We define *Coord-Empty-Conj Schema* to deal with it. Figure 5 shows an example analysis.

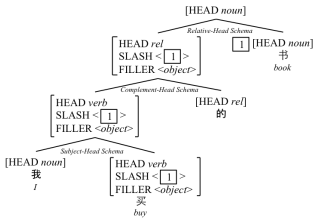


(I go to the book store and buy a book.)

Figure 5. Analysis of serial verb construction.

2.2.5 Relative Clause

In Chinese, a relative clause is marked by a relativizer '的' and exists in the left of the head noun. Because Chinese noun phrases are right-headed in general, we analyze a relative clause as a nominalization that modifies a head noun (Li and Thompson, 1989). Inside of a relative clause, the relativizer is treated as head. When the relativizer is omitted, we define a unary schema, *Empty-Relativizer Schema*, which functions by combining a relative clause with an empty relativizer. Furthermore, we introduce a *Relative-Head Schema* to handle the long-distance dependency for the extracted argument[2] (refer to Figure 6).



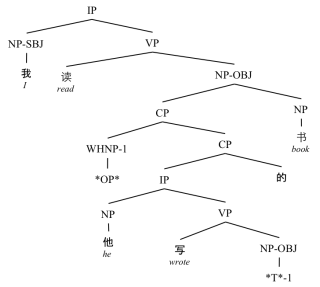
(the book that I buy)

Figure 6. Analysis of relative clause.

3 Converting the CTB into an HPSG Treebank

3.1 Partially-specified Derivation Tree Annotation

In order to convert the CTB into an HPSG treebank, we first annotate the bracketing trees in the CTB to be partially-specified derivation trees[3], which conform to the grammatical constraints designed in Section 2. Three types of rules are defined to fulfill this annotation.



(I read the book that he wrote.)

Figure 7. The CTB annotation for a sentence.

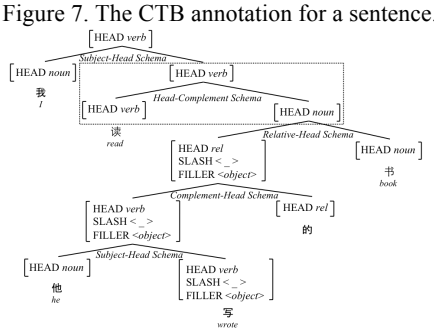


Figure 8. Partially-specified derivation tree for Figure 7.

For example, Figure 7 shows the bracketing tree of a sentence in the CTB, while Figure 8 shows the partially-specified derivation tree after re-annotation.

[2] The extracted adjunct is not treated as a long-distance dependency in our current grammar.

[3] *Partially-specified derivation tree* means a tree structure that is annotated with schema names and some features of the HPSG signs (Miyao, 2006).

3.1.1 Rules for Annotation Conversion

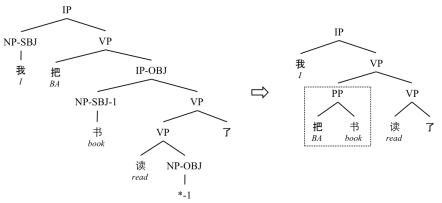
In the CTB, there exist some annotations that do not coincide with our HPSG analysis for Chinese. Therefore, we define pattern rules to convert the annotations in the CTB to fit with our HPSG analysis. 76 annotation rules are defined for 15 Chinese constructions (refer to Table 2). Due to page constraints, we focus on the constructions that we discussed in Section 2.

Construction	Rule #
Relative clause	20
BEI construction	21
Coordination	7
Subject/object control	5
Non-verbal predicate	4
Logical subject	3
Right node raising	3
Parenthesis	3
BA construction	3
Aspect/question/negation particle	2
Subordination	1
Serial Verb construction	1
Modal verb	1
Topic construction	1
Apposition	1

Table 1. Chinese constructions and annotation rules.

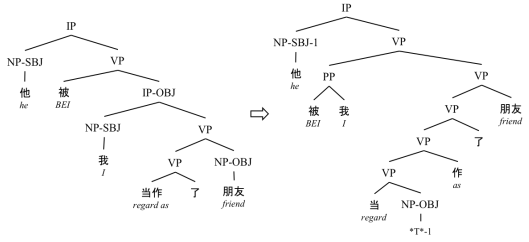
Rules for BA and BEI Construction

As analyzed in Section 2, we treat BA and BEI as prepositions that attach to the verb as left-complements. However, in the CTB, BA and BEI are annotated as verbs that take a sentential complement (Xue and Xia, 2000). By applying the annotation rules, the BA/BEI and the subject of the sentential complement of BA/BEI are re-annotated as a prepositional phrase (as indicated in the dash-boxed part in Figure 9).



(I read the book.)

Figure 9. Conversion of BA construction.



(He is regarded as a friend by me.)

Figure 10. Verb division in BEI construction.

In addition, in the CTB, some BA/BEI constructions are not annotated with trace, which makes it difficult to retrieve the semantic relation between the verb and the moved object. The principal reason for this is that the moved object in these constructions has a semantic relation with only part of the verb. For example, in Figure 10, the moved noun '他/he' is the object of '当/regard', but not for '当作/regard as'. Analysis shows that only a closed set of characters (e.g. '作/as') can be attached to verbs in such a case. Therefore, we manually collect these characters from the CTB, and then define pattern rules to automatically split the verb, which ends with the collected characters, in the BA and BEI construction. Finally, we annotate trace for the split verb. Figure 10 exemplifies the conversion of an example sentence.

Rules for Topic Construction

In the CTB, a functional tag 'TPC' is used to indicate a topic (Xue and Xia, 2000). Therefore, we use this functional tag to detect topic phrases during conversion.

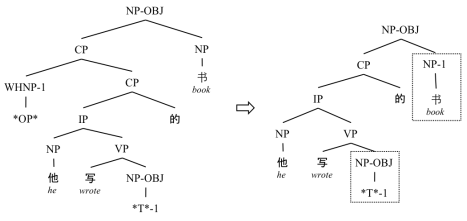
Rules for Serial Verb Construction

We define pattern rules to detect the parallel verb phrases with no conjunction inside (as shown in Figure 11), and treat these verb phrases as a serial verb construction. However, when the verb in the first phrase is a modal verb, such as the case of '我/I 想/want to 唱歌/sing (I want to sing)', the parallel verb phrases should not be treated as a serial verb construction. Therefore, a list of modal verbs is manually collected from the CTB to filter out these exceptional cases during conversion.

(go downstairs and eat meal)

Figure 11. An example of parallel verb phrases.

Rules for Relative Clause



(the book that he wrote)

Figure 12. Conversion of relative clause.

We define annotation rules to slightly modify the annotation of a relative clause in CTB, as shown in Figure 12, to make the tree structure easy to be

analyzed. Furthermore, in CTB, relative clauses are annotated with both extracted arguments and extracted adjuncts. But in our grammar, we only deal with extracted arguments, and the gap in a relative clause (as indicated in the dash-boxed part in Figure 12). When the extracted phrase is an adjunct of the relative clause, we simply view the clause as a modifier of the extracted phrase.

3.1.2 Rules for Correcting Inconsistency

There are some inconsistencies in the annotation of the CTB, which presents difficulties for performing the derivation tree annotation. Therefore, we define 49 rules, as done in (Hockenmaier and Steedman, 2002) for English, to mitigate inconsistencies before annotation (refer to Table 3).

3.1.3 Rules for Assisting Annotation

We also define 48 rules (refer to Table 2), which are similar to the rules used in (Miyao, 2006) for English, to help the derivation tree annotation. For example, 12 pattern rules are defined to assign the schemas to corresponding constituents.

Rule Type	Rule Description	Rule #
Rules for correcting inconsistent annotation	Fix tree annotation	37
	Fix phrase tag annotation	5
	Fix functional tag annotation	5
	Fix POS tag annotation	2
Rules for assisting annotation	Slash recognization	27
	Schema assignment	12
	Head/Argument/Modifier marking	8
	Binarization	1

Table 2. Rules for correcting inconsistency and assisting annotation.

3.2 HPSG Treebank Acquisition

In this phase, the schemas and principles are applied to the annotated partially-specified trees, in order to fill out unspecified constraints and validate the consistency of the annotated constraints. In effect, an HPSG treebank is obtained.

For instance, by applying the *Head-Complement Schema* to the dash-boxed nodes in Figure 8, the constraints of the right daughter are percolated to *RCOMPS* of the left daughter (as indicated as 4 in Figure 13). After applying the schemas and the principles to the whole tree in Figure 8, a HPSG derivation tree is acquired (refer to Figure 13).

3.3 Lexicon Extraction

With the HPSG treebank acquired in Section 3.2, we automatically collect lexical entries as the combination of words and lexical entry templates from the terminal nodes of the derivation trees. For example, from the HPSG derivation tree shown in Figure 13, we obtain a lexical entry for the word '写/write' as shown in Figure 14.

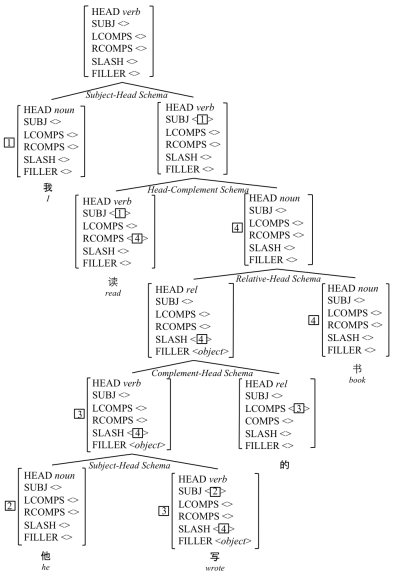


Figure 13. HPSG derivation tree for Figure 8.

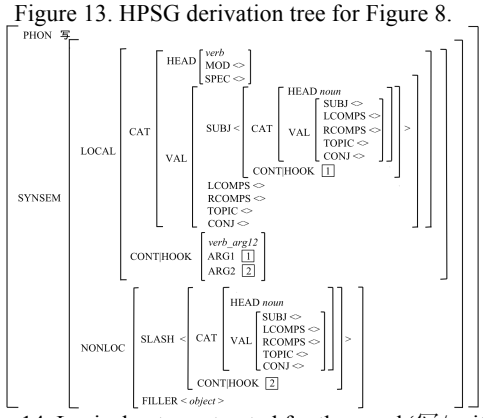
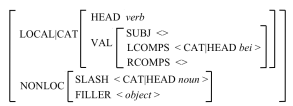
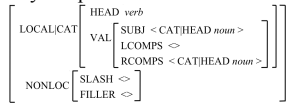


Figure 14. Lexical entry extracted for the word '写/write'.

3.3.1 Lexical Entry Template Expansion



(a) Lexical entry template for the verb in BEI construction



(b) Lexical entry template for the verb in original word order

Figure 15. Application of a lexical rule.

Some Chinese constructions change the word order of sentences, such as the BA/BEI constructions. Therefore, we apply lexical rules (Nakanishi et al., 2004) to the lexical entry templates to convert them into those for the original word order, and expand the lexical entry templates consequently. 18 lexical rules are defined for the verbs in the BA/BEI constructions. For example, by applying a lexical rule to the lexical entry template in Figure 15(a), the moved object indi-

cated by *SLASH* is restored into *RCOMPS*, and the subject introduced by BEI in *LCOMPS* is restored into *SUBJ* (refer to Figure 15(b)).

3.3.2 Mapping of Semantics

In our grammar, we use predicate-argument dependencies for semantic representation. 44 types of predicate-argument relations are defined to represent the semantic structures of 13 classes of words. For example, we define a predicate-argument relation '*verb_arg12*', in which a verb takes two arguments '*ARG1*' and '*ARG2*', to express the semantics of transitive verbs. 72 semantics mapping rules are defined to associate these predicate-argument relations with the lexical entry templates. Figure 16 exemplifies a semantics mapping rule. The input of this rule is the lexical entry template (as shown in the left part), and the output is a predicate-argument relation '*verb_arg12*' (as shown in the right part), which associates the syntactic arguments *SUBJ* and *SLASH* with the semantic arguments *ARG1* and *ARG2* (as indicated by 1 and 2 in Figure 16).

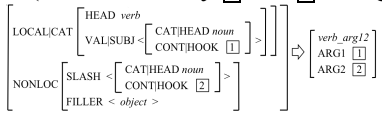


Figure 16. A semantics mapping rule.

4 Evaluation

4.1 Experimental Setting

We used the CTB 6.0 for HPSG grammar development and evaluation. We split the corpus into development, testing, and training data sets, following the recommendation from the corpus author. The development data was used to tune the design of grammar constraints and the annotation rules. However, the testing data set was reserved for further evaluation on parsing. Thus, the training data was further divided into two parts for training and testing in this work. During the evaluation, unknown words were handled in the same way as done in (Hockenmaier and Steedman, 2002).

4.2 Evaluation Metrics

In order to verify the quality of the grammar developed in our work, we evaluated the extracted lexicon by the accuracy for assessing the semi-automatic conversion process, and the coverage for quantifying the upper-bound coverage of the future HPSG parser based on this grammar.

The accuracy of the extracted lexicon was evaluated by *lexical accuracy*, which counts the number of the correct lexical entries among all the obtained lexical entries.

In addition, two evaluation metrics as used in (Hockenmaier and Steedman, 2002; Xia, 1999; Miyao, 2006) were used to evaluate the coverage of the obtained lexicon. The first one is *lexical coverage* (Hockenmaier and Steedman, 2002; Xia, 1999), which means that the percentage that the lexical entries extracted from the testing data are covered by the lexical entries acquired from the training data. The second one is *sentential coverage* (Miyao, 2006): a sentence is considered to be covered only when the lexical entries of all the words in this sentence are covered.

4.3 Results of Accuracy

Since there was no gold standard data for the automatic evaluation of accuracy, we randomly selected 100 sentences from the testing data, and manually checked the lexical entries extracted from these sentences. Results show that 1,558 lexical entries were extracted at 97.24% (1,515/1,558) accuracy.

Error analysis shows all the incorrect lexical entries came from the error in the derivation tree annotation. For example, our current design failed to find the correct boundary of coordinated noun phrases when the word '等/etc' was attached at the end, such as '产权/property right 出让/selling 、 资产/assets 出租/renting 等/etc (property right selling and assets renting etc.)'. We will improve the derivation tree annotation to solve this issue.

4.4 Results of Coverage

Table 3 shows the coverage of the extracted lexical entries, which indicates that a large HPSG lexicon was successfully extracted from the CTB for unseen text, with reasonable coverage. The statistics of the HPSG lexicon extraction in our experiments (refer to Table 4) also indicates that we successfully extracted lexical entries from more than 95% of the sentences in the CTB.

Among all the uncovered lexical entries, 78.55% are for content words, such as verb and noun. In addition, the classification of uncovered lexical entries in Table 4 indicates that about 1/3 of the uncovered lexical entries came from the unknown lexical entry templates ('+*w*/-*t*'). We analyzed the 193 '+*w*/-*t*' failures in the testing data, among which 169 failures resulted from the shortage of training data, which indicated that the correct lexical entry template did not appear in

the training data. The learning curve in Figure 17 shows that we can resolve this issue by enlarging the training data. The other 24 failures came from the error in the derivation tree annotation. For example, our current grammar failed at detecting the coordinated clauses when they were separated by a colon. We will be able to reduce this type of failure by improving the derivation tree annotation.

Sent. Cov.	Lex. Cov.	Uncovered Lexical Entries	
		+w/+t	*+w/-t*
76.51%	98.51%	1.05%	0.43%

Table 3[4]. Coverage of extracted HPSG lexicon.

Data Set	Total Sent #	Succeed Sent #	Word #	Lexical Entry Template #
Training	20,230	19,257(95.19%)	510,815	4,836
Develop	2,067	2,009(97.19%)	55,714	1,582
Testing	2,000	1,941(97.05%)	44,924	1,163

Table 4. Statistics of HPSG lexicon extraction.

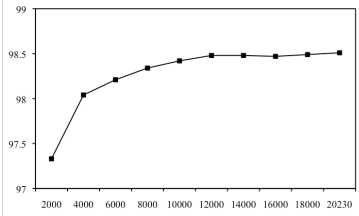


Figure 17. Lexical coverage (Y axis) vs. corpus size (X axis).

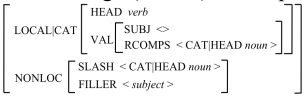


Figure 18. A lexical entry template extracted from testing data.

The other type of failures (‘*+w/+t*’) indicate that a word was incorrectly associated with a lexical entry template, even though both of them existed in the training data. Error analysis shows that 64.39% of failures were related to verbs. For example, for a relative clause ‘投资/invest 台湾/Taiwan 的 商人/businessman (the businessman that invests Taiwan)’ in the testing data, we associated a lexical entry template as shown in Figure 18 with the verb ‘投资/invest’. In the training data, however, the lexical entry template shown in Figure 18 cannot be extracted for ‘投资/invest’, since this word never appears in a relative clause with an extracted subject. Introducing lexical rules to expand the lexical entry template of verbs in a relative clause is a possible way to solve this problem.

4.5 Comparison with Previous Work

Guo’s work (Guo et al., 2007; Guo, 2009) is the only previous work on Chinese lexicalized grammar development from the CTB, which induced wide-coverage LFG resources from the CTB. By using the hand-made gold-standard f-structures of 200 sentences from the CTB 5.1, the LFG f-structures developed in Guo’s work achieved 96.34% precision and 96.46% recall for unseen text (Guo, 2009). In our work, we applied the similar strategy in evaluating the accuracy of the developed Chinese HPSG grammar, which achieved 97.24% lexical accuracy on 100 unseen sentences from the CTB 6.0. When evaluating the coverage of our grammar, we used a much larger data set (including 2,000 unseen sentences), and achieved 98.51% lexical coverage. Although these results cannot be compared to Guo’s work directly because of the different size and content of data set, it indicates that the Chinese HPSG grammar developed in our work is comparable in quality with Guo’s work.

In addition, there were previous works about developing lexicalized grammar for English. Considering the small size of the CTB, in comparison to the Penn Treebank used in the previous works, the results listed in Table 5 verify that, the quality of the Chinese HPSG grammar developed in our work is comparable to these previous works.

Previous Work	Sent. Cov.	Lex. Cov.
Miyao (2006)	82.50%	98.97%
Hockenmaier and Steedman (2002)	-	98.50%
Xia (1999)	-	96.20%

Table 5. Evaluation results of previous work.

4.6 Discussion

There are still some sentences in the CTB from which we failed to extract lexical entries. We analyzed the 59 failed sentences in the testing data and listed the reasons in Table 6.

Reason	Sent #
Error in the derivation tree annotation	31
Short of semantics mapping rule	23
Inconsistent annotation in the CTB	5

Table 6. Reasons for lexicon extraction failures.

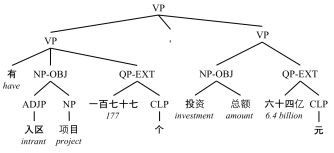
The principal reason for 31 sentence failures, is the error in the derivation tree annotation. For instance, our current annotation rules could convert the regular relative clause shown in Figure 12. Nonetheless, when the relative clause is inside of a parenthesis, such as ‘“ 原始/primitive 的 ”方法/method (the method that is primitive)’, the annotation rules failed at finding the extracted head noun to create a derivation tree. This type of failure can be reduced by improving the annotation rules.

[4] ‘*+w/+t*’ means both the word and lexical entry template have been seen in the lexicon. ‘*+w/-t*’ means only the word has been seen in the lexicon (Hockenmaier and Steedman, 2002).

The second reason, for which 23 sentences failed, is the shortage of the semantics mapping rules. For example, we did not define semantics mapping rule for a classifier that acts as a predicate with two topics. This type of failure can be reduced by adding semantic mapping rules.

The last reason for sentence failures is inconsistencies in the CTB annotation. In our future work, these inconsistencies will be collected to enrich our inconsistency correction rules.

In addition to the reasons above, some sentences with special constructions in the development and training data also could not be analyzed by our current grammar, since the special construction is difficult for the current HPSG to analyze. The special constructions include the argument-cluster coordination shown in Figure 19. Introducing the similar rules used in CCG (Hockenmaier and Steedman, 2002) could be a possible solution to this problem.



(have 177 intrant projects and 6.4 billion investments)

Figure 19. An argument-cluster coordination in CTB.

5 Related Work

To the extent of our knowledge, the only previous work about developing Chinese lexicalized grammar from treebanks is Guo's work (Guo et al., 2007; Guo, 2009). An LFG-based parsing using wide-coverage LFG approximations induced from the CTB was done in this work. However, they did not train a deep parser based on the LFG resources obtained in their work, but relied on an external PCFG parser to create c-structure trees, and then mapped the c-structure trees into f-structures using their annotation rules (Guo, 2009). In contrast to Guo's work, we paid particular attention to a different grammar framework, i.e. HPSG, with the analysis of more Chinese constructions, such as the serial verb construction. In addition, in our on-going deep parsing work, we use the developed Chinese HPSG grammar, i.e. the lexical entries, to train a full-fledged HPSG parser directly.

Additionally, there are some works that induce lexicalized grammar from corpora for other languages. For example, by using the Penn Treebank, Miyao et al. (2005) automatically extracted a large HPSG lexicon, Xia (1999), Chen and Shanker (2000), Hockenmaier and Steedman (2002), and Chiang (2000) invented LTAG/CCG specific procedures for lexical entry extraction. From the German Tiger corpus, Cramer and Zhang (2009) constructed a German HPSG grammar; Hockenmaier (2006) created a German CCGbank; and Rehbei and Genabith (2009) acquired LFG resources. In addition, Schluter and Genabith (2009) automatically obtained wide-coverage LFG resources from a French Treebank. Our work implements a similar idea to these works, but we apply different grammar design and annotation rules, which are specific to Chinese. Furthermore, we obtained a comparative result to state-of-the-art works for English.

There are some researchers who worked on Chinese HPSG grammar development manually. Zhang (2004) implemented a Chinese HPSG grammar using the LinGO Grammar matrix (Bender et al., 2002). Only a few basic constructions were considered, and a small lexicon was constructed in this work. Li (1997) and Wang et al. (2009) designed frameworks for Chinese HPSG grammar; however, only small grammars were implemented in these works.

Furthermore, some linguistic works focused mainly on the discussion of specific Chinese constructions in the HPSG or LFG framework, without implementing a grammar for real-world text (Bender, 2000; Gao, 2000; Li and McFetridge, 1995; Li, 1995; Xue and McFetridge, 1995; Wang and Liu, 2007; Ng, 1997; Muller and Lipenkova, 2009; Liu, 1996; Kit, 1998).

6 Conclusion and Future Work

In this paper, we described the semi-automatic development of a Chinese HPSG grammar from the CTB. Grammatical constraints are first designed by hand. Then, we convert the bracketing trees in the CTB into an HPSG treebank, by using pre-defined annotation rules. Lastly, we automatically extract lexical entries from the HPSG treebank. We evaluated our work on the CTB 6.0. Results indicated that a large HPSG lexicon was successfully extracted with a 97.24% accuracy. Furthermore, our grammar achieved 98.51% lexical coverage and 76.51% sentential coverage for unseen text.

This is an ongoing work, and there are some future works under consideration, including enriching the design of annotation rules, introducing more semantics mapping rules, and adding lexical rules. In addition, the work on Chinese HPSG parsing is on-going, within which the Chinese HPSG grammar developed in this work will be available soon.

References

Emily Bender. 2000. The Syntax of Madarin Ba: Reconsidering the Verbal Analysis. *Journal of East Asian Linguistics*. 9(2): 105-145.

Emily Bender, Dan Flickinger, and Stephan Oepen. 2002. The Grammar Matrix: An Open-source Starter-lit for the Rapid Development of Cross-linguistically Consistent Broad-coverage Precision Grammars. *Procedings of the Workshop on Grammar Engineering and Evaluation*.

John Chen and Vijay K. Shanker. 2004. Automated Extraction of TAGs from the Penn Treebank. *Proceedings of the 6th IWPT*.

David Chiang. 2000. Statistical Parsing with an Automatically-extracted Tree Adjoining Grammar. *Proceedings of the 38th ACL*. 456-463.

Stephen Clark and James R. Curran. 2004. Parsing the WSJ Using CCG and Log-linear Models. *Proceedings of the 42nd ACL*.

Bart Cramer and Yi Zhang. 2009. Construction of a German HPSG Grammar from a Detailed Treebank. *Proceedings of the 2009 Workshop on Grammar Engineering Across Frameworks*.

Qian Gao. 1992. *Chinese Ba Construction: its Syntax and Semantics*. Technical report.

Qian Gao. 2000. *Argument Structure, HPSG and Chinese Grammar*. Ph.D. Thesis. Ohio State University.

Yuqing Guo. 2009. *Treebank-based acquisition of Chinese LFG Resources for Parsing and Generation*. Ph.D. Thesis. Dublin City University.

Yuqing Guo, Josef van Genabith and Haifeng Wang. 2007. Acquisition of Wide-Coverage, Robust, Probabilistic Lexical-Functional Grammar Resources for Chinese. *Proceedings of the 12th International Lexical Functional Grammar Conference (LFG 2007)*. 214-232.

Julia Hockenmaier. 2006. Creating a CCGbank and a wide-coverage CCG lexicon for German *Proceedings of COLING/ACL 2006*.

Julia Hockenmaier and Mark Steedman. 2002. Acquiring Compact Lexicalized Grammars from a Cleaner Treebank. *Proceedings of the 3rd LREC*.

C-R Huang. 1991. Madarin Chinese and the Lexical Mapping Theory: A Study of the Interaction of Morphology and Argument Changing. *Bulletin of the Institute of History and Philosophy 62*.

Ronald M. Kaplan et al. 2004. Speed and Accuracy in Shallow and Deep Stochastic Parsing. *Proceedings of HLT/NAACL 2004*.

Chunyu Kit. 1998. Ba and Bei as Multi-valence Prepositions in Chinese. *Studia Linguistica Sinica*: 497-522.

Wei Li. 1995. Esperanto Inflection and its Interface in HPSG. *Proceedings of the 11th North West Linguistics Conference*.

Wei Li. 1997. Outline of an HPSG-style Chinese Reversible Grammar. *Proceedings of the 13th North West Linguistics Conference*.

Wei Li and Paul McFetridge. 1995. Handling Chinese NP Predicate in HPSG. *Proceedings of PACLING-II*.

Charles N. Li and Sandra A. Thompson. 1989. *Mandarin Chinese: A Functional Reference Grammar*. University of California Press, London, England.

Takuya Matsuzaki, Yusuke Miyao, and Junichi Tsujii. 2007. Efficient HPSG Parsing with Supertagging and CFG-filtering. *Proceedings of the 20th IJCAI*.

Yusuke Miyao. 2006. *From Linguistic Theory to Syntactic Analysis: Corpus-oriented Grammar Development and Feature Forest Model*. Ph.D. Thesis. The University of Tokyo.

Yusuke Miyao, Takashi Ninomiya and Junichi Tsujii. 2005. Corpus-oriented Grammar Development for Acquiring a Head-driven Phrase Structure Grammar from the Penn Treebank. *Natural Language Processing - IJCNLP 2005*: 684-693.

Yusuke Miyao and Junichi Tsujii. 2008. Feature Forest Models for Probabilistic HPSG Parsing. *Computational Linguistics*. 34(1): 35-80.

Stefan Muller and Janna Lipenkova. 2009. Serial Verb Constructions in Chinese: A HPSG Account. *Proceedings of the 16th International Conference on Head-Driven Phrase Structure Grammar*. 234-254.

Hiroko Nakanishi, Yusuke Miyao and Junichi Tsujii. 2004. An Empirical Investigation of the Effect of Lexical Rules on Parsing with a Treebank Grammar. *Proceedings of the 3rd TLT*. 103-114.

Say K. Ng. 1997. *A Double-specifier Account of Chinese NPs Using Head-driven Phrase Structure Grammar*. Master Thesis. Department of Linguistics, University of Edinburgh.

Carl Pollard and Ivan A. Sag. 1994. *Head-Driven Phrase Structure Grammar*. The University of Chicago Press and CSLI Publications, Chicago, IL and Stanford, CA.

Ines Rehbein and Josef van Genabith. 2009. Automatic Acquisition of LFG Resources for German – As Good as it Gets. *Proceedings of the 14th International Lexical Functional Grammar Conference (LFG 2009)*.

Natalie Schluter and Josef van Genabith. 2008. Treebank-based Acquisition of LFG Parsing Resources for French. *Proceedings of the 6th LREC*.

Mark Steedman. 2000. *The Syntactic Process*. The MIT Press.

Xiangli Wang et al. 2009. Design of Chinese HPSG Framework for Data-driven Parsing. *Proceedings of the 23rd Pacific Asia Conference on Language, Information and Computation*.

Lulu Wang and Haitao Liu. 2007. A Description of Chinese NPs Using Head-driven Phrase Structure Grammar. *Proceedings of the 14th International Conference on Head-Driven Phrase Structure Grammar*. 287-305.

Fei Xia. 1999. Extracting Tree Adjoining Grammars from Bracketed Corpora. *Proceedings of the 5th NLPRS*.

Nianwen Xue, Fudong Chiou, and Martha Palmer. 2002. Building a Large-scale Annotated Chinese Corpus. *Proceedings of COLING 2002*.

Ping Xue and Paul McFetridge. 1995. DP Structure, HPSG, and the Chinese *NP. Proceedings of the 14th Annual Conference of Canadian Linguistics Association*.

Nianwen Xue and Fei Xia. 2000. *The Bracketing Guidelines for the Penn Chinese Treebank*.

Yi Zhang. 2004. Starting to Implement Chinese Resource Grammar using LKB and LinGO Grammar Matrix. Technical report.

Cross-Lingual Induction for Deep Broad-Coverage Syntax: A Case Study on German Participles

Sina Zarrieß **Aoife Cahill** **Jonas Kuhn** **Christian Rohrer**
Institut für Maschinelle Sprachverarbeitung (IMS), University of Stuttgart
{zarriesa,cahillae,jonas.kuhn,rohrer}@ims.uni-stuttgart.de

Abstract

This paper is a case study on cross-lingual induction of lexical resources for deep, broad-coverage syntactic analysis of German. We use a parallel corpus to induce a classifier for German participles which can predict their syntactic category. By means of this classifier, we induce a resource of adverbial participles from a huge monolingual corpus of German. We integrate the resource into a German LFG grammar and show that it improves parsing coverage while maintaining accuracy.

1 Introduction

Parallel corpora are currently exploited in a wide range of induction scenarios, including projection of morphologic (Yarowsky et al., 2001), syntactic (Hwa et al., 2005) and semantic (Padó and Lapata, 2009) resources. In this paper, we use cross-lingual data to learn to predict whether a lexical item belongs to a specific syntactic category that cannot easily be learned from monolingual resources. In an application test scenario, we show that this prediction method can be used to obtain a lexical resource that improves deep, grammar-based parsing.

The general idea of cross-lingual induction is that linguistic annotations or structures, which are not available or explicit in a given language, can be inferred from another language where these annotations or structures are explicit or easy to obtain. Thus, this technique is very attractive for cheap acquisition of broad-coverage resources, as is proven by the approaches cited above. Moreover, this induction process can be attractive for the induction of deep (and perhaps specific) linguistic knowledge that is hard to obtain in a monolingual context. However, this latter perspective has been less prominent in the NLP community so far.

This paper investigates a cross-lingual induction method based on an exemplary problem arising in the deep syntactic analysis of German. This showcase is the syntactic flexibility of German participles, being morphologically ambiguous between verbal, adjectival and adverbial readings, and it is instructive for several reasons: first, the phenomenon is a notorious problem for linguistic analysis and annotation of German, such that standard German resources do not represent the underlying analysis. Second, in Zarrieß et al. (2010), we showed that integrating the phenomenon of adverbial participles in a naive way into a broad-coverage grammar of German leads to significant parsing problems, due to spurious ambiguities. Third, it is completely straightforward to detect adverbial participles in cross-lingual data since in other languages, e.g. English or French, adverbs are often morphologically marked.

In this paper, we use instances of adverbially translated participles in a parallel corpus to bootstrap a classifier that is able to identify an adverbially used participle based on its monolingual syntactic context. In contrast to what is commonly assumed, we show that it is possible to detect adverbial participles using only a relatively narrow context window. This classifier enables us to identify an occurence of an adverbial participle independently of its translation in a parallel corpus, going far beyond the induction methodology in Zarrieß et al. (2010). By means of the participle classifier, we can extract new types of adverbial participles from a larger corpus of German newspaper text and substantially augment the size of the resource extracted only on Europarl data. Finally, we integrate this new resource into the German LFG grammar and show that it improves coverage without negatively affecting performance.

Coling 2008: Poster Volume, pages 1426–1434,
Beijing, August 2010

The paper is structured as follows: in Section 2, we describe the linguistic and computational problems related to the parsing of adverbial participles in German. Section 3 introduces the general idea of using the translation data to find instances of different participle categories. In Section 4, we illustrate the training of the classifier, evaluating the impact of the context window and the quality of the training data obtained from cross-lingual text. In Section 5, we apply the classifier to new, monolingual data and describe the extension of the resource for adverbial participles. Section 6 evaluates the extended resource by means of parsing experiments using the German LFG grammar.

2 The Problem

In German, past perfect participles are ambiguous with respect to their morphosyntactic category. As in other languages, they can be used as part of the verbal complex (Example (1-a)) or as adjectives (Example (1-b)). Since German adjectives can generally undergo conversion into adverbs, participles can also be used adverbially (Example (1-c)). The verbal and adverbial participle forms are morphologically identical.

(1) a. Sie haben das Experiment **wiederholt**.
'They have repeated the experiment.'
b. Das **wiederholte** Experiment war erfolgreich.
'The repeated experiment was succesful.'
c. Sie haben das Experiment **wiederholt** abgebrochen.
'They cancelled the experiment repeatedly.'

Moreover, German adjectival modifiers can be generally used as predicatives that can be either selected by a verb (Example (2-a)) or that can occur as free predicatives (Example (2-b)).

(2) a. Er scheint **begeistert** von dem Experiment.
'He seems enthusiastic about the experiment.'
b. Er hat **begeistert** experimentiert.
'He has experimented enthusiastic.'

Since predicative adjectives are not inflected, the surface form of a German participle is ambiguous between a verbal, predicative or adverbial use.

2.1 Participles in the German LFG

In order to account for sentences like (1-c), an intuitive approach would be to generally allow for adverb conversion of participles in the grammar. However, in Zarrieß et al. (2010), we show that such a rule can have a strong negative effect on the overall performance of the parsing system, despite the fact that it produces the desired syntactic and semantic analysis for specific sentences. This problem was illustrated using a German LFG grammar (Rohrer and Forst, 2006) constructed as part of the ParGram project (Butt et al., 2002). The grammar is implemented in the XLE, a grammar development environment which includes a very efficient LFG parser and a stochastic disambiguation component which is based on a log-linear probability model (Riezler et al., 2002).

In Zarrieß et al. (2010), we found that the naive implementation of adverbial participles in the German LFG, i.e. in terms of a general grammar rule that allows for participles-adverb conversion, leads to spurious ambiguities that mislead the disambiguation component of the grammar. Moreover, the rule increases the number of timeouts, i.e. sentences that cannot be parsed in a predefined amount of time (20 seconds). Therefore, we observe a drop in parsing accuracy although grammar coverage is improved. As a solution, we induced a lexical resource of adverbial participles based on their adverbial translations in a parallel corpus. This resource, comprising 46 participle types, restricts the adverb conversion such that most of the spurious ambiguities are eliminated.

To assess the impact of specific rules in a broad-coverage grammar, possibly targeting medium-to-low frequency phenomena, we have established a fine-grained evaluation methodology. The challenge posed by these low-frequent phenomena is typically two-fold: on the one hand, if one takes into account the disambiguation component of the grammar and pursues an evaluation of the most probable parses on a general test set, the new grammr rule cannot be expected to show a positive effect since the phenomenon is not likely to occur very often in the test set. On the other hand, if one is interested in a linguistically precise grammar, it is very unsatisfactory to reduce grammar coverage to statistically frequent phenomena. Therefore, we combined a coverage-oriented evaluation on specialised testsuites with a quantitative evaluation including disambiguation, making sure that

the increased coverage does not lead to an overall drop in accuracy. The evaluation methodolgy will also be applied to evaluate the impact of the new participle resource, see Section 6.

2.2 The Standard Flat Analysis of Modifiers

The fact that German adjectival modifiers can generally undergo conversion into adverbs without overt morphological marking is a notorious problem for the syntactic analysis of German: there are no theoretically established tests to distinguish predicative adjectives and adverbials, see Geuder (2004). For this reason, the standard German tag set assigns a uniform tag ("ADJD") to modifiers that are morphologically ambiguous between an adjectival and adverbial reading. Moreover, in the German treebank TIGER (Brants et al., 2002) the resulting syntactic differences between the two readings are annotated by the same flat structure that does not disambiguate the sentence.

Despite certain theoretical problems related to the analysis of German modifiers, their interpretation in real corpus sentences is often unambiguous for native speakers. As an example, consider example (3) from the TIGER treebank. In the sentence, the participle *unterschrieben* (*signed*) clearly functions as a predicative modifier of the sentence's subject. The other, theoretically possible reading where the participle would modify the verb *send* is semantically not acceptable. However, in TIGER, the participle is analysed as an ADJD modifier attached under the VP node which is the general analysis for adjectival and adverbial modifiers.

(3) Die sollte **unterschrieben** an die Leitung
It should signed to the administration
zurückgesandt werden.
sent back be.
'It should be sent back signed to the administation.'

Sentence (4) (also taken from TIGER) illustrates the case of an adverbial participle. In this example, the reading where *angemessen* (*adequately*) modifies the main verb is the only one that is semantically plausible. In the treebank, the participle is tagged as ADJD and analysed as a modifier in the VP.

(4) Der menschliche Geist läßt sich rechnerisch nicht
The human mind lets itself computationally not
angemessen simulieren.
adequately simulate.
'The human mind cannot be adequately simulated in a computational way.'

The flat annotation strategy adopted for modifiers in the standard German tag set and in the treebank TIGER entails that instances of adverbs (and adverbial participles) cannot be extracted from automatically tagged, or parsed, text. Therefore, it would be very hard to obtain training material from German resources to train a system that automatically identifies adverbially used participles. However, the intuition corroborated by the examples presented in this section is that the structures can actually be disambiguated in many corpus sentences.

In the following sections, we show how we exploit parallel text to obtain training material for learning to predict occurences of adverbial participles, without any manual effort. Moreover, by means of this technique, we can substantially extend the grammatical resource for adverbial participles compared to the resource that can be directly extracted from the parallel text.

3 Participles in the Parallel Corpus

The intuition of the cross-lingual induction approach is that adverbial participles can easily be extracted from parallel corpora since in other languages (such as English or French) adverbs are often morphologically marked and easily labelled by statistical PoS taggers. As an example, consider sentence (5) extracted from Europarl, where the German participle *verstärkt* is translated by an English adverb (*increasingly*).

(5) a. Nicht ohne Grund sprechen wir **verstärkt**
Not without reason speak we increasingly
vom Europa der Regionen.
of a Europe of the Regions.
b. It is not without reason that we **increasingly** speak in terms of a Europe of the Regions.

The idea is to project specific morphological information about adverbs which is overt in languages like English onto German where adverbs cannot be directly extracted from tagged data. While this idea might seem intuitively straightfor-

ward, we also know that translation pairs in parallel data are not always lingusitically parallel, and as a consequence, word-alignment is not always reliable. To assess the impact of non-parallelism in adverbial translations of German participles, we manually annotated a sample of 300 translations. This data also constitutes the basis for the experiments reported in Section 4.

3.1 Data

Our experiments are based on the same data as in (Zarrieß et al., 2010). For convenience, we provide a short description here.

We limit our investigations to non-lexicalised participles occuring in the Europarl corpus and not yet recorded as adverbs in the lexicon of the German LFG grammar (5054 participle types in total). Given the participle candidates, we extract the set of sentences that exhibit a word alignment between a German participle and an English, French or Dutch adverb. The word alignments have been obtained with GIZA++. The extraction yields 27784 German-English sentence pairs considering all alignment links, and 5191 sentence pairs considering only bidirectional alignments between a participle and an English adverb.

3.2 Systematic Non-Parallelism

For data exploration and evaluation, we annotated 300 participle alignments out of the 5191 German-English sentences (with a bidirectional participle-adverb alignment). We distinguish the following annotation categories: (i) parallel translation, adverb information can be projected, (ii) incorrect alignment, (iii) correct alignment, but translation is a multi-word expression, (iv) correct alignment, but translation is a paraphrase (possibly involving a translation shift).

Parallel Cases In our annotated sample of English adverb - German participle pairs, 43%[1] of the translation instances are parallel in the sense that the overt adverb information from the English side can be projected onto the German participle. This means that if we base the induction technique on word-alignments alone, its precision would be relatively low.

Non-Parallel Cases Taking a closer look at the non-parallel cases in our sample (57% of the translation pairs), we find that 47% of this set are due to incorrect word alignments. The remaining 53% thus reflect regular cases of non-parallel translations. A typical configuration which makes up 30% of the the non-parallel cases is exemplified in (6) where the German main verb *vorlegen* is translated by the English multiword expression *put forward*.

(6) a. Wir haben eine Reihe von Vorschlägen **vorgelegt**.
b. We have **put forward** a number of proposals.

An example for the general paraphrase or translation shift category is given in Sentence (7). Here, the translational correspondence between *gekommen* (*arrived*) and the adverb *now* is due to language-specific, idiomatic realisations of an identical underlying semantic concept. The paraphrase translations make up 23% of the non-parallel cases in the annotated sample.

(7) a. Die Zeit ist noch nicht **gekommen** .
That time is yet not arrived.
b. That time is not **now** .

Furthermore, it is noticeable that the cross-lingual approach seems to inherently factor out the ambiguity between predicative and adverbial participles. In our annotated sample, there are no predicative participles that have been translated by an English adverb.

3.3 Filtering Mechanisms

The data analysis in the previous section, showing only 43% of parallel cases in English adverb translations for German participles, mainly confirms other studies in annotation projection which find that translational correspondences only allow for projection of linguistic analyses in a more or less limited proportion (Yarowsky et al., 2001; Hwa et al., 2005; Mihalcea et al., 2007).

In previous studies on annotation projection, quite distinct filtering methods have been proposed: in Yarowsky et al. (2001), projection errors are mainly attributed to word alignment errors and filtered based on translation probabilities.

[1] The diverging figures we report in Zarrieß et al. (2010) were due to a small bug in the script and it does not affect the overall interpretation of the data.

Hwa et al. (2005) find that errors in the projection of syntactic relations are also due to systematic grammatical divergences between languages and propose correcting these errors by means of specific, manually designed filters. Bouma et al. (2008) make similar observations to Hwa et al. (2005), but try to replace manual correction rules by filters from additional languages.

In Zarrieß et al. (2010), we compared a number of filtering techniques on our participle data. The 300 annotated translation instances are used as a test set for evaluation. In particular, we have established that a combination of syntactic dependency-based filters and multilingual filters can very accurately separate non-parallel translations from parallel ones where the adverb information can be projected. In Section 4, we show that these filtering techniques are also very useful for removing noise from the training material that we use to build a classifier.

4 Bootstrapping a German Participle Classifier from Crosslingual Data

In the previous section, we have seen that German adverbial participles can be easily found in crosslingual text by looking at their translations in a language that morphologically marks adverbials. In previous work, we exploited this observation by directly extracting types of adverbial participles based on word alignment links and the filtering mechanisms mentioned in Section 3. However, this method is very closely tied to data in the parallel corpus, which only comprises around 5000 participle-adverb translations in total, which results in 46 types of adverbial participles after filtering. Thus, we have no means of telling whether we would discover new types of adverbial participles in other corpora, from different domains to Europarl. As this corpus is rather small and genre specific, it even seems very likely that one could find additional adverbial participles in a bigger corpus. Moreover, we cannot be sure that certain adverbial participles have systematically diverging translations in other languages, due to crosslingual lexicalisation differences. Generally, it is not clear whether we have learned something general about the syntactic phenomenon of adverbial participles in German or whether we have just extracted a small, corpus-dependent subset of the class of adverbial participles.

In this section, we use instances of adverbially translated participles as training material for a classifier that learns to predict adverbial participles based on their monolingual syntactic context. Thus, we exploit the translations in the parallel corpus as a means of obtaining "annotated" or disambiguated training data without any manual effort. During training, we only consider the monolingual context of the participle, such that the final application of the classifier is not dependent on cross-lingual data anymore.

4.1 Context-based Identification of Adverbial Participles

Given the general linguistic problems related to adverbial participles (see Section 2), one could assume that it is very difficult to identify them in a given context. To assess the general difficulty of this syntactic problem, we run a first experiment comparing a grammar-based identification method against a classifier that only considers relatively narrow morpho-syntactic context. For evaluation, we use the 300 annotated participle instances described in Section 3. This test set divides into 172 negative instances, i.e. non-adverbial participles, and 128 positive instances. We report accuracy of the identification method, as well as precision and recall relating to the number of correctly predicted adverbial participles.

For the grammar-based identification, we use the German LFG which integrates the lexical resource for adverbial participles established in (Zarrieß et al., 2010). We parse the 300 Europarl sentences and check whether the most probable parse proposed by the grammar analyses the respective participle as an adverb or not. The grammar obtains a complete parse for 199 sentences out of the test set and we only consider these in the evaluation. The results are given in Table 1.

The high precision and accuracy of the grammar-based identification of adverbial participles suggests that in a lot of sentences, the adverbial analysis is the only possible reading, i.e. the only analysis that makes the sentence grammatical. But of course, we have substantially restricted the adverb participle-conversion in the grammar,

Training Data	Precision	Recall	Accuracy
Grammar	97.3	90.12	94.97
Classifier Unigram	87.10	84.38	87.92
Classifier Bigram	88.28	88.28	89.93
Classifier Trigram	89.60	87.5	90.27

Table 1: Evaluation on 300 participle instances from Europarl

so that it does not propose adverbial analyses for participles that are very unlikely to function as modifiers of verbs.

For the classifier-based identification, we use the adverbially translated participle tokens in our Europarl data (5191 tokens in total) as training material. We remove the 300 test instances from this training set, and then divide it into a set of positive and negative instances. To do this, we use the filtering mechanisms already proposed in Zarrieß et al. (2010). These filters apply on the type level, such that we first identify the positive types (46 total) and then use all instances of these types in the 4891 sentences as positive instances of adverbial participles (1978 instances). The remaining sentences are used as negative instances.

For the training of the classifier, we use maximum-entropy classification, which is also commonly used for the general task of tagging (Ratnaparkhi, 1996). In particular, we use the open source TADM tool for parameter estimation (Malouf, 2002). The tags of the words surrounding the participles are used as features in the classification task. We explore different sizes of the context window, where the trigram window is the most succesful (see Table 1). Beyond the trigram window, the results of the classifier start decreasing again, probably because of too many misleading features. Generally, this experiment shows that the grammar-based identification is more precise, but that the classifier still performs surprisingly well. Compared to the results from the grammar-based identification, the high accuracy of the classifier suggests that even the narrow syntactic contexts of adverbial vs. non-adverbial participles are quite distinct.

4.2 Designing Training Data for Participle Classification

There are several questions related to the design of the training data that we use to build our classifier. First, it is not clear how many negative instances are helpful for learning the adverbial - non-adverbial distinction. In the above experiment, we simply use the instances that do not pass the cross-lingual filters. In this section, we experiment with an augmented set of negative instances that was also obtained by extracting German participle that are bi-directionally aligned to an English participle in Europarl. This is based on the assumption that these participles are very likely to be verbal. Second, it is not clear whether we really need the filtering mechanisms proposed in Zarrieß et al. (2010) and whether we could improve the classifier by training it on a larger set of positive instances. Therefore, we also experiment with two further sets of positive instances: one where we used all participles (not necessarily bidirectionally) aligned to an adverb, one where we only use the bidirectional alignments. The results obtained for the different sizes of positive and negative instance sets are given in Table 2.

The picture that emerges from the results in Table 2 is very clear: the stricter the filtering of the training material (i.e. the positive instances) is, the better the performance of the classifier. The fact that we (potentially) loose certain positive instances in the filtering does not negatively impact on the classifier which substantially benefits from the fact that noise gets removed. Moreover, we find that if the training material is appropriately filtered, adding further negative instances does not help improving the accuracy. By contrast, if we train on a noisy set of positive instances, the classifier benefits from a larger set of negative instances. However, the positive effect that we get from augmenting the non-filtered training data is still weaker than the positive effect we get from the filtering.

5 Induction of Adverbial Participles on Monolingual Data

Given the classifier from Section 4 that predicts the syntactic category of a participle instance

Training Data	Pos. Instances	Neg. Instances	Precision	Recall	Accuracy
Non-Filtered Instances (all alignments)	27.184	10.000	43.10	100	43.10
Non-Filtered Instances (all alignments)	27.184	50.000	74.38	92.97	83.22
Non-Filtered Instances (symm. alignments)	4891	10.000	78.08	89.06	84.56
Non-Filtered Instances (symm. alignments)	4891	50.000	82.31	83.59	85.23
Filtered Instances	1978	10.000	**91.60**	**85.16**	**90.27**
Filtered Instances	1978	50.000	90.83	77.34	86.91

Table 2: Evaluation on 300 participle instances from Europarl

based on its monolingual syntactic context, we can now detect new instances or types of adverbial participles in any PoS-tagged German corpus. In this section, we investigate whether the classifier can be used to augment the resource of adverbial participles directly induced from Europarl with new types.

5.1 Data Extraction

We run our extraction experiment on the Huge German Corpus (HGC), a corpus of 200 million words of newspaper and other text. This corpus has been tagged with TreeTagger (Schmid, 1994). For each of the 5054 participle candidates, we extract all instances from the HGC which have not been tagged as finite verbs (at most 2000 tokens per participle). For each participle token, we also extract its syntactic context in terms of the 3 preceding and the 3 following tags. For classification, we use only those participles that have more than 50 instances in the corpus (2953 types).

In contrast to the cross-lingual filtering mechanisms developed in Zarrieß et al. (2010) which operate on the type-level, the classifier makes a prediction for every token of a given participle candidate. Thus, for each of the participle candidates, we obtain a percentage of instances that have been classified as adverbs. As we would expect, the percentage of adverbial instances is very low for most of the participles in our candidate set: for 75% of the 2953 types, the percentage is below 5%. This result confirms our initial intuition that the property of being used as an adverb is strongly lexically restricted to a certain class of participles.

5.2 Evaluation

Since we know that the classifier has an accuracy of 90% on the Europarl data, we only consider participles as candidates for adverbs where the classifier predicted more than 14% adverbial instances. This leaves us with a set of 210 participles, which comprises 13 of the original 46 participles extracted from Europarl, meaning we have discovered 197 new adverbial participle types.

We performed a manual evaluation of 50 randomly selected types out of the set of 197 new participle types. Therefore, we looked at the instances and their context which the classifier predicted to be adverbial. If there was at least one adverbial instance among these, the participle type was evaluated as correctly annotated by the classifier. By this means, we find that 76% of the participles were correctly classified.

This evaluation suggests that the accuracy of our classifier which we trained and tested on Europarl data is lower on the HGC data. The reason for this drop in performance will be explained in the following Section 5.3. However, assuming an accuracy of 76%, we have discovered 150 new types of adverbial participles. We argue that this is a very satisfactory result given that we have not invested any manual effort into the annotation or extraction of adverbial participles. This results also makes clear that the previous resource we induced on Europarl data, comprising only 46 participle types, was a very limited one.

5.3 Error Analysis

Taking a closer look at the 12 participle candidates that the classifier incorrectly labels as adverbial, we observe that their adverbially classified instances are mostly instances of a predicative use. This means that our Europarl training data does not contain enough evidence to learn the distinction between adverbial and predicative participles. This is not surprising since the set of negative instances used for training the classifier mainly comprises verbal instances of participles. Moreover, the syntactic contexts and constructions in which some predicatives and adverbials are used

Grammar	Prec.	Rec.	F-Sc.	Time in sec
46 Part-Adv	84.12	78.2	81.05	665
243 Part-Adv	84.12	77.67	80.76	665

Table 3: Evaluation on 371 TIGER sentences

Grammar	Parsed Sent.	Starred Sent.	Time-outs	Time in sec
No Part-Adv	665	315	64	3033
46 Part-Adv	710	269	65	3118
243 Part-Adv	767	208	69	3151

Table 4: Performance on the specialised TIGER test set (1044 sentences)

are very similar. Thus, in future work, we will have to include more data on predicatives (which is more difficult to obtain) and analyse the syntactic contexts in more detail.

6 Assessing the Impact of Resource Coverage on Grammar-based Parsing

In this section, we evaluate the classifier-based induction of adverbial participles from a grammar-based perspective. We integrate the entire set of induced adverbial participles (46 from Europarl and 197 from the HGC) into the German LFG grammar. As a consequence, the grammar allows the adverb conversion for 243 lexical participle types. We use the evaluation methodolgy explained in Section 2.

First, we conduct an accuracy-oriented evaluation on the standard TIGER test set. We compare against the German LFG that only integrates the small participle resource from Europarl. The results are given in Table 3. The difference between the 46 Part-Adv and 243 Part-Adv resource is not statistically signficant. Thus, the larger participle resource has no overall negative effect on the parsing performance. As established by an automatic upperbound evaluation in Zarrieß et al. (2010), we cannot not expect to find a positive effect in this evaluation because the phenomenon does not occur in the standard test set.

To show that the augmented resource indeed improves the coverage of the grammar, we built a specialised testsuite of 1044 TIGER sentences that contain an instance of a participle from the resource. Since this testsuite comprises sentences from the training set, we can only report a coverage-oriented evaluation here, see Table 4. The 243 Part-Adv increases the coverage by 8% on the specialised testsuite.

Moreover, we manually evaluated 20 sentences covered by the 243-Part-Adv grammar and not by 46-Part-Adv as to whether they contain a correctly analysed adverbial participle. In two sentences, the grammar obtained an adverbial analysis for clearly predicative modifiers, based on the enlarged resource. In three different sentences, it was difficult to decide whether the participle acts as an adverb or a predicative. In the remaining 15 sentences, the grammar established the the correct analysis of a clearly adverbially used participle.

7 Conclusion

We have proposed a cross-lingual induction method to automatically obtain data on adverbial participles in German. We exploited this cross-lingual data as training material for a classifier that learns to predict the syntactic category of a participle from its monolingual syntactic context. Since this category is usually not annotated in German resources and hard to describe in theory, the finding that adverbial participles can be predicted relatively precisely is of general interest for theoretic and computational approaches to the syntactic analysis of German.

We showed that, in order to obtain an accurate participle classifier, the quality of the training material induced from the parallel corpus is of crucial importance. By applying the filtering techniques from Zarrieß et al. (2010), the accuracy of the classifier increases between 5% and 7%. In future work, we plan to include more data on predicative participles to learn a more accurate distinction between predicative and adverbial participles.

Finally, we used the participle classifier to extract a lexical resource of adverbial participles for the German LFG grammar. In comparison to the relatively small resource of 46 types that can be directly induced from Europarl, we discovered a large number of new participle types (197 types in total). In a parsing experiment, we showed that this much bigger resource does not negatively impact on parsing performance and improves grammar coverage.

References

Bouma, Gerlof, Jonas Kuhn, Bettina Schrader, and Kathrin Spreyer. 2008. Parallel LFG Grammars on Parallel Corpora: A Base for Practical Triangulation. In Butt, Miriam and Tracy Holloway King, editors, *Proceedings of the LFG08 Conference*, pages 169–189, Sydney, Australia. CSLI Publications, Stanford.

Brants, Sabine, Stefanie Dipper, Silvia Hansen, Wolfgang Lezius, and George Smith. 2002. The TIGER Treebank. In *Proceedings of the Workshop on Treebanks and Linguistic Theories*.

Butt, Miriam, Helge Dyvik, Tracy Holloway King, Hiroshi Masuichi, and Christian Rohrer. 2002. The Parallel Grammar Project.

Geuder, Wilhelm. 2004. Depictives and transparent adverbs. In Austin, J. R., S. Engelbrecht, and G. Rauh, editors, *Adverbials. The Interplay of Meaning, Context, and Syntactic Structure*, pages 131–166. Benjamins.

Hwa, Rebecca, Philip Resnik, Amy Weinberg, Clara Cabezas, and Okan Kolak. 2005. Bootstrapping parsers via syntactic projection across parallel texts. *Nat. Lang. Eng.*, 11(3):311–325.

Malouf, Robert. 2002. A comparison of algorithms for maximum entropy parameter estimation. In *Proceedings of the Sixth Conference on Natural Language Learning (CoNLL-2002)*, pages 49–55.

Mihalcea, Rada, Carmen Banea, and Jan Wiebe. 2007. Learning multilingual subjective language via cross-lingual projections. In *Proceedings of the Association for Computational Linguistics (ACL 2007)*, pages 976–983, Prague.

Padó, Sebastian and Mirella Lapata. 2009. Cross-lingual annotation projection of semantic roles. *Journal of Artificial Intelligence Research*, 36:307–340.

Ratnaparkhi, Adwait. 1996. A maximum entropy model for part-of-speech tagging. In *Proceedings of EMNLP 96*, pages 133–142.

Riezler, Stefan, Tracy Holloway King, Ronald M. Kaplan, Richard Crouch, John T. Maxwell, and Mark Johnson. 2002. Parsing the Wall Street Journal using a Lexical-Functional Grammar and Discriminative Estimation Techniques . In *Proceedings of ACL 2002*.

Rohrer, Christian and Martin Forst. 2006. Improving coverage and parsing quality of a large-scale LFG for German. In *Proceedings of LREC-2006*.

Schmid, Helmut. 1994. Probabilistic part-of-speech tagging using decision trees. In *Proceedings of International Conference on New Methods in Language Processing*.

Yarowsky, David, Grace Ngai, and Richard Wicentowski. 2001. Inducing multilingual text analysis tools via robust projection across aligned corpora. In *Proceedings of HLT 2001, First International Conference on Human Language Technology Research*.

Zarrieß, Sina, Aoife Cahill, Jonas Kuhn, and Christian Rohrer. 2010. A Cross-Lingual Induction Technique for German Adverbial Participles. In *Proceedings of the 2010 Workshop on NLP and Linguistics: Finding the Common Ground, ACL 2010*, pages 34–42, Uppsala, Sweden.

Fusion of Multiple Features and Ranking SVM for Web-based English-Chinese OOV Term Translation

Yuejie Zhang, Yang Wang, Lei Cen, Yanxia Su, Cheng Jin, Xiangyang Xue
School of Computer Science, Shanghai Key Laboratory of Intelligent Information Processing, Fudan University
{yjzhang,072021176,082024072, 09210240074,jc,xyxue}@fudan.edu.cn

Jianping Fan
Department of Computer Science, The University of North Carolina at Charlotte
jfan@uncc.edu

Abstract

This paper focuses on the Web-based English-Chinese OOV term translation pattern, and emphasizes particularly on the translation selection strategy based on the fusion of multiple features and the ranking mechanism based on Ranking Support Vector Machine (Ranking SVM). By utilizing the CoNLL2003 corpus for the English Named Entity Recognition (NER) task and selected new terms, the experiments based on different data sources show the consistent results. Our OOV term translation model can "*filter*" the most possible translation candidates with better ability. From the experimental results for combining our OOV term translation model with English-Chinese Cross-Language Information Retrieval (CLIR) on the data sets of Text Retrieval Evaluation Conference (TREC), it can be found that the obvious performance improvement for both query translation and retrieval can also be obtained.

1 Introduction

In Cross-Language Information Retrieval (CLIR), most of users' queries are generally composed of short terms, in which there are many Out-of-Vocabulary (OOV) terms like Named Entities (NEs), new words, terminologies and so on. The translation quality of OOV term directly influences the precision of querying relevant multilingual information. Therefore, OOV term translation has become a very important and challenging issue in CLIR.

With the increasing growth of Web information which includes multilingual hypertext resources with abundant topics, it appears that Web information can mitigate the problem of the restricted OOV term translation accuracy (Lu and Chien, 2002). However, how to select the correct translations from Web information and locate the appropriate translation resources rapidly is still the main goal for OOV term translation. Hence, finding the effective feature representation and the optimal ranking pattern for translation candidates is the core part for the Web-based OOV term translation.

This paper focuses on the Web-based English-Chinese OOV term translation pattern, and emphasizes particularly on the translation selection strategy based on the fusion of multiple features and the translation ranking mechanism based on Ranking Support Vector Machine (Ranking SVM). By utilizing the CoNLL2003 corpus for the English Named Entity Recognition (NER) task and manually selected new terms in various fields, the established OOV term translation model can "*filter*" the most possible translation candidates with better ability. This paper also attempts to apply the OOV term translation mechanism above in English-Chinese CLIR. It can be observed from the experimental results on the data sets of Text Retrieval Evaluation Conference (TREC) that the obvious performance improvement for query translation can be obtained, which is very beneficial to CLIR and can improve the whole retrieval performance.

2 Related Work

At present, the methods for OOV term translation have changed from the basic pattern based on bilingual dictionary, transliteration or parallel corpus to the intermediate pattern based on comparable corpus (Lee et al., 2006; Shao and Ng, 2004; Virga and Khudanpur, 2003), and

Coling 2008: Poster Volume, pages 1435–1443,
Beijing, August 2010

then become a new pattern based on Web mining (Fang et al., 2006; Sproat et al., 2006).

In recent years, many researchers have utilized Web to find the translation candidates on webpages (Wu and Chang, 2007). Al-Onaizan and Knight (2002) used Web statistics information to validate the translation candidates generated by language model, and obtained the accuracy of 72.6% in Arabic-English OOV word translation. Lu and Chien (2004) utilized the statistics information about the anchor texts in Web search results to recognize the translation candidates, and got the accuracy of 63.6% in English-Chinese title query term translation. Zhang and Vines (2004) extracted the translation candidates for OOV query terms in CLIR from Web, and improved the performance of English-Chinese/Chinese-English CLIR to some extent. Zhang et al. (2005) searched the translation candidates by using cross-language query expansion and Web, and obtained the Top-1 accuracy of 81.0% in Chinese-English OOV word translation. Chen and Chen (2006) used the combination of Web statistics and the vocabulary, and acquired the Top-1 accuracy of 87.6% in Chinese-English OOV word translation. Jiang et al. (2007) utilized the combination of Web mining, transliteration and ranking based on Maximum Entropy (ME), only focused on English-Chinese person name translation and got the Top-1 accuracy of 47.5%.

Although the methods above can improve the translation performance for OOV term to a certain degree, there are still three common problems in the OOV term translation based on Web mining. (1) **Chinese key term extraction pattern from Web documents is over complex and the complexity is always higher.** Because of the inherent property of having no segmentation delimitation in Chinese, it's very difficult for English-Chinese OOV term translation to extract Chinese key terms from Web documents. The cost for the extraction computation is generally overlarge (Wang et al., 2004; Zhang and Vines, 2004). (2) **The feature information for the evaluation of translation candidates is not enough and comprehensive.** Most of OOV term translation methods implement the evaluation for candidates through mining simple local and Boolean features, that is, inherent features in candidates and their surrounding context features. However, if only a certain Web document that an OOV term appears is explored, the global information contained in the whole Web document set will be ignored, and the inconsistency and polysemy of candidates cannot be considered. (3) **The relevance measurement for translation pairs is very simple, or the computation cost is too high.** For ranking candidates, most of OOV term translation approaches adopt the simple combination computation of the feature values used, or get assessment based on classification models. Hence, the feature weights are determined according to the corresponding induction and suitable for some specific fields, but cannot guarantee the accuracy of the final translation ranking results. However, the Ranking SVM model can effectively express multiple ranking constraints, and has better universality and applicability (Cao et al., 2006; Joachimes, 2002; Vapnik, 1995).

3 Our Solutions

To support more precise English-Chinese OOV term translation, we establish a multiple-feature-based translation pattern based on Web mining and Ranking SVM. On the one hand, a Chinese key term extraction strategy is built on the simplified extraction computation for PAT-Tree, in which the optimization processing for the confidence of word building is improved to a certain extent. On the other hand, translation candidates are chosen by the fusion of multiple features. The representation forms of local, global and Boolean feature are constructed under the consideration of the complex characteristics of English/Chinese OOV term and Web information. Moreover, for the relevance measurement between an OOV term to be translated and its translation candidates, the supervised learning based on Ranking SVM is introduced to rank candidates precisely.

At first, given an OOV term to be translated as a query, it is input into the Google search engine to acquire the returned webpage snippet set. Next, Chinese key terms are extracted from the PAT-Tree built on the snippet set to determine the translation candidates. Subsequently, local, global and Boolean features are extracted from the candidates based on the fusion of multiple features. Finally, the candidates are filtered and ranked through the supervised learning based on Ranking SVM.

4 Chinese Key Term Extraction

In Web mining of English-Chinese OOV term translation, an important problem is to extract the target translation candidates from the returned Chinese Web documents, which can be considered as a key term extraction task.

The PAT-Tree structure is an efficient indexing method in both IR and Information Extraction (IE) domains (Chien, 1997; Gonnet et al., 1992). Its superior feature is the Semi Infinite String, which can store all the strings from the whole corpus (i.e., the returned snippet set in this paper) in a binary tree. The branch node indicates the search direction and the leaf node stores the index and frequency for a string.

Generally, a Chinese character corresponds to a binary-coded form with 2 bytes (16 bits). Chinese strings can be transformed into binary strings. There is an ending tag for each string and its binary form is "00000000". Take "中文信息抽取" (*Chinese IE*) and "信息检索" (*IR*) as an example, the binary strings for them are described in Figure 1. Thus a PAT-Tree can be built based on these strings, as shown in Figure 2. The branch node stands for the comparison bit (Comp-bit), which represents the position of different bit in binary strings. Some binary strings have the value of 0 in such a bit and are classified into the left branch, while others have 1 and turn to the right branch.

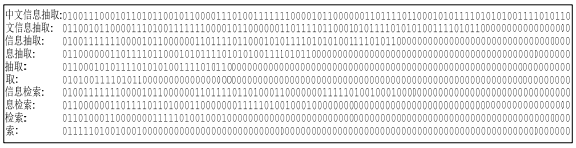


Figure 1. Binary string representation instantiation.

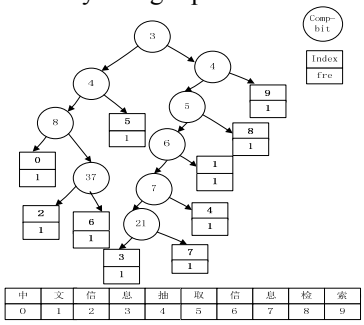


Figure 2. PAT-Tree Instantiation for Figure 1.

In the extraction process, the PAT-tree is traversed first, and the branch nodes with the Comp-bit values larger than 32 are selected. This is because the minimum length of a Chinese common string is 2 characters and each has 16 binary bits. Next, the frequency values of both two child nodes are added as the frequency of the common string (i.e., the parent branch node). At last, the common strings with the frequency values larger than 2 are extracted as the key terms. For the PAT-Tree in Figure 2, there is a branch node with the Comp-bit value of 37, which indicates that at least the prefixes of two strings contain two identical characters. It can be known from the leaf nodes that two strings are "信息抽取" (*IE*) and "信息检索" (*IR*). Hence, the prefix substring "信息" (*information*) with the frequency of 2 is extracted as the common string. Thus the key terms with the arbitrary lengths and frequency values can be retrieved from the built PAT-Tree.

However, with the common strings being extracted, large amounts of noisy terms and fragments are also extracted. To filter noisy fragments, Wang et al. (2004) used SPDCD and the Local-Maxima algorithm, but the computation cost was too expensive. Therefore, the simplified filtering manner is adopted here:

$$\alpha(c_i \cdots c_j) = \frac{f(c_i \cdots c_j) - f(c_1 \cdots c_n)}{f(c_i \cdots c_j)} \qquad (1)$$

where $c_1 \ldots c_n$ is a n-gram that contains the substring $c_i \ldots c_j$; $c_i \ldots c_j$ is the n-1-gram to be estimated, i.e., $c_i \ldots c_j = c_1 \ldots c_{n-1}$ or $c_i \ldots c_j = c_2 \ldots c_n$; $f(\ )$ denotes the string frequency; α represents the cohesion factor of the n-1-gram string, that is, the ability of independent word building. The closer to 1 the value of α is, the more possible meaningful key term $c_i \ldots c_j$ is.

5 Multiple Feature Representation

Local Feature (LF) is constructed based on neighboring tokens and the token itself. There are two types of contextual information to be considered when extracting LFs, namely internal lexical and external contextual information.

(1) **Term length** (*Len*) – Aims to consider the length of the translation candidate.

(2) **Phonetic Value** (*PV*) – Aims to investigate the phonetic similarity between an OOV term and its translation candidates. Because the associated syllabification representations can often be found between Chinese and English syllables with fewer ambiguities, the syllabification has become an effective channel in phonetic feature expression. *PV* means that for measuring the edit distance similarity between the syllabification sequences of an OOV term

and its candidates, the processing is executed according to the specific linguistic rules.

$$PV(S_{OOV}, T_{OOV}) = 1 - \frac{EditDist(S_{OOV}', T_{OOV}')}{Len(S_{OOV}') + Len(T_{OOV}')} \quad (2)$$

where S_{OOV} and T_{OOV} denote the OOV term in the source language and its translation candidate in the target language respectively, S_{OOV}' and T_{OOV}' are the character strings after the syllabification and removing the vowels, *EditDist*(,) indicates the edit distance between two strings, and *Len*( ) is the string length.

(3) **Length Ratio of OOV Term and Its Translation Candidate** (*LR*) – Aims to explore the composition possibility that the extracted key term can be regarded as the translation for an OOV term. An OOV term and its translation should have the similar length, so the *LR* value is close to 1 as possible. A Chinese term is segmented into significant pieces first, and the number of pieces is taken as its length. For example, "非典型肺炎" (*SARS*) is segmented into "非" (*non*), "典型" (*typical*) and "肺炎" (*pneumonia*), and its length is 3. For an English term, the number of words is counted as the length. If there is only one word composed of capital letters, its length is defined as the number of letters, e.g., "*SARS*" has the length of 4. Thus the *LR* value of "*SARS*" and its candidate "非典型肺炎" is 4/3=1.3.

(4) **Phonetic and Semantic Integration Feature** (*P&S_IF*) – Aims to consider the phonetic information and senses of an OOV term and its candidates synthetically. It is set up for multi-word OOV terms, especially for NEs and new terms. Each constituent can be translated by the phonetic information or senses.

$$P\&S\_IF(S_{OOV}, T_{OOV}) = \frac{LScore(S_{OOV}, T_{OOV}) + PV(S_{OOV}'', T_{OOV}'')}{LScore(S_{OOV}, T_{OOV}) + 1} \quad (3)$$

where *LScore*(,) is the matching word number of non-transliteration words in S_{OOV} and T_{OOV}, while S_{OOV}'' and T_{OOV}'' are the remaining strings of S_{OOV} and T_{OOV} after computing *LScore*. For example, given S_{OOV} "*Capitoline Museum*" and its T_{OOV} "卡比多里尼博物馆" (*Capitoline Museum*), the non-transliteration words "*Museum*" and "博物馆" (*museum*) are matched, then *LScore*(S_{OOV}, T_{OOV})=1; the *PV* value between the remaining strings "*Capitoline*" and "卡比多里尼" (*Capitoline*) is 0.8, so the final *P&S_IF* value is 1.8/2=0.9.

Global Feature (GF) is extracted from other occurrences of the same or similar tokens in the Web document set. The common case in the Web-based OOV term translation is that the translation candidates in the previous parts of Web documents will often occur with the same or similar forms in the latter parts. The contextual information from the same and other Web documents may play an important role in determining the final translation. To utilize such global information, GFs are constructed based on the characteristics of Web documents.

(1) **Global Term Frequency** (*G_Freq*) – Aims to utilize the frequency information that an OOV term and its translation candidates appear in the Web document set. It is always the most important feature and includes four parameters. $\boldsymbol{Freq_{Soov}}$ denotes the frequency of S_{OOV} in all the returned webpage snippets. $\boldsymbol{TF_{Toov}}$ indicates the number of T_{OOV}s in all the snippets. $\boldsymbol{DF_{Toov}}$ represents the number of snippets that contain T_{OOV}. ***CO_Freq*** means the number of snippets that contain both S_{OOV} and T_{OOV}, i.e, co-occurrence frequency.

(2) **Chi-Square** (χ^2) **Feature Value** (*CV*) – Aims to evaluate the semantic similarity between an OOV term and its translation candidates by their occurrence in Web documents.

$$CV_{\chi^2}(S_{OOV}, T_{OOV}) = \frac{N \times (a \times d - b \times c)^2}{(a+b) \times (a+c) \times (b+d) \times (c+d)} \quad (4)$$

where a is the number of snippets that contain both S_{OOV} and T_{OOV}, b is the number of snippets that contain S_{OOV} but do not contain T_{OOV}, c is the number of snippets that do not contain S_{OOV} but contain T_{OOV}, d is the number of snippets that do not contain neither of S_{OOV} and T_{OOV}, and $N=a+b+c+d$.

(3) **Co-occurrence Distance** (*CO_Dist*) – Aims to investigate the distance between an OOV term and its candidates in Web documents. This distance is often very closer.

For each snippet that contains both S_{OOV} and T_{OOV}, three positions are considered, that is, the first position that S_{OOV} and T_{OOV} appear ($p1$), the second position ($p2$) and the last one ($p3$). In the following snippet, S_{OOV} is "*AARP*" and T_{OOV} is "美国退休者协会" (*America Association of Retired Persons*, *AARP*).

拿什么创造《AARP杂志》的成功--传媒--人民网
2008年10月18日 ... 它是一个协会杂志，隶属于美国退休者协会（AARP）。《AARP杂志》自称是"世界上发行量 ... 笔者认为，《AARP杂志》的成功，很大程度上得益于以下几点。 ...
media.people.com.cn/GB/22114/45733/136325/8192992.html - 38k - 网页快照 - 类似网页

$p1_{SOOV}$=6, $p2_{SOOV}$=62, $p3_{SOOV}$=97;
$p1_{TOOV}$=54, $p2_{TOOV}$=-1, $p3_{TOOV}$=54.

The position is indexed from 0 and $p2_{TOOV}$=-1 means only one candidate exists in the snippet. Then the nearest position pair $p2_{SOOV}$ and $p1_{TOOV}$ can be found for this example. The distance *Dist* between S_{OOV} and T_{OOV} is computed as:

$$Dist(S_{OOV}, T_{OOV}) = \begin{cases} pi_{S_{OOV}} - pj_{T_{OOV}} - Len(T_{OOV}), & pi_{S_{OOV}} > pj_{T_{OOV}} \\ pj_{T_{OOV}} - pi_{S_{OOV}} - Len(SOOV), & pi_{S_{OOV}} < pj_{T_{OOV}} \end{cases} \quad (5)$$

Given the example above, $Dist=p2_{SOOV}-p1_{TOOV}-7=62-54-7=1$, that is, S_{OOV} and T_{OOV} are a left bracket '(' apart. Finally, the average distance *CO-Dist* in the snippet set can be computed as:

$$CO\_Dist(S_{OOV}, T_{OOV}) = AVG\_Dist(S_{OOV}, T_{OOV}) = \frac{Sum(Dist)}{CO\_Freq(S_{OOV}, T_{OOV})} \quad (6)$$

where *Sum*( ) is the sum of *Dist* in each snippet. (4) **Rank Value** (*RV*) – Aims to consider the rank for translation candidates in the Web document set. It includes five parameters. ***Top_Rank*** (*T_Rank*) is the rank of the snippet that first contains T_{OOV} and given by the search engine. ***Average_Rank*** (*A_Rank*) is the average position of T_{OOV} in the returned snippets.

$$A\_Rank(T_{OOV}) = \frac{Sum(Rank)}{DF_{T_{OOV}}(T_{OOV})} \quad (7)$$

where *Sum*( ) denotes the rank sum of each snippet. ***Simple_Rank*** (*S_Rank*) is computed as $S\_Rank(T_{OOV})=TF_{TOOV}(T_{OOV})*Len(T_{OOV})$, which aims at investigating the impact of the frequency and length of T_{OOV} on ranking. ***R_Rank*** is utilized as a comparison basis.

$$R\_Rank(T_{OOV}) = \beta \times \frac{|T_{OOV}|}{MAX\_WL} + (1-\beta) \times \frac{TF_{T_{OOV}}(T_{OOV})}{Freq_{S_{OOV}}(S_{OOV})} \quad (8)$$

where β is set as 0.25 empirically, $|T_{OOV}|$ is the length of T_{OOV}, and *MAX_WL* denotes the maximum length of candidate terms. ***DF_Rank*** (*D_Rank*) is similar to *S_Rank* and computed as $D\_Rank(T_{OOV})=DF_{TOOV}(T_{OOV})*Len(T_{OOV})$.

Boolean Feature (BF) is a binary feature and equivalent to a heuristic rule designed for the particular relationship between an OOV term and its translation candidates. BFs are used to explore the different occurrence forms with higher possibility for the translation candidates in Web documents. (1) **Position Distance with OOV Term** (*PD_S*$_{OOV}$) – If T_{OOV} occurs close to S_{OOV} (within 10 characters), then this feature is set as 1, else -1. (2) **Neighbor Relationship with OOV Term** (*NR_S*$_{OOV}$) – If T_{OOV} occurs prior or next to S_{OOV}, then this feature is set as 1. (3) **Bracket Neighbor Relationship with OOV Term** (*BNR_S*$_{OOV}$) – If T_{OOV} locates prior or next to S_{OOV} and occurs with the form "T_{OOV} (S_{OOV})" or "S_{OOV} (T_{OOV})", then this feature is set as 1. (4) **Special Mark Word** (*SMW*) – This is an intuitive feature. Within a certain co-occurrence distance (usually less than 10 characters) between an OOV term and its candidates, if there is such a term like "全称" (*full name*), "叫" (*be named as*), "译为" (*be translated as* …), "名称" (*name*), or "(或/又)称为" ((*or*/*also*) *be called as* …), or within 5 characters if there are some punctuations like "( )", "[ ]" and "（）", then this feature is set as 1.

6 Ranking based on Ranking SVM

For the OOV term translation based on Web mining, another difficulty is how to evaluate the relevance between an OOV term and its translation candidates, that is, how to rank the translation candidates from "*best*" to "*worst*".

The candidate ranking can be regarded as a binary classification problem. However, usually only highly related fragments of OOV terms can be found, rather than their correct translations. Instead of regarding the candidate ranking as binary classification, it is solved as an Ordinal Regression problem. Ranking SVM maps different objects into a certain kind of order relation. The key is modeling the judgements for user's preferences, and then the constraint relations for ranking can be derived (Herbrich et al., 1999; Xu et al., 2005).

For a given OOV term S_{OOV}, if there are two translation candidates T_{OOVi} and T_{OOVj}, the preference judgement can be formulated as $T_{OOVi} \succ_{SOOV} T_{OOVj}$. Thus more training samples are constructed, which contain multiple constraint features. The preference judgement can be transformed into the feature function as:

$$f(w, T_{OOVi}, S_{OOV}) \succ_{SOOV} f(w, T_{OOVj}, S_{OOV}) \quad (9)$$

where w is a parameter and represented as a n-dimensional vector $w=\{w_1, w_2, \ldots, w_n\}$. This feature function can also be expressed as:

$$f(w, T_{OOV}, S_{OOV}) = \sum_{k=1}^{p} w_k LF_k(T_{OOV}, S_{OOV}) + \sum_{l=p+1}^{q} w_l GF_l(T_{OOV}, S_{OOV}) + \sum_{m=q+1}^{n} w_m BF_m(T_{OOV}, S_{OOV}) \quad (10)$$

where LF_k(,), GF_l(,) and BF_m(,) are the local, global and Boolean feature representation respectively. These three kinds of feature representation are incorporated as a whole and represented as a feature function family with the multi-dimensional feature vector in (11).

$$f(w, T_{OOV}, S_{OOV}) = w \cdot h(T_{OOV}, S_{OOV}) \quad (11)$$

That is the ranking results for candidates. Thus the relevance for each feature vector x (translation candidate) containing a group of features can be evaluated through Ranking SVM.

7 Experiment and Analysis

7.1 Data Set and Evaluation Metrics

For the performance evaluation, 4,593 English NEs are selected from the English corpus of the NER task in CoNLL2003. The test set contains 446 Person Names (PRNs), 329 Location Names (LCNs) and 455 Organization Names (OGNs), and the remaining is taken as the training set (including 1,137 PRNs, 1,152 LCNs and 1,074 OGNs) through manually tagging. Additionally, 300 English new terms are chosen randomly from 9 categories, including movie name, book title, brand name, terminology, idiom, rare animal name, rare PRN and OGN. Such terms are used to investigate the generalization ability of our model.

Top-N-Inclusion-Rate is used as a measurement for the translation performance. For a set of OOV terms to be translated, its *Top-N-Inclusion-Rate* is defined as the percentage of the OOV terms whose translations could be found in the first N extracted translations.

7.2 Experiment on Parameter Setting

For Chinese key term extraction, the test on the threshold α is performed. As shown in Figure 3, when the lower bound of α is set as 0.4, the best performance can be achieved.

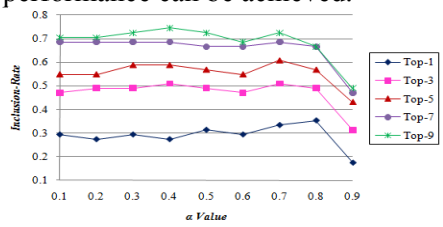


Figure 3. Results for α value setting.

To get the most relevant candidates into top-10 before the final ranking, an initial ranking test is performed on *S_Rank*, *R_Rank* and *D_Rank*. It can be seen from Figure 4 that *D_Rank* exhibits the better performance.

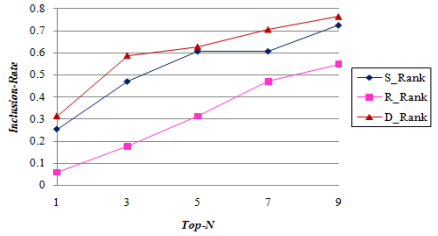


Figure 4. Results for initial ranking manner.

To find how many returned webpage snippets are suitable for the translation acquisition, the test on the snippet number is performed. As shown in Figure 5, the best performance can be obtained by using 200 snippets.

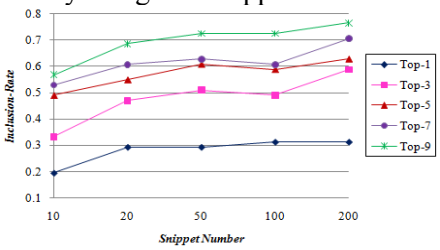


Figure 5. Results for webpage snippet number.

7.3 Experiment on Multiple Feature Fusion

To verify the effectiveness for multiple feature fusion, the test on the feature combination for OOV term translation is implemented. As shown in Table 1, the highest accuracy (the percentage of the correct translations in all the extracted translations) of 83.1367% can be acquired by using all the features.

Feature				Accuracy	Reduction
All Features				83.1367%	—
Numerical Feature	*Local Numerical Feature*	*-Len*		81.7355%	-1.4012%
		-PV		77.4494%	-5.6873%
		-LR		81.4231%	-1.7136%
		-P&S_IF		79.9002%	-3.2365%
	Global Numerical Feature	*Global Frequency*	*-TF_{Toov}*	82.9877%	-0.1490%
			-DF_{Toov}	83.2112%	+0.0745%
			-CO_Freq	83.0870%	-0.0497%
		-CV		82.3125%	-0.8242%
		-CO_Dist		81.8577%	-1.2790%
		RV	*-T_Rank*	83.0125%	-0.1242%
Boolean Feature		*-PD_S_{OOV}*		82.1806%	-0.9561%
		-NR_S_{OOV}		82.2923%	-0.8444%
		-BNR_S_{OOV}		80.7525%	-2.3842%
		-SMW		83.1740%	+0.0373%

Table 1. Results for feature combination.

In Table 1, '-' before the specific feature denotes that the OOV term is translated by combining all the other features except this feature; "*Reduction*" represents the difference value between the translation accuracy obtained by using all the features and that by removing a specific feature. The positive "*Reduction*" indicates that the accuracy is improved after removing a specific feature, while the negative shows the accuracy is decreased.

It can be seen from Table 1 that for mining the translations for OOV terms, the most important three features are *PV*, *P&S_IF* and *BNR*, then *LR*, *Len* and *CO_Dist*. As for the frequency feature, its contribution is limited, because many translation candidates with higher *PV* or *P&S_IF* values are the terms with low frequency. It shows that *PV* and *P&S_IF* play a very crucial role in mining the translation candidates with low frequency. In addition,

the contribution degree of *CV* is also positive. However, when training based on only the features that are beneficial to the whole translation performance, the best translation accuracy is 83.1243%, which is worse than that by combining all the features. From a view of the effect of the single feature on the whole translation performance, some features may have slightly negative impact. Nevertheless, through combining all the features, the multiple feature fusion mechanism can indeed efficiently improve the translation accuracy.

7.4 Experiment on OOV Term Translation

Some translation examples based on different ranking patterns are given in Table 2, in which the score represents the correlation degree between the translation pair. The closer to -1 the score is, the more irrelevant the translation pair is; while the closer to 3 the score is, the more relevant the translation pair is.

PRN -- "*Santamaria*"		
Candidates (Top-5)	**SVM Score**	**Ranking SVM Score**
桑塔马利雅	1.1746	3.17754
辛达马利亚	0.7087	2.81014
桑塔玛利亚	0.9326	2.68914
圣何塞	0.2879	2.26468
蒙哥山塔马利亚	0.2051	2.1525
LCN -- "*Gettysburg National Military Park*"		
Candidates (Top-5)	**SVM Score**	**Ranking SVM Score**
葛底斯堡国家军事公园	0.7500	2.4998
堡国家军事公园	0.6666	2.4159
国家军事公园	0.3973	1.8539
盖茨堡国家军事公园	0.2877	1.5172
在葛底斯堡建立了国家军事公园	-0.3407	0.8019
OGN -- "*Federal Reserve Board*"		
Candidates (Top-5)	**SVM Score**	**Ranking SVM Score**
美国联准会	0.9784	2.7435
美国联邦储备委员会	0.9483	2.7314
美国联邦准备制度	0.5387	2.7178
联邦储备金监察小组	1.2031	2.6684
联邦准备理事会	0.7425	2.6003

Table 2. OOV term translation examples.

Furthermore, Jiang et al. (2007) utilized the combination of Web mining, transliteration and ME-based ranking to implement English-Chinese PRN translation, which is very similar to our approach. To make a contrast with it, we accomplished this method on the same data set. The comparison results are shown in Table 3.

Ranking Pattern	**Category**	**Top-1**	**Top-2**	**Top-3**
***based on SVM* (*Multiple Features*)**	***PRN***	64.44%	85.07%	91.42%
	LCN	53.93%	73.33%	81.82%
	OGN	49.68%	70.70%	82.16%
	All	**56.10%**	**76.59%**	**85.45%**
***based on Ranking-SVM* (*Multiple Features*)**	***PRN***	77.14%	89.20%	93.96%
	LCN	64.24%	75.15%	85.45%
	OGN	63.05%	79.61%	89.17%
	All	**68.46%**	**81.87%**	**89.92%**
[Jiang et al., 2007] *based on ME* ($PV+CV+NR\_S_{OOV}+BNR\_S_{OOV}$)	***PRN* (Only)**	49.07%	57.33%	60.43%

Table 3. Performance comparison results.

From the experimental results above, it can be concluded that the ranking based on the supervised learning significantly outperforms the conventional ranking strategies, and Ranking SVM is superior to SVM and ME for translation candidate ranking. From the contrast between our model and Jiang's method, it can be found that our approach is superior to Jiang's and the better performance can be achieved based on the fusion of multiple features proposed in this paper. Meanwhile, it can also be observed from Table 3 that the performance for LCN and OGN translation is better, while the best performance is obtained for PRN translation. It shows that our translation model is sensitive to the category and the popularity degree of OOV term to some extent.

In order to test the translation performance for the other kinds of English OOV term, another test is performed based on the OOV new terms selected randomly from 9 categories. The experimental results are shown in Table 4.

Top-N-Inclusion-Rate	**Top-1**	**Top-3**	**Top-5**	**Top-7**	**Top-9**
Other OOV Terms	49.41%	71.02%	72.46%	81.51%	84.30%

Table 4. Results for other OOV terms.

Furthermore, the translations for some OOV terms based on different translation manners are compared, including our proposed model, Google Translate and the Live Trans translation model developed by WKD Lab at National Taiwan University, as shown in Table 5.

OOV Terms	**Translation from Our Model**	**Translation from Google Translate**	**Translation from Live Trans**
Forrest Gump	阿甘正传/电影	阿甘正传	阿甘正传/亚伦席维斯崔
Estee Lauder	雅诗兰黛/化妆品	雅诗兰黛	雅诗兰黛/香水/化妆品
Arteriosclerosis	动脉硬化	动脉粥样硬化	心脏/动脉硬化
Woman Pace-Setter	三八红旗手	女子的步伐/制定	三八红旗手
Dream of the Red Mansion	红楼/红楼梦	红楼梦	红楼梦/文章书目
SARS	非典型肺炎/非典	严重急性呼吸系统综合症	病毒/非典型肺炎
NASA	美国宇航局	美国航天局	美国太空总署

Table 5. Comparison for different translation manners.

The results above demonstrate that our model can be applicable to all kinds of OOV terms and has better translation performance.

7.5 Experiment on English-Chinese CLIR

To explore the applicability and usefulness of our OOV term translation model in English-Chinese CLIR, four CLIR runs based on *long query* (terms in both title and description fields) and *short query* (only terms in the title field) are carried out on the English topic set (25 topics) and Chinese corpus (127,938 documents) from TREC-9. (1) *E-C_LongCLIR*1 – using long query and the bilingual-dictionary-based query translation; (2) *E-C_LongCLIR*2 – using long query, the bilingual-dictionary-based

query translation and our OOV term translation; (3) *E-C_ShortCLIR*1 – using short query and the bilingual-dictionary-based query translation; (4) *E-C_ShortCLIR*2 – using short query, the bilingual-dictionary-based query translation and our OOV term translation. The Precision-Recall curves and Median Average Precision (MAP) values are shown in Figure 6.

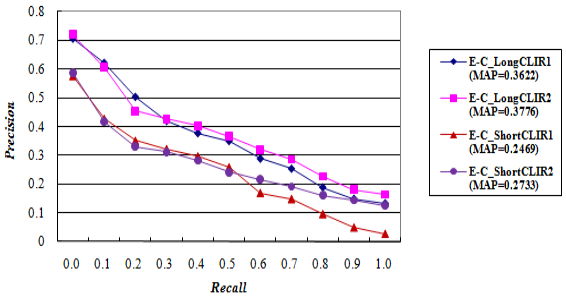


Figure 6. Results for English-Chinese CLIR combining our OOV term translation model.

It can be seen from Figure 6 that the best run is *E-C_LongCLIR*2, and its results exceed those by another run *E-C_LongCLIR*1 based on long query. By adopting both query translation based on bilingual dictionary and OOV term translation, the English-Chinese CLIR for long query has gained the significant improvement on the whole retrieval performance. Compared with the traditional query translation based on bilingual dictionary, such a combination manner is exactly a better way for query translation from the source language to the target language. Additionally, through comparing the results for the other two runs *E-C_ShortCLIR*1 and *E-C_ShortCLIR*2 based on short query, it can also be further confirmed that our OOV term translation mechanism can also support CLIR for short query effectively.

7.6 Analysis and Discussion

Through analyzing the results for translation extraction and ranking, it can be found that the translation quality is highly related to the following aspects. (1) **The translation results are associated with the search engine used, especially for some specific OOV terms.** For example, given an OOV term "*Cross-Strait Three-links*", the mining result based on Google in China is "两岸大三通", while some meaningless information is mined by Live Trans. (2) **Some terms are conventional terminologies and cannot be translated literally.** For example, "*Woman Pace-Setter*", a proper noun with the Chinese characteristic, should be translated into "三八红旗手", rather than "女子的步伐" (*women's pace*) or "制定" (*establishment*) given by Google Translate. (3) **The proposed model is sensitive to the notability degree of OOV term.** This phenomenon is the main reason why there is obvious difference among the translation performance for PRN, LCN and OGN. (4) **There is a "*fragment effect*" in PAT-Tree-based Chinese key term extraction.** The fragments of Chinese terms have become the main noisy data. Such a problem should be solved by setting the specific threshold for additional features like heuristic rules and occurrence distance. (5) **Word Sense Disambiguation (WSD) should be added to improve the translation performance.** Although most of OOV terms have a unique semantic definition, there are still a few OOV terms with ambiguity, e.g., "*AARP*" (*American Association of Retired Persons* or *AppleTalk Address Resolution Protocol*). (6) **The ranking pattern based on the supervised learning is able to synthesize various feature representations for translation candidates.** Thus the rank for a candidate can be precisely predicted through tagging and training.

8 Conclusions

In this paper, the proposed model improves the acquirement ability for OOV term translation through Web mining, and solves the translation pair selection and evaluation in a novel way by fusing multiple features and introducing the supervised learning based on Ranking SVM. Furthermore, it is significant to apply the key techniques in machine translation into OOV term translation, such as OOV term recognition, statistical machine learning, alignment of sentence and phoneme, and WSD. All these aspects will be our research focus in the future.

Acknowledgements

This work is supported by National Natural Science Fund of China (No. 60773124), Shanghai Natural Science Fund (No. 09ZR1403000), National Science and Technology Pillar Program of China (No. 2007BAH09B03), 973 Program of China (No. 2010CB327906), Shanghai Municipal R&D Foundation (No. 08dz1500109) and Shanghai Key Laboratory of Intelligent Information Processing. Cheng Jin from Fudan University is the corresponding author.

References

Y. Al-Onaizan, K. Knight. 2002. *Translating Named Entities using Monolingual and Bilingual Resources*. In: The 30th Meeting of the Association for Computational Linguistics (ACL 2002), 400-408.

Y.B. Cao, J. Xu, T.Y. Liu, H. Li, Y.L. Huang, and H.W. Hon. 2006. *Adapting Ranking-SVM to Document Retrieval*. In: The 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2006), 186-193.

C. Chen, H.H. Chen. 2006. *A High-Accurate Chinese-English NE Backward Translation System Combining Both Lexical Information and Web Statistics*. In: The Joint Conference of the International Committee on Computational Linguistics and the Association for Computational Linguistics (COLING-ACL 2006), 81-88.

L.F. Chien. 1997. *PAT-Tree-based Keyword Extraction for Chinese Information Retrieval*. In: The 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 1997), 50-58.

G.L. Fang, H. Yu, and F. Nishino. 2006. *Chinese-English Term Translation Mining Based on Semantic Prediction*. In: The Joint Conference of the International Committee on Computational Linguistics and the Association for Computational Linguistics (COLING-ACL 2006), 199-206.

G.H. Gonnet, R.A. Baeza-Yates, and T. Sinder. 1992. *New Indices for Text: PAT Trees and PAT Arrays*. Information Retrieval Data Structures & Algorithms, 66-82.

R. Herbrich, T. Graepel, and K. Obermayer. 1999. *Support Vector Learning for Ordinal Regression*. In: The 9th International Conference on Neural Networks (ICANN 1999), 97-102.

L. Jiang, M. Zhou, L.F. Chien, and C. Niu. 2007. *Named Entity Translation with Web Mining and Transliteration*. In: The 20th International Joint Conference on Artificial Intelligence (IJCAI 2007), 1629-1634.

T. Joachimes. 2002. *Optimizing Search Engines using Click through Data*. In: The 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD 2002), 133-142.

C.J. Lee, J.S. Chang, and J.R. Jang. 2006. *Alignment of Bilingual Named Entities in Parallel Corpora Using Statistical Models and Multiple Knowledge Sources*. ACM Transactions on Asian Language Processing, 5(2):121-145.

W.H. Lu, L.F. Chien. 2002. *Translation of Web Queries using Anchor Text Mining*. ACM Transactions on Asian Language Information Processing, 1(2):159-172.

W.H. Lu, L.F. Chien. 2004. *Anchor Text Mining for Translation of Web Queries: A Transitive Translation Approach*. ACM Transactions on Information Systems, 22(2):242-269.

L. Shao, H.T. Ng. 2004. *Mining New Word Translations from Comparable Corpora*. In: The 20th International Conference on Computational Linguistics (COLING 2004), 618-624.

R. Sproat, T. Tao, and C.X. Zhai. 2006. *Named Entity Transliteration with Comparable Corpora*. In: The Joint Conference of the International Committee on Computational Linguistics and the Association for Computational Linguistics (COLING-ACL 2006), 73-80.

V.N. Vapnik. 1995. *The Nature of Statistical Learning Theory*. Springer-Verlag New York, Inc., New York, NY.

P. Virga, S. Khudanpur. 2003. *Transliteration of Proper Names in Cross-Language Applications*. In: The 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2003), 365-366.

J.H. Wang, J.W. Teng, P.J. Cheng, W.H. Lu, and L.F. Chien. 2004. *Translating Unknown Cross-Lingual Queries in Digital Libraries using a Web-based Approach*. In: The Joint Conference on Digital Libraries (JCDL 2004), 108-116.

J.C. Wu, J.S. Chang. 2007. *Learning to Find English to Chinese Transliterations on the Web*. In: The Joint Meeting of the Conference on Empirical Methods in Natural Language Processing and the Conference on Computational Natural Language Learning (EMNLP-CoNLL 2007), 996-1004.

J. Xu, Y.B. Cao, H. Li, and M. Zhao. 2005. *Ranking Definitions with Supervised Learning Methods*. In: The 14th International World Wide Web Conference (WWW 2005), 811-819.

Y. Zhang, P. Vines. 2004. *Using the Web for Automated Translation Extraction in Cross-Language Information Retrieval*. In: The 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2004), 162-169.

Y. Zhang, P. Vines. 2004. *Detection and Translation of OOV Terms Prior to Query Time*. In: The 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2004), 524-525.

Y. Zhang, F. Huang, and S. Vogel. 2005. *Mining Translations of OOV Terms from the Web through Cross-Lingual Query Expansion*. In: The 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2005), 669-670.

Machine Transliteration: Leveraging on Third Languages

Min Zhang **Xiangyu Duan** **Vladimir Pervouchine** **Haizhou Li**
Institute for Infocomm Research, A-STAR
{mzhang, xduan, vpervouchine, hli}@i2r.a-star.edu.sg

Abstract

This paper presents two pivot strategies for statistical machine transliteration, namely *system-based* pivot strategy and *model-based* pivot strategy. Given two independent source-pivot and pivot-target name pair corpora, the *model-based* strategy learns a direct source-target transliteration model while the *system-based* strategy learns a source-pivot model and a pivot-target model, respectively. Experimental results on benchmark data show that the *system-based* pivot strategy is effective in reducing the high resource requirement of training corpus for low-density language pairs while the *model-based* pivot strategy performs worse than the *system-based* one.

1 Introduction

Many technical terms and proper names, such as personal, location and organization names, are translated from one language into another language with approximate phonetic equivalents. This phonetic translation using computer is referred to as machine transliteration. With the rapid growth of the Internet data and the dramatic changes in the user demographics especially among the non-English speaking parts of the world, machine transliteration play a crucial role in most multilingual NLP, MT and CLIR applications (Hermjakob *et al.*, 2008; Mandl and Womser-Hacker, 2004). This is because proper names account for the majority of OOV issues and translation lexicons (even derived from large parallel corpora) usually fail to provide good coverage over diverse, dynamically increasing names across languages.

Much research effort has been done to address the transliteration issue in the research community (Knight and Graehl, 1998; Wan and Verspoor, 1998; Kang and Choi, 2000; Meng *et al.*, 2001; Al-Onaizan and Knight, 2002; Gao *et al.*, 2004; Klementiev and Roth, 2006; Sproat, 2006; Zelenko and Aone, 2006; Li *et al.*, 2004, 2009a, 2009b; Sherif and Kondrak, 2007; Bertoldi *et al.*, 2008; Goldwasser and Roth, 2008). These previous work can be categorized into three classes, i.e., grapheme-based, phoneme-based and hybrid methods. Grapheme-based method (Li *et al.*, 2004) treats transliteration as a direct orthographic mapping process and only uses orthography-related features while phoneme-based method (Knight and Graehl, 1998) treats transliteration as a phonetic mapping issue, converting source grapheme to source phoneme followed by a mapping from source phoneme to target phoneme/grapheme. Hybrid method in machine transliteration refers to the combination of several different models or decoders via re-ranking their outputs. The report of the first machine transliteration shared task (Li *et al.*, 2009a, 2009b) provides benchmarking data in diverse language pairs and systemically summarizes and compares different transliteration methods and systems using the benchmarking data.

Although promising results have been reported, one of major issues is that the state-of-the-art machine transliteration approaches rely heavily on significant source-target parallel name pair corpus to learn transliteration model. However, such corpora are not always availa-

Coling 2008: Poster Volume, pages 1444–1452,
Beijing, August 2010

ble and the amounts of the current available corpora, even for language pairs with English involved, are far from enough for training, letting alone many low-density language pairs. Indeed, transliteration corpora for most language pairs without English involved are unavailable and usually rather expensive to manually construct. However, to our knowledge, almost no previous work touches this issue.

To address the above issue, this paper presents two pivot language-based transliteration strategies for low-density language pairs. The first one is *system*-based strategy (Khapra *et al.*, 2010), which learns a source-pivot model from source-pivot data and a pivot-target model from pivot-target data, respectively. In decoding, it first transliterates a source name to *N*-best pivot names and then transliterates each pivot names to target names which are finally re-ranked using the combined two individual model scores. The second one is *model*-based strategy. It learns a direct source-target transliteration model from two independent[1] source-pivot and pivot-target name pair corpora, and then does direct source-target transliteration. We verify the proposed methods using the benchmarking data released by the NEWS2009[2] (Li *et al.*, 2009a, 2009b). Experiential results show that without relying on any source-target parallel data the system-based pivot strategy performs quite well while the model-based strategy is less effective in capturing the phonetic equivalent information.

The remainder of the paper is organized as follows. Section 2 introduces the baseline method. Section 3 discusses the two pivot language-based transliteration strategies. Experimental results are reported at section 4. Finally, we conclude the paper in section 5.

2 The Transliteration Model

Our study is targeted to be language-independent so that it can be applied to different language pairs without any adaptation effort. To achieve this goal, we use joint source-channel model (JSCM, also named as *n*-gram transliteration model) (Li *et la.*, 2004) under grapheme-based framework as our transliteration model due to its state-of-the-art performance by only using orthographical information (Li *et al.*, 2009a). In addition, unlike other feature-based methods, such as CRFs (Lafferty *et al.*, 2001), MaxEnt (Berger *et al.*, 1996) or SVM (Vapnik, 1995), the JSCM model directly computes model probabilities using maximum likelihood estimation (Dempster *et al.*, 1977). This property facilitates the implementation of the model-based strategy.

JSCM directly models how both source and target names can be generated simultaneously. Given a source name *S* and a target name *T*, it estimates the joint probability of *S* and *T* as follows:

$$\begin{aligned} P(S,T) &= P(s_1 \dots s_i \dots s_K, t_1 \dots t_i \dots t_K) \\ &= P(< s_1, t_1 >, \dots, < s_i, t_i >, \\ &\qquad \dots, < s_K, t_K >) \\ &= P(< s,t >_1, \dots, < s,t >_i, \\ &\qquad \dots < s,t >_K) \\ &= \prod_{k=1}^{K} P\left(< s,t >_k \mid < s,t >_1^{k-1}\right) \\ &\approx \prod_{k=1}^{K} P\left(< s,t >_k \mid < s,t >_{k-n+1}^{k-1}\right) \end{aligned}$$

where s_i and t_i is an aligned transliteration unit[3] pair, and *n* is the *n*-gram order.

In implementation, we compare different unsupervised transliteration alignment methods, including Giza++ (Och and Ney, 2003), the JSCM-based EM algorithm (Li *et al.*, 2004), the edit distance-based EM algorithm (Pervouchine *et al.*, 2009) and Oh et al.'s alignment tool (Oh *et al.*, 2009). Based on the aligned transliteration corpus, we simply learn the transliteration model using maximum likelihood estimation (Dempster *et al.*, 1977) and decode the transliteration result $T^* = \text{argmax}_T P(S,T)$ using stack decoder (Schwartz and Chow, 1990).

[1] Here "independent" means the source-pivot and pivot-target data are not derived from the same English name source.

[2] http://www.acl-ijcnlp-2009.org/workshops/NEWS2009/pages/sharedtask.html

[3] Transliteration unit is language dependent. It can be a Chinese character, a sub-string of English words, a Korean Hangual or a Japanese Kanji or several Japanese Katakanas.

3 Pivot Transliteration Strategies

3.1 System-based Strategy

The system-based strategy is first proposed by Khapra *et al.* (2010). They worked on system-based strategy together with CRF and did extensively empirical studies on Indic/Slavic/Semetic languages and English.

Given a source name *S*, a target name *T* and let *Z*(*S*, *Ź*) be the n-best transliterations of *S* in one or more pivot language *Ź* [4], the system-based transliteration strategy under JSCM can be formalized as follows:

$$\begin{aligned} P(S,T) &= \sum_{Ź} \sum_{Z(S,Ź)} P(S, Z(S,Ź), T) \\ &\approx \sum_{Z} P(S,Z,T) \\ &\approx \sum_{Z} P(T|S,Z) * P(S,Z) \end{aligned}$$

In the above formula, we assume that there is only one pivot language used in the derivation from the first line to the second line. Under the pivot transliteration framework, we can further simplify the above formula by assuming that T is independent of S when given Z. The assumption holds because the parallel name corpus between *S* and *T* is not available under the pivot transliteration framework. The n-best transliterations in pivot language are expected to be able to carry enough information of the source name *S* for translating *S* to target name *T*. Then, we have:

$$\begin{aligned} P(S,T) &= \sum_{Z} P(T|Z) * P(S,Z) \\ &= \sum_{Z} \frac{P(S,Z) * P(T,Z)}{P(Z)} \end{aligned} \quad (1)$$

Obviously we can train the two JSCMs of $P(S,Z)$ and $P(T,Z)$ using the two parallel corpora of (S,Z) and (T,Z), and train the language model $P(Z)$ using the monolingual corpus of Z. Following the nature of JSCM, Eq. (1) directly models how the source name *S* and pivot name *Z* and how the pivot name *Z* and the target name *T* are generated simultaneously. Since *Z* is considered twice in $P(S,Z)$ and $P(T,Z)$, the duplicated impact of *Z* is removed by dividing the model by $P(Z)$.

Given the model as described at Eq. (1), the decoder can be formulized as:

$$\begin{aligned} T^* &= \operatorname*{argmax}_T P(S,T) \\ &= \operatorname*{argmax}_T \left(\sum_Z \frac{P(S,Z) * P(T,Z)}{P(Z)} \right) \end{aligned} \quad (2)$$

If we consider multiple pivot languages, the modeling and decoding process are:

$$\begin{aligned} & P(S,T) \\ &= \sum_{Ź} \sum_{Z(S,Ź)} \left(\frac{P\left(S, Z(S,Ź)\right) * P(T, Z(S,Ź))}{P(Z(S,Ź))} \right) \end{aligned}$$

$$\begin{aligned} & T^* \\ &= \operatorname*{argmax}_T \left(\sum_{Ź} \sum_{Z(S,Ź)} \frac{P\left(S, Z(S,Ź)\right) * P\left(T, Z(S,Ź)\right)}{P\left(Z(S,Ź)\right)} \right) \end{aligned}$$

3.2 Model-based Strategy

Rather than combining the transitive transliteration results at system level, the model-based strategy aims to learn a direct model $P(S,T)$by combining the two individual models of $P(S,Z)$ and $P(T,Z)$, which are learned from the two parallel corpora of (S,Z) and (T,Z), respectively. Now let us use bigram as an example to illustrate how to learn the transliteration model $P(S,T) = \prod_{k=1}^{K} P(< s,t >_k \mid < s,t >_{k-1})$ using the model-based strategy.

$$\begin{aligned} & P(< s,t >_k \mid < s,t >_{k-1}) \\ &= \frac{P(< s,t >_k, < s,t >_{k-1})}{P(< s,t >_{k-1})} \end{aligned} \quad (3)$$

where,

$$\begin{aligned} & P(< s,t >_k, < s,t >_{k-1}) \\ &= P(s_k, s_{k-1}, t_k, t_{k-1}) \\ &= \sum_{z_k, z_{k-1}} P(s_k, s_{k-1}, t_k, t_{k-1}, z_k, z_{k-1}) \\ &= \sum_{z_k, z_{k-1}} P(t_k, t_{k-1} | s_k, s_{k-1}, z_k, z_{k-1}) \\ &\qquad * P(s_k, s_{k-1}, z_k, z_{k-1}) \end{aligned}$$

[4] There can be multiple pivot languages used in the two strategies. However, without loss of generality, we only use one pivot language to facilitate our discussion. It is very easy to extend one pivot language to multiple ones by considering all the pivot transliterations in all pivot languages.

The same as the system-based strategy, we can further simplify the above formula by assuming that T is independent of S when given Z. Indeed, $P(t_k, t_{k-1}|s_k, s_{k-1}, z_k, z_{k-1})$ cannot be estimated directly from training corpus. Then we have:

$$\begin{aligned} P(< s,t >_k, < s,t >_{k-1}) &= \sum_{z_k, z_{k-1}} P(t_k, t_{k-1}|s_k, s_{k-1}, z_k, z_{k-1}) \\ &\quad * P(s_k, s_{k-1}, z_k, z_{k-1}) \\ &\approx \sum_{z_k, z_{k-1}} P(t_k, t_{k-1}|z_k, z_{k-1}) \\ &\quad * P(s_k, s_{k-1}, z_k, z_{k-1}) \\ &\approx \sum_{z_k, z_{k-1}} P(t_k, t_{k-1}, z_k, z_{k-1}) \\ &\quad * P(s_k, s_{k-1}, z_k, z_{k-1}) \\ &\quad / P(z_k, z_{k-1}) \end{aligned} \quad (4)$$

where $P(t_i, t_{i-1}, z_i, z_{i-1})$, $P(s_i, s_{i-1}, z_i, z_{i-1})$ and $P(z_i, z_{i-1})$ can be directly learned from training corpus. $P(< s,t >_{k-1})$ for Eq (3) can also be estimatedas follows.

$$P(< s,t >_{k-1}) = \sum_{<s,t>_k} P(< s,t >_k, < s,t >_{k-1})$$

In summary, eq. (1) formulizes the system-based strategy and eq. (3), (4) and (5) formulize the model-based strategy, where we can find that they share the same nature of generating source, pivot and target names simultaneously. The difference is that the model-based strategy operates at fine-grained transliteration unit level.

3.3 Comparison with Previous Work

Almost all previous work on machine transliteration focuses on direct transliteration or transliteration system combination. There is only one recent work (Khapra *et al.*, 2010) touching this issue. They work on system-based strategy together with CRF. Compared with their work, this paper gives more formal definitions and derivations of system-based strategy from modeling and decoding viewpoints based on the JSCM model.

The pivot-based strategies at both system and model levels have been explored in machine translation. Bertoldi *et al.* (2008) studies two pivot approaches for phrase-based statistical machine translation. One is at system level and one is to re-construct source-target data and alignments through pivot data. Cohn and Lapata (2007) explores how to utilize multilingual parallel data (rather than pivot data) to improve translation performance. Wu and Wang (2007, 2009) extensively studies the model-level pivot approach and also explores how to leverage on rule-based translation results in pivot language to improve translation performance. Utiyama and Isahara (2007) compares different pivot approaches for phrase-based statistical machine translation. All of the previous work on machine translation works on phrase-based statistical machine translation. Therefore, their translation model is to calculate phrase-based conditional probabilities at unigram level ($P(t_k|s_k)$) while our transliteration model is to calculate joint transliteration unit-based conditional probabilities at bigram level ($P(< s,t >_k | < s,t >_{k-1})$).

4 Experimental Results

4.1 Experimental Settings

We use the NEWS 2009 benchmark data as our experimental data (Li *et al.*, 2009). The NEWS 2009 data includes 8 language pairs, where we select English to Chinese/Japanese /Korean data (E-C/J/K) and based on which we further construct Chinese to Japanese/Korean and Japanese to Korean for our data.

Language Pair	Training	Dev	Test
English-Chinese	31,961	2896	2896
English-Japanese	23,225	1492	1489
English-Korean	4,785	987	989
Chinese-Japanese	12,417	75	77
Chinese-Korean	2,148	32	31
Japanese-Korean	6,035	65	69

Table 1. Statistics on the data set

Table 1 reports the statistics of all the experimental data. To have a more accurate evaluation, the test sets have been cleaned up to make sure that there is no overlapping between any test set with any training set. In addition, the three E-C/J/K data are generated independently so that there is very small percentage of over-

lapping between them. This can ensures the evaluation of the pivot study fair and accurate.

We compare different alignment algorithms on the DEV set. Finally we use Pervouchine *et al.* (2009)'s alignment algorithm for Chinese-English/Japanese/Korean and Oh *et al.* (2009)'s alignment algorithm for English-Korean and Li *et al.* (2004)'s alignment algorithm for English-Japanese and Japanese-Korean. Given the aligned corpora, we directly learn each individual JSCM model (i.e., *n*-gram transliteration model) using SRILM toolkits (Stolcke, 2002). We also use SRILM toolkits to do decoding. For the system-based strategy, we output top-20 pivot transliteration results.

For the evaluation matrix, we mainly use top-1 accuracy (ACC) (Li *et al.*, 2009a) to measure transliteration performance. For reference purpose, we also report the performance using all the other evaluation matrixes used in NEWS 2009 benchmarking (Li *et al.*, 2009a), including F-score, MRR, MAP_ref, MAP_10 and MAP_sys. It is reported that F-score has less correlation with other matrixes (Li *et al.*, 2009a).

4.2 Experimental Results

4.2.1 Results of Direct Transliteration

Table 2 reports the performance of direct transliteration. The first three experiments (line 1-3) are part of the NEWS 2009 share tasks and the others are our additional experiments for our pivot studies.

Comparison of the first three experimental results and the results reported at NEWS 2009 shows that we achieve comparable performance with their best-reported systems at the same conditions of using single system and orthographic features only. This indicates that our baseline represents the state-of-the-art performance. In addition, we find that the *back*-transliteration (line 4-6) consistently performs worse than its corresponding *forward*-transliteration (line 1-3). This observation is consistent with what reported at previous work (Li *et al.*, 2004; Zhang *et al.*, 2004). The main reason is because English has much more transliteration units than foreign C/J/K languages. This makes the transliteration from English to C/J/K a many-to-few mapping issue and *back*-transliteration a few-to-many mapping issue. Therefore *back*-transliteration has more ambiguities and thus is more difficult.

Overall, the lower six experiments (line 7-12) shows worse performance than the upper six experiments which has English involved. This is mainly due to the less available training data for the language pairs without English involved. This observation motivates our study using pivot language for machine transliteration.

4.2.2 Results of System-based Strategy

Table 3 reports three empirical studies of system-based strategies: Japanese to Chinese through English, Chinese to Japanese through English and Chinese to Korean through English. Considering the fact that those language pairs with English involved have the most training data, we select English as pivot language in the system-based study. Table 3 clearly shows that:

- The system-based pivot strategy is very effective, achieving significant performance improvement over the direct transliteration by 0.09, 0.07 and 0.03 point of ACC in the three language pairs, respectively;
- Different from other pipeline methodologies, the system-based pivot strategy does not suffer heavily from the error propagation issue. Its ACC is significantly better than the product of the ACCs of the two individual systems;
- The combination of pivot system and direct system slightly improves overall ACC.

We then conduct more experiments to figure out the reasons. Our further statistics and analysis show the following reasons for the above observations:

The pivot approach is able to use source-pivot and pivot-target data whose amount is much more than that of the available direct source-target data.

- The nature of transliteration is phonetic translation. Therefore a little bit variation in orthography may not hurt or even help to improve transliteration performance in some cases as long as the orthographical variations keep the phonetic equivalent

Language Pairs	ACC	F-Score	MRR	MAP_ref	MAP_10	MAP_sys
English → Chinese	0.678867	0.871497	0.771563	0.678867	0.252382	0.252382
English → Japanese	0.482203	0.831983	0.594235	0.471766	0.201510	0.201510
English → Korean	0.439838	0.722365	0.543039	0.439585	0.171621	0.171621
Chinese → English	0.395250	0.867702	0.518292	0.372403	0.222787	0.222787
Japanese → English	0.334839	0.838212	0.450984	0.319277	0.168032	0.168032
Korean → English	0.088505	0.494205	0.109249	0.088759	0.034380	0.034380
Chinese → Japanese	0.385965	0.769245	0.473851	0.348319	0.159948	0.159948
Japanese → Chinese	0.402597	0.714193	0.491595	0.402597	0.165581	0.165581
Chinese → Korean	0.290323	0.571587	0.341129	0.290323	0.178652	0.178652
Korean → Chinese	0.129032	0.280645	0.156042	0.129032	0.048163	0.048163
Japanese → Korean	0.313433	0.678240	0.422862	0.313433	0.208310	0.208310
Korean → Japanese	0.089286	0.321617	0.143948	0.091270	0.049992	0.049992

Table 2. Performance of direct transliterations

Language Pairs	ACC	F-Score	MRR	MAP_ref	MAP_10	MAP_sys
Jap→Eng→Chi (**Pivot**)	0.493506	0.750711	0.617440	0.493506	0.195151	0.195151
Jap→Eng→Chi (**Pivot**) + Jap → Chi (**Direct**)	0.506494	0.753958	0.622851	0.506494	0.196017	0.196017
Jap → Chi (**Direct**)	0.402597	0.714193	0.491595	0.402597	0.165581	0.165581
Jap → Eng (**Direct**)	0.334839	0.838212	0.450984	0.319277	0.168032	0.168032
Eng → Chi (**Direct**)	0.678867	0.871497	0.771563	0.678867	0.252382	0.252382
Chi→Eng→Jap (**Pivot**)	0.456140	0.777494	0.536591	0.414961	0.183222	0.183222
Chi→Eng→Jap (**Pivot**) + Chi → Jap (**Direct**)	0.491228	0.801443	0.563297	0.450049	0.191742	0.191742
Chi → Jap (**Direct**)	0.385965	0.769245	0.473851	0.348319	0.159948	0.159948
Chi → Eng (**Direct**)	0.395250	0.867702	0.518292	0.372403	0.222787	0.222787
Eng → Jap (**Direct**)	0.482203	0.831983	0.594235	0.471766	0.201510	0.201510
Chi→Eng→Kor (**Pivot**)	0.322581	0.628146	0.432642	0.322581	0.175822	0.175822
Chi→Eng→Kor (**Pivot**) + Chi → Kor (**Direct**)	0.331631	0.632967	0.439143	0.334222	0.176543	0.176543
Chi → Kor (**Direct**)	0.290323	0.571587	0.341129	0.290323	0.178652	0.178652
Chi → Eng (**Direct**)	0.395250	0.867702	0.518292	0.372403	0.222787	0.222787
Eng → Kor (**Direct**)	0.439838	0.722365	0.543039	0.439585	0.171621	0.171621

Table 3. Performance comparison of system-based strategy on Jap (Japanese) to Chi (Chinese) and Chi (Chinese) to Jap (Japanese)/Kor (Korean) through Eng (English) as pivot language, where "…(**Pivot**) + …(**Direct**)" means that for the same language pair we merge and re-rank the pivot transliteration and direct transliteration results

information. Indeed, given one source English names, there are usually more than one correct transliteration references in Japanese/Korean. This case also happens to English to Chinese although not so heavy as in English to Japanese/Korean.

Language Pairs	ACC	F-Score	MRR	MAP_ref	MAP_10	MAP_sys
Chi→Eng→Jap (**Model-based Pivot: O**)	0.087719	0.538454	0.117446	0.085770	0.040645	0.040645
Chi→Eng→Jap (**Model-based Pivot: R**)	0.210526	0.746497	0.381210	0.201267	0.156106	0.156106
Chi→Eng→Jap (**System-based Pivot**)	0.456140	0.777494	0.536591	0.414961	0.183222	0.183222
Chi → Jap (**Direct**)	0.385965	0.769245	0.473851	0.348319	0.159948	0.159948
Jap→Chi→Eng (**Model-based Pivot**)	0.148504	0.724623	0.224253	0.141791	0.088966	0.088966
Jap→Chi→Eng (**System-based Pivot**)	0.201581	0.741627	0.266507	0.191926	0.098024	0.134730
Jap → Eng (**Direct**)	0.334839	0.838212	0.450984	0.319277	0.168032	0.168032
Eng→Jap→Kor (**Model-based Pivot**)	0.206269	0.547732	0.300641	0.206269	0.145882	0.145882
Eng→Jap→Kor (**System-based Pivot**)	0.315470	0.629640	0.404769	0.315723	0.167587	0.225892
Eng → Kor (**Direct**)	0.439838	0.722365	0.543039	0.439585	0.171621	0.171621

Table 4. Performance of Model-based Pivot Transliteration Strategy

- The N-best accuracy of machine transliteration (of both to and from English) is very high[5]. It means that in most cases the correct transliteration in pivot language can be found in the top-20 results and the other 19 results hold the similar pronunciations with the correct one, which can serve as alternative "quasi-correct" inputs to the second stage transliterations and thus largely improve the overall accuracy.

The above analysis holds when using English as pivot language. Now let us see the case of using non-English as pivot language. Table 4 reports two system-based strategies using Chinese and Japanese as pivot languages, where we can find that the performance of two system-based strategies is worse than that of the direct transliterations. The main reason is because that the direct transliteration utilizes much more training data than the pivot approach. However, the good thing is that the system-based pivot strategy using non-English as pivot language still does not suffer from error propagation issue. Its ACC is significantly better than the product of the ACCs of the two individual systems.

4.2.3 Results of Model-based Strategy

Table 4 reports the performance of model-based strategy. It clearly shows that the model-based strategy is less effective and performs much worse than both the system-based strategy and direct transliteration.

While the model-based strategy works well at phrase-based statistical machine translation (Wu and Wang, 2007, 2009), it does not work at machine transliteration. To investigate the reasons, we conduct many additional experiments and do statistics on the model and

[5] Both our studies and previous work (Li et al., 2004; Zhang et al., 2004) shows that the top-20 accuracy from English to J/K is more than 0.85 and more than 0.95 in English-Chinese case. The top-20 accuracy is a little worse from C/J/K to English, but still more than 0.7.

aligned training data[6]. From this in-depth analysis, we find that main reason is due to the fact that the model-based strategy introduces too many entries (ambiguities) to the final transliteration model. For example, in the Jap→Chi→Eng experiment, the unigram and bigram entries of the transliteration model obtained by the model-based strategy are 45 and 6.6 times larger than that of the transliteration model trained directly from parallel data. This is not surprising. Given a transliteration unit in pivot language, it can generate $m * n$ source-to-target transliteration unit mappings (unigram entry of the model), where m is the number of the source units that can be mapped to the pivot unit and n is the number of the target units that can be mapped from the pivot unit.

Besides the ambiguities introduced by the large amount of entries in the model, another reason that leads to the worse performance of model-based strategy is the size inconsistence of transliteration unit of pivot language. As shown at Table 4, we conduct three experiments. In the first experiment (Chi→Eng→Jap), we use English as pivot language. We find that the English transliteration unit size in Chi→Eng model is much larger than that in Eng→Jap model. This is because from phonetic viewpoint, in Chi→Eng model, the English unit is at syllable level (corresponding one Chinese character) while in Eng→Jap model, the English unit is at sub-syllable level (consonant or vowel or syllable, corresponding one Japanese Katakana). This is the reason why we conduct two model-based experiments for Chi→Eng→Jap. One is based on the original alignments (**Model-based Pivot: O**) and one is based on the reconstructed alignments[7] (**Model-based Pivot: R**). Experimental results clearly show that the reconstruction improves performance significantly. In the second and third experiments (Jap→Chi→Eng, Eng→Jap→Kor), we use Chinese and Japanese as pivot languages. Therefore we do not need to re-construct transliteration units and alignments. However, the performance is still very poor. This is due to the first reason of the large amount of ambiguities.

The above two reasons (ambiguities and transliteration unit inconsistence) are mixed together, leading to the worse performance of the model-based strategy. We believe that the fundamental reason is because the pivot transliteration unit is too small to be able to convey enough phonetic information of source language to target language and thus generates too many alignments and ambiguities.

5 Conclusions

A big challenge to statistical-based machine transliteration is the lack of the training data, esp. to those language pairs without English involved. To address this issue, inspired by the research in the SMT research community, we study two pivot transliteration methods. One is at system level while another one is at model level. We conduct extensive experiments using NEW 2009 benchmarking data. Experimental results show that system-based method is very effective in capturing the phonetic information of source language. It not only avoids successfully the error propagation issue, but also further boosts the transliteration performance by generating more alternative pivot results as the inputs of the second stage. In contrast, the model-based method in its current form fails to convey enough phonetic information from source language to target language.

For the future work, we plan to study how to improve the model-based strategy by pruning out the so-called "bad" transliteration unit pairs and re-sampling the so-called "good" unit pairs for better model parameters. In addition, we also would like to explore other pivot-based transliteration methods, such as constructing source-target training data through pivot languages.

References

Yaser Al-Onaizan and Kevin Knight. 2002. *Translating named entities using monolingual and bilingual resources*. ACL-02

Adam L. Berger, Stephen A. Della Pietra and Vincent J. Della Pietra. 1996. *A Maximum Entropy Approach to Natural Language Processing*. Computational Linguistics. 22(1):39–71

[6] However, due to space limitation, we are not allowed to report the details of those experiments.

[7] Based on the English transliteration units obtained from Chi →Eng, we reconstruct the English transliteration units and alignments in Eng→Jap by merging the adjacent units of both English and Japanese to syllable level.

N. Bertoldi, M. Barbaiant, M. Federico and R. Cattoni. 2008. *Phrase-based Statistical Machine Translation with Pivot Languages*. IWSLT-08

Trevor Cohn and Mirella Lapata. 2007. *Machine Translation by Triangulation: Making Effective Use of Multi-Parallel Corpora*. ACL-07

Andrew Finch and Eiichiro Sumita. 2008. *Phrase-based machine transliteration*. IJCNLP-08

Wei Gao, Kam-Fai Wong and Wai Lam. 2004. *Phoneme-based Transliteration of Foreign Names for OOV Problems*. IJCLNP-04

Dan Goldwasser and Dan Roth. 2008. *Transliteration as constrained optimization*. EMNLP-08

A.P. Dempster, N.M. Laird, D.B.Rubin.1977. *Maximum likelihood from incomplete data via the EM algorithm*, J. Roy. Stat. Soc., Ser. B. Vol. 39

Ulf Hermjakob, K. Knight and Hal Daum ́e. 2008. *Name translation in statistical machine translation: Learning when to transliterate*. ACL-08

John Lafferty, Fernando Pereira, Andrew McCallum. 2001. *Conditional random fields: Probabilistic models for segmenting and labeling sequence data*. ICML-01

B.J. Kang and Key-Sun Choi. 2000. *Automatic Transliteration and Back-transliteration by Decision Tree Learning*. LREC-00

Mitesh Khapra, Kumaran A and Pushpak Bhattacharyya. 2010. *Everybody loves a rich cousin: An empirical study of transliteration through bridge languages.* NAACL-HLT-10

Alexandre Klementiev and Dan Roth. 2006. *Weakly supervised named entity transliteration and discovery from multilingual comparable corpora*. COLING-ACL-06

K. Knight and J. Graehl. 1998. *Machine Transliteration*, Computational Linguistics, Vol 24, No. 4

P. Koehn, F. J. Och and D. Marcu. 2003. *Statistical phrase-based translation*. HLT-NAACL-03

J. Lafferty, A. McCallum and F. Pereira. 2001. *Conditional random fields: Probabilistic models for segmenting and labeling sequence data*. ICML-01

Haizhou Li, A Kumaran, Vladimir Pervouchine and Min Zhang. 2009a. *Report of NEWS 2009 Machine Transliteration Shared Task*. IJCNLP-ACL-09 Workshop: NEWS-09

Haizhou Li, A Kumaran, Min Zhang and Vladimir Pervouchine. 2009b. *Whitepaper of NEWS 2009 Machine Transliteration Shared Task*. IJCNLP-ACL-09 Workshop: NEWS-09

Haizhou Li, Ming Zhang and Jian Su. 2004. *A Joint Source-Channel Model for Machine Transliteration*. ACL-04

Thomas Mandl and Christa Womser-Hacker. 2004. *How do Named Entities Contribute to Retrieval Effectiveness?* CLEF-04

Helen M. Meng, Wai-Kit Lo, Berlin Chen and Karen Tang. 2001. *Generate Phonetic Cognates to Handle Name Entities in English-Chinese cross-language spoken document retrieval*. ASRU-01

Jong-Hoon Oh, Kiyotaka Uchimoto, and k. Torisawa. 2009. *Machine Transliteration with Target-Language Grapheme and Phoneme: Multi-Engine Transliteration Approach.* NEWS 2009

Franz Josef Och and Hermann Ney. 2003. *A Systematic Comparison of Various Statistical Alignment Models*. Computational Linguistics 29(1)

V. Pervouchine, H. Li and B. Lin. 2009. *Transliteration Alignment*. ACL-IJCNLP-09

R. Schwartz and Y. L. Chow. 1990. *The N-best algorithm: An efficient and exact procedure for finding the N most likely sentence hypothesis*, ICASSP-90

Tarek Sherif and Grzegorz Kondrak. 2007. *Substring-based transliteration*. ACL-07

Richard Sproat, Tao Tao and ChengXiang Zhai. 2006. *Named entity transliteration with comparable corpora*. COLING-ACL-06

Andreas Stolcke. 2002. *SRILM - an extensible language modeling toolkit.* ICSLP-02

Masao Utiyama and Hitoshi Isahara. 2007. *A Comparison of Pivot Methods for Phrase-based Statistical Machine Translation*. NAACL-HLT-07

Vladimir N. Vapnik. 1995. *The Nature of Statistical Learning Theory*. Springer

Stephen Wan and Cornelia Maria Verspoor. 1998. *Automatic English-Chinese name transliteration for development of multilingual resources*. COLING-ACL-98

Hua Wu and Haifeng Wang. 2007. *Pivot Language Approach for Phrase-based Statistical Machine Translation*. ACL-07

Hua Wu and Haifeng Wang. 2009. *Revisiting Pivot Language Approach for Machine Translation*. ACL-09

Dmitry Zelenko and Chinatsu Aone. 2006. *Discriminative methods for transliteration*. EMNLP-06

Min Zhang, Haizhou Li and Jian Su. 2004. *Direct Orthographical Mapping for machine transliteration*. COLING-04

Discriminant Ranking for Efficient Treebanking

Yi Zhang **Valia Kordoni**
LT-Lab, German Research Center for Artificial Intelligence (DFKI GmbH)
Department of Computational Linguistics, Saarland University
{yzhang,kordoni}@coli.uni-sb.de

Abstract

Treebank annotation is a labor-intensive and time-consuming task. In this paper, we show that a simple statistical ranking model can significantly improve treebanking efficiency by prompting human annotators, well-trained in disambiguation tasks for treebanking but not necessarily grammar experts, to the most relevant linguistic disambiguation decisions. Experiments were carried out to evaluate the impact of such techniques on annotation efficiency and quality. The detailed analysis of outputs from the ranking model shows strong correlation to the human annotator behavior. When integrated into the treebanking environment, the model brings a significant annotation speed-up with improved inter-annotator agreement.[†]

1 Introduction

The development of a large-scale treebank (Marcus et al., 1993; Hajič et al., 2000; Brants et al., 2002) with rich syntactic annotations is a highly rewarding task. But the huge amount of manual labor required for the annotation task itself, as well as the difficulties in standardizing linguistic analyses, results in long development cycles of such valuable language resources, which typically amounts to years or even decades. Despite the profound scientific and practical value of detailed syntactic treebanks, the requirement and necessity for long-term commitment raises the risk cost of such projects, a fact which often makes them not feasible in many current economical environments.

In recent years, computational grammars have been employed to assist the construction of such language resources. A typical development model involves a parser which generates candidate analyses, and human annotators who manually identify the desired tree structure. This treebanking method dramatically reduces the cost of training annotators, for they are not required to spontaneously produce linguistic solutions to various phenomena. Instead, they are trained to associate their language intuition with specific linguistically-relevant decisions. How to select and carefully present such decisions to the annotators is thus crucial for achieving high annotation speed and quality. On the other hand, for large treebanking projects, parallel annotation with multiple annotators is usually necessary. Inter-annotator agreement is a crucial quality measure in such cases. But improvements on annotation speed should not be achieved at expense of the quality of the treebank.

With both speed and quality in mind, a good treebank annotation method should acknowledge the complexity of the decision-making process; for instance, the same tree can be disambiguated by different sets of individual decisions which are mutually dependent. The annotation method should also strive to create a distraction-free environment for annotators who can then focus on making the judgments. To this effect, we present a simple statistical model that learns from the annotation history, and offers a ranking of disambiguation decisions from the most to the least relevant

[†]The first author thanks the German Excellence Cluster of Multimodal Computing and Interaction for the support of the work.

Coling 2008: Poster Volume, pages 1453–1461,
Beijing, August 2010

ones, which enables well-trained human annotators to speed-up treebanking without compromising on the quality of the linguistic decisions guiding the annotation task.

The remaining of this paper is structured as follows: Section 2 gives an overview of the difficulties in syntactic annotation, and the potential ways of improving the annotation efficiency without damaging the quality; Section 3 presents the new annotation method which is based on a statistical discriminant ranking model; Sections 4 and 5 describe the setup and results of a series of annotation experiments; Section 6 concludes the paper and proposes future research directions.

2 Background

Large-scale full syntactic annotation has for quite some time been approached with mixed feelings by researchers. On the one hand, detailed syntactic annotation serves as a basis for corpus-linguistic study and data-driven NLP methods. Especially, when combined with popular supervised machine learning methods, richly annotated language resources, like, for instance, treebanks, play a key role in modern computational linguistics. The public availability of large-scale treebanks in recent years has stimulated the blossoming of data-driven approaches to syntactic and semantic parsing.

On the other hand, though, the creation of detailed syntactic structures turns out to be an extremely challenging task. From the choice of the appropriate linguistic framework and the design of the annotation scheme to the choice of the text source and the working protocols on the syncronization of the parallel development, as well as the quality assurance, none of these steps in the entire annotation procedure is considered a solved issue. Given the vast design choices, very few of the treebanking projects have made it through all these difficult annotation stages. Even the most outstanding projects have not been completed without receiving criticisms.

Our treebanking project is no exception. The aim of the project is to provide annotations of the Wall Street Journal (henceforward WSJ) sections of the Penn Treebank (henceforward PTB (Marcus et al., 1993)) with the help of the English Resource Grammar (henceforward ERG; (Flickinger, 2002)), a hand-written large-scale and wide-coverage grammar of English in the framework of Head-Driven Phrase Structure Grammar (HPSG; (Pollard and Sag, 1994)). Such annotations are very rich linguistically, since apart from syntax they also incorporate semantic information. The annotation cycle is organized into iterations of parsing, treebanking in the sense of disambiguating syntactic and semantic analyses of the various linguistic phenomena contained in the corpus, error analysis and grammar/treebank update cycles. That is, sentences from the WSJ are first parsed with the PET parser (Callmeier, 2001), an efficient unification-based parser, using the ERG. The parsing results are then manually disambiguated by human annotators. However, instead of considering individual trees, the annotation process is mostly invested on binary decisions which are made on either accepting or rejecting constructions or lexical types. Each of such decisions, called discriminants, as we will also see in the following, reduces the number of the trees satisfying the constraints. The process is presented in the next section in more detail. What should, though, be clear is that the aforementioned multi-cycle annotation procedure is as time-consuming and human-error prone as any other, despite the fact that at the center of the entire annotation cycle lies a valuable linguistic resource, which has been developed with a lot of effort over many years, namely the ERG. For the first period of this project, we have established an average speed of 40 sentences per annotator hour, meaning a total of $\sim$1200 hours of annotation for the entire WSJ. Including the long training period at the beginning of the project, and periodical grammar and treebank updates, the project period is roughly two years with two part-time annotators employed.

3 Statistical Discriminant Ranking

3.1 Discriminants & Decisions

One common characteristic of modern treebanking efforts – especially, in so-called dynamic treebanking platforms (cf., for instance, (Oepen et al., 2002) and http://redwoods.stanford.edu), like the one we are describing and referring

to extensively in the following, is that the candidate trees are constructed automatically by the grammar, and then manually disambiguated by human annotators. In doing so, linguistically rich annotation is built efficiently with minimum manual labor. In order to further improve the manual disambiguation efficiency, systems like `[incr tsdb()]` (Oepen, 2001) compute the difference between candidate analyses. Instead of looking at the huge parse forest, the treebank annotators select or reject the features that distinguish between different parses, until no ambiguity remains (either one analysis is accepted from the parse forest, or all of them are rejected). The number of decisions for each sentence is normally around $log_2 n$ where n is the total number of candidate trees. For a sentence with 5000 candidate readings, only about 12 treebanking decisions are required for a complete disambiguation. A similar method was also proposed in (Carter, 1997).

Formally, an attribute that distinguishes between different parses is called a *discriminant*. For Redwoods-style treebanks, this is extracted either from the syntactic derivation trees or the semantic representations (in the form of Minimal Recursion Semantics (`MRS`; (Copestake et al., 2005))).

Figure 1 shows an example graphical annotation interface. At the top of the window, a list of action buttons shows the operations permitted on the sentence level. Then the sentence in its original `PTB` bracket format is shown. $15 : 0$ indicates that at the current disambiguation state, 15 trees remain to be disambiguated while 0 has been eliminated. On the left large panel, the candidate trees are shown in their simplified phrase-structure representation. Note that the actual `HPSG` analyses are not shown in the screenshot and can be displayed on request. On the right large panel, the list of effective discriminants (see Section 3.2) up to this disambiguation state is shown. The highlighted discriminant in Figure 1 suggests a possibility of constructing the entire sentence by choosing a subject-head rule (SUBJH), taking *"ms. Haag"* as the subject and *"plays Elianti."* as the head daughter. When the discriminant is clicked, the annotator can say *yes* or *no* to it, hence narrowing the remaining trees to the *In Parses* or *Out Parses*. The *unknown* button is used to mark the uncertainties and is rarely used.

Note that in this interface, the discriminants are sorted in descending order according to their length, meaning that the discriminants related to higher level constructions are shown before the lexical type choices. When up to 500 parses are stored in the forest, the average number of discriminants per forest is about 100. Scanning through the long list manually can be time-consuming and distracting.

Kordoni and Zhang (2009) show that annotators tend to start with the decisions with the most certainty, and delay the "hard" decisions as much as possible. As the decision progresses, many of the "hard" discriminants will receive an inferred value from the certain decisions. Our annotation guideline only describes specific decisions. The order in which discriminants are chosen is left underspecified and very much depends on personal styles. In practice, we see that our annotators gradually developed complex strategies which involve both top-down and bottom-up pruning.

One potential drawback of such a discriminant-based treebanking method is that the process is very sensitive to decision errors. One wrong judgment can rule out the correct tree and ruin the analysis of the sentence. In such a case, the annotators usually resort to backtracking to previous decisions they had made. To compensate for this, we ask our annotators to double-check the treebanked analysis before saving the disambiguation result. And in case of doubt, they are instructed to avoid ambivalent decisions as much as possible.

3.2 Maximum-Entropy-Based Discriminant Ranking Model

Suppose for a sentence ω, a parse forest Y was generated by the grammar. Note that for efficiency reasons, the parse forest might have been trimmed to only contain up to n top readings ranked by the parse disambiguation model. For convenience, we note the parse forest Y as a set of parses $\{y_1, y_2, \ldots, y_n\}$. Each discriminant d defines a binary valued function δ on the parse forest ($\delta : Y \mapsto \{0, 1\}$) , which can be interpreted as whether a parse y_i has attribute d or not. By the nature of this definition, each discriminant

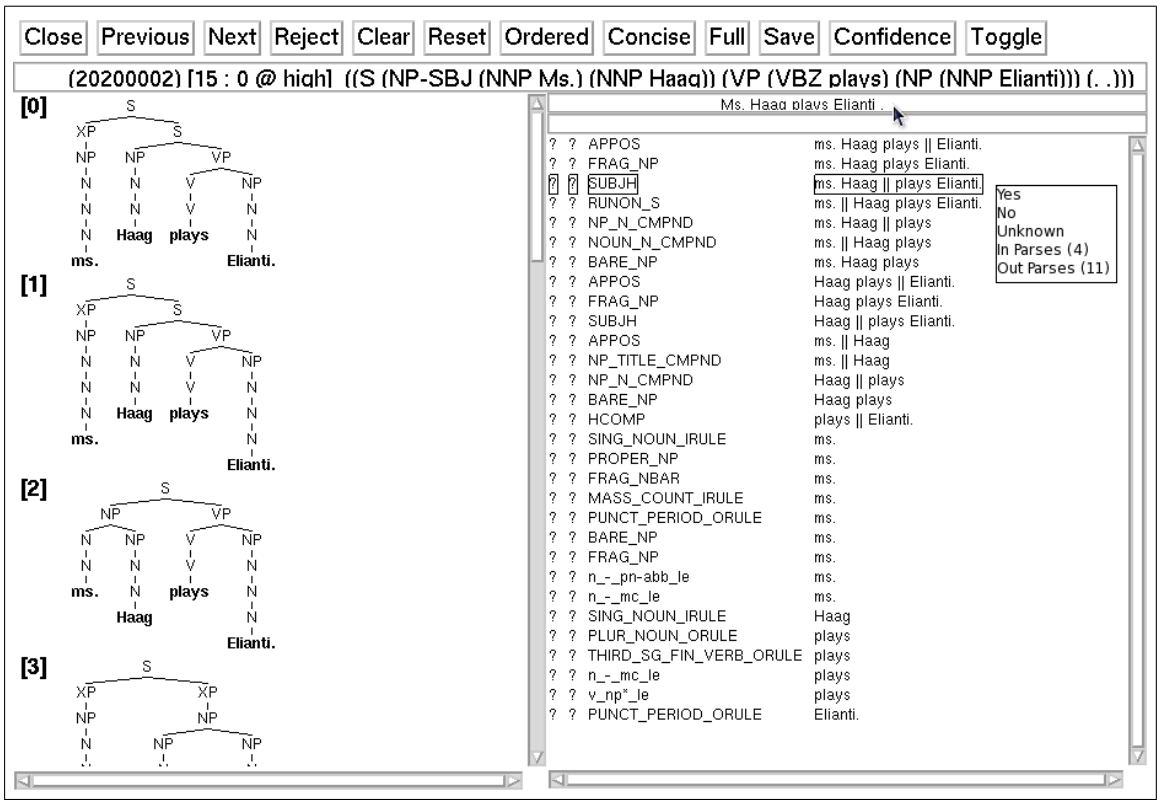


Figure 1: Screenshot of the discriminant-based treebanking graphical annotator interface

function defines a bi-partition of the parse forest. When both subsets of the partition are non-empty, i.e., there exists at least one y_p and y_q such that $\delta(y_p) = 0$ and $\delta(y_q) = 1$, the discriminant is considered *effective* on the forest Y. In the following discussion, we are only considering the set of effective discriminants D for parse forest Y.

Instead of directly predicting the outcome of disambiguation decision on each discriminant (i.e., whether the GOLD tree has discriminant function value 0 or 1), our model tries to measure the probability of a discriminant being chosen by human annotators, regardless of the *yes/no* decision. For each discriminant d, and the parse forest Y, a set of feature functions $f_1, f_2, \ldots, f_k$ receive real values, and contribute to the following log-linear model:

$$P(d|Y, D) = \frac{exp(\sum_{i=1}^{k} \lambda_i f_i(d, Y))}{\sum_{d' \in D} exp(\sum_{i=1}^{k} \lambda_i f_i(d', Y))} \quad (1)$$

where $\lambda_1, \lambda_2, \ldots, \lambda_k$ are the parameters of the model.

To estimate these model parameters, we gather the annotation logs from our treebank annotators on the completed datasets with detailed information about each discriminant. Apart from the necessary information to reconstruct the discriminants from the forest, the log also contains the status information of i) whether the discriminant takes value 0 or 1 on the gold tree; ii) whether the human annotator has said *yes* or *no* to the discriminant. Note that the human annotator does not need to manually decide on the value of each discriminant. Whenever a new decision is made, the forest will be pruned to the subset of trees compatible with the decision. And all remaining discriminants are checked for effectiveness on the pruned forest. Discriminants which become ineffective from previous decisions are said to have received *inferred* values.

The parameters of the model are estimated by the open-source maximum entropy parameter es-

timation toolkit TADM[1]. For training, we use all the manually disambiguated discriminants as positive instances, and automatically inferred discriminants as negative instances.

The discriminant ranking model is applied during the manual annotation sessions. When a parse forest is loaded and the discriminants are constructed, each discriminant is assigned an (unnormalized) score $\sum_{i=1}^{k} \lambda_i f_i(d, Y)$, and the list of discriminants is sorted by descending order of the score accordingly. The scoring and sorting adds negligible additional computation on the treebanking software, and is not noticeable to the human annotators. By putting those discriminants that are potentially to be manually judged near the top of the list, the model saves manual labor on scanning through the lengthy list by filtering out ambivalent discriminants.

Note that this discriminant ranking model predicts the possibility of a discriminant being manually disambiguated. It is not modeling the specific decision that the human annotator makes on the discriminant. Including the decision outcome in the model can potentially damage the annotation quality if annotators develop a habit of overtrusting the model prediction, making the whole manual annotation pointless. A discriminant ranking model, however, only suggestively re-orders the discriminants on the presentation level, which are much safer when the annotation quality is concerned.

3.3 Feature Model for Syntactic Discriminants

In practice, there are different ways of finding discriminants from the parse forest. For instance, the [incr tsdb()] system supports both syntax-based and semantics-based discriminants. The syntax-based discriminants are extracted from the derivation trees of the HPSG analyses. All HPSG rule applications (unary or binary) and choices of lexical entries are picked as candidate discriminants and checked for effectiveness. The semantics-based discriminants, on the other hand, represent the differences on the semantic structures (MRS in the cases of DELPH-IN[2] grammars). With a few exceptions, many DELPH-IN HPSG treebanks choose to use the syntactic discriminants which allow human annotators to pick the low-level constructions. The above proposed ranking model works for different types of discriminants (and potentially a mixture of different discriminant types). But for the evaluation of this paper, we show the feature model designed for the syntactic discriminants only.

The syntactic discriminants record the differences between derivation trees by memorizing direct rule applications and lexical choices. Beside the rule or lexical entry name, the discriminant also records the information concerning the corresponding constituent, e.g., the category and spanning of the constituent, the parent and daughters of the constituent, etc. Furthermore, given the discriminant d and the parse forest Y, we can calculate the distribution of parses over the value of the discriminant function δ, which can be characterized by $\sum_{y \in Y} \delta(y)/|Y|$. This numeric feature indicates how many parses can be ruled out with the given discriminant.

As example, for the highlighted discriminant in Figure 1, the extracted features are listed in Table 1.

4 Experiment Setup

To test the effectiveness of the discriminant ranking models, we carried out a series of experiments, investigating their effects on both annotation speed and quality. The experiment was done in the context of our ongoing annotation project of the WSJ sections of the PTB described in Section 2. Despite sharing the source of texts, the new project aims to create an independently annotated corpus. Therefore, the trees from the PTB were not used to guide the disambiguation process. In this annotation project, two annotators (both graduate students, referred to as A and B below) are employed to manually disambiguate the parsing outputs of the ERG. For quality control and adjudication in case of disagreement, a third linguist/grammarian annotates parts of the treebank in parallel.

With the help of our annotation log files, which record in details the manual decision-making process, we trained three discriminant ranking mod-

[1] http://tadm.sourceforge.net/

[2] http://www.delph-in.net/

Feature	Possible Values	Example
discriminant type	RULE/LEX	RULE
edge position	FULL/FRONT/BACK	FULL
edge span	$length$(constituent)/$length$(sentence)	4/4
edge category	rule or lexical type name	SUBJH
level of discrimination	$\sum_{y \in Y} \delta(y)/\lvert Y \rvert$	4/15
branch splitting	$length$(left-dtr)/$length$(constituent)	2/4

Table 1: Features for syntactic discriminant ranking model and example values for the highlighted discriminant in Figure 1

els with the datasets completed so far: MODEL-A and MODEL-B trained with annotation logs from two annotators separately, and MODEL-BOTH trained jointly with data from both annotators. For each annotator's model (MODEL-A and MODEL-B), we used about 6,000 disambiguated parse forests for training. For each of these 6,000 forests, the log file contains about 600,000 effective discriminants, among which only $\sim$6% received a manual decision.

To evaluate the treebanking speed, we have the annotators work under a distraction-free environment and record their annotation speed. The speed is averaged over several 1-hour annotation sessions. Different discriminant ranking models were used without the annotators being informed of the details of the setting.

As testing dataset, we use the texts from the PARC 700 Dependency Bank (King et al., 2003), which include 700 carefully selected sentences from the WSJ sections of the PTB. These sentences were originally chosen for the purpose of parser evaluation. Many linguistically challenging phenomena are included in these sentences, although the sentence length is shorter in average than the sentence length in the entire WSJ. The language is also less related to the financial domain specific language observed in the WSJ. We parsed the dataset with the Feb. 2009 version of the ERG, and recorded up to 500 trees per sentence (ranked by a MaxEnt parse selection model trained on previously treebanked WSJ sections).

5 Results

Although we employed a typical statistical ranking model in our system, it is difficult to directly evaluate the absolute performance of the predicted ranking. Annotators only annotate a very small subset of the discriminants, and their order is not fully specified. To compare the behavior of models trained with data annotated by different annotators, we plot the relative ranking (normalized to $[0, 1]$ for each sentence, with 0 being the highest rank and 1 the lowest) of discriminants for 50 sentences in Figure 2.

The plot shows a strong positive linear correlation between the two ranking models. The particularly strong correlation at the low and high ends of the ranking shows that the two annotators share a similar behavior pattern concerning the most and least preferred discriminants. The correlation is slightly weaker in the middle ranking zone, where different preferences or annotation styles can be observed.

To further visualize the effect of the ranking model, we highlighted with color the discriminants which are manually annotated by annotator B under a basic setting without using the ranking models. 75% of these "prominent" discriminants are grouped within the top-25% region of the plot. Without surprise, the model B gives an average relative ranking of 0.18 as oppose to 0.21 with model A. The overall distribution of rankings for manually disambiguated discriminants are shown in Figure 3.

In Table 2, the average treebanking speed of two annotators over multiple annotation sessions is shown. The baseline model ranks the discriminants by the spanning length of the corresponding constituent, and uses the alphabetical order of the rule or lexical type names as tie-breaker. The own-model refers to the annotation sessions which have been carried out by the annotators us-

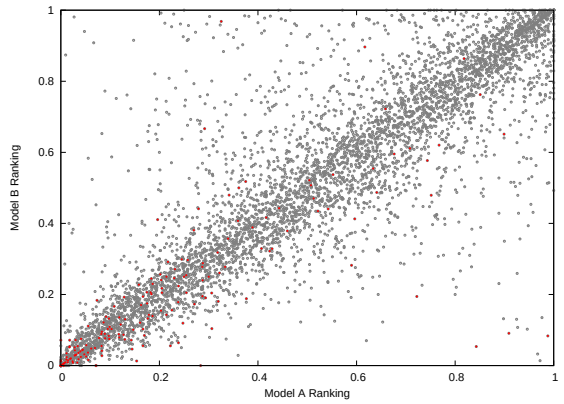


Figure 2: Correlation of discriminant ranks with different models and manual annotation

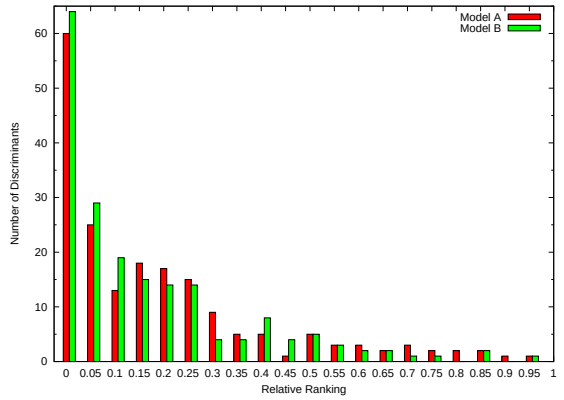


Figure 3: Histogram of rankings given by two models on discriminants manually picked by annotator B

ing their own ranking model. The peer-model refers to the annotation sessions where the annotators use their peer colleague's model. And finally, the joint-model refers to the annotations done by the jointly trained model.

The annotation efficiency was boosted by over 50% with all the discriminant ranking models. The own-model setting achieved best speed. This is probably due to the fact that the model most closely reflects the annotation habit of the annotator. But the advantage over other models is very small.

To measure the inter-annotator agreement, we calculate the Cohen's KAPPA (Carletta, 1996) on the constituents of the derivation trees:

$$\kappa = \frac{Pr(a) - Pr(e)}{1 - Pr(e)} \quad (2)$$

Ranking Model	Speed (s/h)	Speed-up (%)
Baseline	61.9	–
Own-model	**96.1**	**55%**
Peer-model	94.6	53%
Joint-model	95.0	53%

Table 2: Average annotation speed with different discriminant ranking models

where $Pr(a)$ is the relative observed agreement between annotators, and $Pr(e)$ is the probability of two annotators agreeing by chance. The calculation of $Pr(a)$ can be done in a similar way to the calculation of PARSEVAL labeled bracketing accuracy, while $Pr(e)$ is estimated by averaging the agreement over a large set of tree pairs randomly sampled from the parse forest. Since the calculation of κ takes into account the agreement occurring by chance, it is a safer (though has the tendency of being overly conservative) measure of agreement.

Ranking Model	Cohen's KAPPA (κ)
Baseline	0.5404
Own-model	**0.5798**
Peer-model	0.5567
Joint-model	0.5536

Table 3: Inter-annotator agreement measured by constituent-level Cohen's KAPPA

The numbers in Table 3 show that the use of discriminant ranking models results in a small improvement to the inter-annotator agreement, with the best agreement achieved by each annotator using the model trained with their own annotation records. These numbers are comforting in that the annotation quality is not damaged by our new way to present the linguistic decisions.

Note that the relatively low inter-annotator agreement in this experiment is due to the fact that we used a dataset which involves non-trivial linguistic phenomena that are on average more difficult than the texts in the WSJ corpus. Another fact is that these annotations were done under time pressure. The annotators are not encouraged to go backwards to check and correct the previous sentences during these sessions. On the entire WSJ, we have recorded a stable and persistently higher

agreement level at $\kappa = 0.6$. Given the highly detailed linguistic annotations specified by the grammar (over 260 rules and 800 lexical types), this figure indicates a very substantial agreement between our annotators. Our further investigation has shown that the agreement figure hits the ceiling at around $\kappa = 0.65$. Further training and discussion is not rewarded with sustainable improvement of annotation quality.

Apart from the numerical evaluation, we also interview our annotators for subjective feelings about the various ranking models. There is generally a very positive attitude towards all the ranking models over the baseline. An easily decidable discriminant is usually found within the top-3 with very few exceptions, which leads to a self-noticeable speed-up that confirms our numeric findings. It is also interesting to note that, despite the substantial difference between the statistical models, the difference is hardly noticed by the annotators. And the results only show small variations in both the annotation speed, as well as the inter-annotator agreement.

The annotators also claim that the speed-up is somewhat diminished over the "rejected" sentences, for which none of the candidate trees are acceptable. In such cases, the annotators still have to go through a long sequence of discriminants, and sometimes have to redo the previous steps in fear of the chain-effect of wrong decisions. How to compensate for the psychological insatisfaction of rejecting all analyses while maintaining good annotation speed and quality is a new topic for our future research.

6 Conclusion & Future Work

We propose to use a statistical ranking model to assist the discriminant-based treebank annotation. Our experiment shows that such a model, trained on annotation history, brings a huge efficiency improvement together with slightly improved inter-annotator agreement.

Although the reported experiments were carried out on the specific `HPSG` treebank, we believe that the proposed ranked discriminant-based annotation method can be applied in annotation tasks concerning different linguistic frameworks, or even different layers of linguistic representation. Apart from the specific features presented in Section 3.3, the model itself does not assume a phrase-structure tree annotation, and the discriminants can take various forms. Assuming a "grammar" produces a number of candidate analyses, the annotators can rely on the ranking model to efficiently pick relevant discriminants, and focus on making linguistically relevant decisions. This is especially suitable for large annotation tasks aiming for parallel rich annotation by multiple annotators, where fully manual annotation is not feasible and high inter-annotator agreement hard to achieve.

The ranking model is based on annotation history and influences the future progress of treebanking. It can be dynamically integrated into the treebank development cycles in which the annotation habit evolves over time. Such a model can also shorten the training period for new annotators, which is an interesting aspect for our future investigation.

From a different point of view, the rankings of the discriminants show annotators' confidence on various ambiguities. The clearly uneven distribution over discriminants can also provide grammar writers with interesting feedback, helping with the improvement of the linguistic analysis. We would also like to integrate confidence measures into the computer-assisted treebank annotation process, which could potentially help annotators make difficult decisions, such as whether to reject all trees for a sentence.

References

[Brants et al.2002] Brants, Sabine, Stefanie Dipper, Silvia Hansen, Wolfgang Lezius, and George Smith. 2002. The tiger treebank. In *Proceedings of the workshop on treebanks and linguistic theories*, pages 24–41.

[Callmeier2001] Callmeier, Ulrich. 2001. Efficient parsing with large-scale unification grammars. Master's thesis, Universität des Saarlandes, Saarbrücken, Germany.

[Carletta1996] Carletta, Jean. 1996. Assessing agreement on classification tasks: the kappa statistic. *Computational Linguistics*, 22(2):249–254.

[Carter1997] Carter, David. 1997. The treebanker: a tool for supervised training of parsed corpora. In

Proceedings of the ACL Workshop on Computational Environments for Grammar Development and Linguistic Engineering, pages 9–15, Madrid, Spain.

[Copestake et al.2005] Copestake, Ann, Dan Flickinger, Carl J. Pollard, and Ivan A. Sag. 2005. Minimal recursion semantics: an introduction. *Research on Language and Computation*, 3(4):281–332.

[Flickinger2002] Flickinger, Dan. 2002. On building a more efficient grammar by exploiting types. In Oepen, Stephan, Dan Flickinger, Jun'ichi Tsujii, and Hans Uszkoreit, editors, *Collaborative Language Engineering*, pages 1–17. CSLI Publications.

[Hajič et al.2000] Hajič, Jan, Alena Böhmová, Eva Hajičová, and Barbora Vidová-Hladká. 2000. The Prague Dependency Treebank: A Three-Level Annotation Scenario. In Abeillé, A., editor, *Treebanks: Building and Using Parsed Corpora*, pages 103–127. Amsterdam:Kluwer.

[King et al.2003] King, Tracy H., Richard Crouch, Stefan Riezler, Mary Dalrymple, and Ronald M. Kaplan. 2003. The PARC 700 Dependency Bank. In *Proceedings of the 4th International Workshop on Linguistically Interpreted Corpora, held at the 10th Conference of the European Chapter of the Association for Computational Linguistics (EACL'03)*, Budapest, Hungary.

[Kordoni and Zhang2009] Kordoni, Valia and Yi Zhang. 2009. Annotating wall street journal texts using a hand-crafted deep linguistic grammar. In *Proceedings of The Third Linguistic Annotation Workshop (LAW III)*, Singapore.

[Marcus et al.1993] Marcus, Mitchell P., Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of english: The penn treebank. *Computational Linguistics*, 19(2):313–330.

[Oepen et al.2002] Oepen, Stephan, Kristina Toutanova, Stuart Shieber, Christopher Manning, Dan Flickinger, and Thorsten Brants. 2002. The LinGO Redwoods treebank: motivation and preliminary applications. In *Proceedings of COLING 2002: The 17th International Conference on Computational Linguistics: Project Notes*, Taipei, Taiwan.

[Oepen2001] Oepen, Stephan. 2001. [incr tsdb()] — competence and performance laboratory. User manual. Technical report, Computational Linguistics, Saarland University, Saarbrücken, Germany.

[Pollard and Sag1994] Pollard, Carl J. and Ivan A. Sag. 1994. *Head-Driven Phrase Structure Grammar*. University of Chicago Press, Chicago, USA.

Extracting and Ranking Product Features in Opinion Documents

Lei Zhang
Department of Computer Science
University of Illinois at Chicago
lzhang3@cs.uic.edu

Bing Liu
Department of Computer Science
University of Illinois at Chicago
liub@cs.uic.edu

Suk Hwan Lim
Hewlett-Packard Labs
suk-hwan.lim@hp.com

Eamonn O'Brien-Strain
Hewlett-Packard Labs
eob@hpl.hp.com

Abstract

An important task of opinion mining is to extract people's opinions on features of an entity. For example, the sentence, "*I love the GPS function of Motorola Droid"* expresses a positive opinion on the "*GPS function*" of the Motorola phone. "*GPS function*" is the feature. This paper focuses on mining features. *Double propagation* is a state-of-the-art technique for solving the problem. It works well for medium-size corpora. However, for large and small corpora, it can result in low precision and low recall. To deal with these two problems, two improvements based on *part-whole* and "*no*" patterns are introduced to increase the recall. Then feature ranking is applied to the extracted feature candidates to improve the precision of the top-ranked candidates. We rank feature candidates by feature importance which is determined by two factors: feature relevance and feature frequency. The problem is formulated as a bipartite graph and the well-known web page ranking algorithm HITS is used to find important features and rank them high. Experiments on diverse real-life datasets show promising results.

1 Introduction

In recent years, opinion mining or sentiment analysis (Liu, 2010; Pang and Lee, 2008) has been an active research area in NLP. One task is to extract people's opinions expressed on features of entities (Hu and Liu, 2004). For example, the sentence, "*The picture of this camera is amazing*", expresses a positive opinion on the *picture* of the camera. "*picture*" is the feature. How to extract features from a corpus is an important problem. There are several studies on feature extraction (e.g., Hu and Liu, 2004, Popescu and Etzioni, 2005, Kobayashi et al., 2007, Scaffidi et al., 2007, Stoyanov and Cardie. 2008, Wong et al., 2008, Qiu et al., 2009). However, this problem is far from being solved.

Double Propagation (Qiu et al., 2009) is a state-of-the-art unsupervised technique for solving the problem. It mainly extracts noun features, and works well for medium-size corpora. But for large corpora, this method can introduce a great deal of noise (low precision), and for small corpora, it can miss important features. To deal with these two problems, we propose a new feature mining method, which enhances that in (Qiu et al., 2009). Firstly, two improvements based on *part-whole* patterns and "*no*" patterns are introduced to increase recall. Part-whole or *meronymy* is an important semantic relation in NLP, which indicates that one or more objects are parts of another object.

Coling 2008: Poster Volume, pages 1462–1470,
Beijing, August 2010

For example, the phrase "*the engine of the car*" contains the part-whole relation that "*engine*" is part of "*car*". This relation is very useful for feature extraction, because if we know one object is part of a product class, this object should be a feature. "*no*" pattern is another extraction pattern. Its basic form is the word "no" followed by a noun/noun phrase, for instance, "no noise". People often express their short comments or opinions on features using this pattern. Both types of patterns can help find features missed by double propagation. As for the low precision problem, we present a feature ranking approach to tackle it. We rank feature candidates based on their importance which consists of two factors: feature relevance and feature frequency. The basic idea of feature importance ranking is that if a feature candidate is correct and frequently mentioned in a corpus, it should be ranked high; otherwise it should be ranked low in the final result. Feature frequency is the occurrence frequency of a feature in a corpus, which is easy to obtain. However, assessing feature relevance is challenging. We model the problem as a bipartite graph and use the well-known web page ranking algorithm HITS (Kleinberg, 1999) to find important features and rank them high. Our experimental results show superior performances. In practical applications, we believe that ranking is also important for feature mining because ranking can help users to discover important features from the extracted hundreds of fine-grained candidate features efficiently.

2 Related work

Hu and Liu (2004) proposed a technique based on association rule mining to extract product features. The main idea is that people often use the same words when they comment on the same product features. Then frequent itemsets of nouns in reviews are likely to be product features while the infrequent ones are less likely to be product features. This work also introduced the idea of using opinion words to find additional (often infrequent) features.

Popescu and Etzioni (2005) investigated the same problem. Their algorithm requires that the product class is known. The algorithm determines whether a noun/noun phrase is a feature by computing the pointwise mutual information (PMI) score between the phrase and class-specific discriminators, e.g., "*of xx*", "*xx has*", "*xx comes with*", etc., where *xx* is a product class. This work first used part-whole patterns for feature mining, but it finds part-whole based features by searching the Web. Querying the Web is time-consuming. In our method, we use predefined part-whole relation patterns to extract features in a domain corpus. These patterns are domain-independent and fairly accurate.

Following the initial work in (Hu and Liu 2004), several researchers have further explored the idea of using opinion words in product feature mining. A dependency based method was proposed in (Zhuang et al., 2006) for a movie review analysis application. Qiu et al. (2009) proposed a double propagation method, which exploits certain syntactic relations of opinion words and features, and propagates through both opinion words and features iteratively. The extraction rules are designed based on different relations between opinion words and features, and among opinion words and features themselves. Dependency grammar was adopted to describe these relations. In (Wang and Wang, 2008), another bootstrapping method was proposed. In (Kobayashi et al. 2007), a pattern mining method was used. The patterns are relations between feature and opinion pairs (they call *aspect-evaluation* pairs). The patterns are mined from a large corpus using pattern mining. Statistics from the corpus are used to determine the confidence scores of the extraction.

In general information extraction, there are two approaches: rule-based and statistical. Early extraction systems are mainly based on rules (e.g., Riloff, 1993). In statistical methods, the most popular models are Hidden Markov Models (HMM) (Rabiner, 1989), Maximum Entropy Models (ME) (Chieu et al., 2002) and Conditional Random Fields (CRF) (Lafferty et al., 2001). CRF has been shown to be the most effective method. It was used in (Stoyanov et al., 2008). However, a limitation of CRF is that it only captures local patterns rather than long range patterns. It has been shown in (Qiu et al., 2009) that many feature and opinion word pairs have long range dependencies. Experimental results in (Qiu et al., 2009) indicate that CRF does not perform well.

Other related works on feature extraction mainly use topic modeling to capture topics in

reviews (Mei et al., 2007). In (Su et al., 2008), the authors also proposed a clustering based method with mutual reinforcement to identify features. However, topic modeling or clustering is only able to find some general/rough features, and has difficulty in finding fine-grained or precise features, which is more related to information extraction.

3 The Proposed Method

As discussed in the introduction section, our proposed method deals with the problems of double propagation. So let us give a short explanation why double propagation can cause problems in large or small corpora.

Double propagation assumes that features are nouns/noun phrases and opinion words are adjectives. It is shown that opinion words are usually associated with features in some ways. Thus, opinion words can be recognized by identified features, and features can be identified by known opinion words. The extracted opinion words and features are utilized to identify new opinion words and new features, which are used again to extract more opinion words and features. This propagation or bootstrapping process ends when no more opinion words or features can be found. The biggest advantage of the method is that it requires no additional resources except an initial seed opinion lexicon, which is readily available (Wilson et al., 2005, Ding et al., 2008). Thus it is domain independent and unsupervised, avoiding laborious and time-consuming work of labeling data for supervised learning methods. It works well for medium–size corpora. But for large corpora, this method may extract many nouns/noun phrases which are not features. The precision of the method thus drops. The reason is that during propagation, adjectives which are not opinionated will be extracted as opinion words, e.g., "*entire*" and "*current*". These adjectives are not opinion words but they can modify many kinds of nouns/noun phrases, thus leading to extracting wrong features. Iteratively, more and more noises may be introduced during the process. The other problem is that for certain domains, some important features do not have opinion words modifying them. For example, in reviews of mattresses, a reviewer may say "*There is a valley on my mattress*", which implies a negative opinion because "*valley*" is undesirable for a mattress. Obviously, "*valley*" is a feature, but "*valley*" may not be described by any opinion adjective, especially for a small corpus. Double propagation is not applicable in this situation.

To deal with the problem, we propose a novel method to mine features, which consists of two steps: feature extraction and feature ranking. For feature extraction, we still adopt the double propagation idea to populate feature candidates. But two improvements based on part-whole relation patterns and a "no" pattern are made to find features which double propagation cannot find. They can solve part of the recall problem. For feature ranking, we rank feature candidates by feature importance.

A part-whole pattern indicates one object is part of another object. For the previous example "*There is a valley on my mattress*", we can find that it contains a part-whole relation between "*valley*" and "*mattress*". "*valley*" belongs to "*mattress*", which is indicated by the preposition "*on*". Note that "*valley*" is not actually a part of mattress, but an effect on the mattress. It is called a *pseudo part-whole* relation. For simplicity, we will not distinguish it from an actual part-whole relation because for our feature mining task, they have little difference. In this case, "$noun_1$ on $noun_2$" is a good indicative pattern which implies $noun_1$ is part of $noun_2$. So if we know "*mattress*" is a class concept, we can infer that "*valley*" is a feature for "*mattress*". There are many phrase or sentence patterns representing this type of semantic relation which was studied in (Girju et al, 2006). Beside part-whole patterns, "no" pattern is another important and specific feature indicator in opinion documents. We introduce these patterns in detail in Sections 3.2 and 3.3.

Now let us deal with the first problem: noise. With opinion words, part-whole and "no" patterns, we have three feature indicators at hands, but all of them are ambiguous, which means that they are not hard rules. We will inevitably extract wrong features (also called noises) by using them. Pruning noises from feature candidates is a hard task. Instead, we propose a new angle for solving this problem: feature ranking. The basic idea is that we rank the extracted feature candidates by feature importance. If a feature candidate is correct and important, it should be ranked high. For unimportant feature or

noise, it should be ranked low in the final result. Ranking is also very useful in practice. In a large corpus, we may extract hundreds of fine-grained features. But the user often only cares about those important ones, which should be ranked high. We identified two major factors affecting the feature importance: one is feature relevance and the other is feature frequency.

Feature relevance: it describes how possible a feature candidate is a correct feature. We find that there are three strong clues to indicate feature relevance in a corpus. The first clue is that a correct feature is often modified by multiple opinion words (adjectives or adverbs). For example, in the mattress domain, "*delivery*" is modified by "*quick*" "*cumbersome*" and "*timely*". It shows that reviewers put emphasis on the word "*delivery*". Thus we can infer that "*delivery*" is a possible feature. The second clue is that a feature could be extracted by multiple part-whole patterns. For example, in the car domain, if we find following two phrases, "*the engine of the car*" and "*the car has a big engine*", we can infer that "*engine*" is a feature for car, because both phrases contain part-whole relations to indicate "*engine*" is a part of "*car*". The third clue is the combination of opinion word modification, part-whole pattern extraction and "no" pattern extraction. That is, if a feature candidate is not only modified by opinion words but also extracted by part-whole or "no" patterns, we can infer that it is a feature with high confidence. For example, for sentence "*there is a bad hole in the mattress*", it strongly indicates that "*hole*" is a feature for a mattress because it is modified by opinion word "*bad*" and also in the part-whole pattern. What is more, we find that there is a mutual enforcement relation between opinion words, part-whole and "no" patterns, and features. If an adjective modifies many correct features, it is highly possible to be a good opinion word. Similarly, if a feature candidate can be extracted by many opinion words, part-whole patterns, or "no" pattern, it is also highly likely to be a correct feature. This indicates that the Web page ranking algorithm HITS is applicable.

Feature frequency: This is another important factor affecting feature ranking. Feature frequency has been considered in (Hu and Liu, 2004; Blair-Goldensohn et al., 2008). We consider a feature f_1 to be more important than feature f_2 if f_1 appears more frequently than f_2 in opinion documents. In practice, it is desirable to rank those frequent features higher than infrequent features. The reason is that missing a frequently mentioned feature in opinion mining is bad, but missing a rare feature is not a big issue.

Combining the above factors, we propose a new feature mining method. Experiments show good results on diverse real-life datasets.

3.1 Double Propagation

As we described above, double propagation is based on the observation that there are natural relations between opinion words and features due to the fact that opinion words are often used to modify features. Furthermore, it is observed that opinion words and features themselves have relations in opinionated expressions too (Qiu et al., 2009). These relations can be identified via a dependency parser (Lin, 1998) based on the dependency grammar. The identification of the relations is the key to feature extraction.

Dependency grammar: It describes the dependency relations between words in a sentence. After parsed by a dependency parser, words in a sentence are linked to each other by a certain relation. For a sentence, "*The camera has a good lens*", "*good*" is the opinion word and "*lens*" is the feature of camera. After parsing, we can find that "*good*" depends on "*lens*" with relation ***mod***. Here ***mod*** means that "*good*" is the adjunct modifier for "*lens*". In some cases, an opinion word and a feature are not directly dependent, but they directly depend on a same word. For example, from the sentence "*The lens is nice*", we can find that both feature "*lens*" and opinion word "*nice*" depend on the verb "*is*" with the relation ***s*** and ***pred*** respectively. Here ***s*** means that "*lens*" is the surface subject of "*is*" while ***pred*** means that "*nice*" is the predicate of the "*is*" clause.

In (Qiu et al., 2009), it defines two categories of dependency relations to summarize all types of dependency relations between two words, which are illustrated in Figure 1. Arrows are used to represent dependencies.

Direct relations: It represents that one word depends on the other word directly or they both depend on a third word directly, shown in (a) and (b) of Figure 1. In (a), *B* depends on *A* directly, and in (b) they both directly depend on *D*.

Indirect relation: It represents that one word

depends on the other word through other words or they both depend on a third word indirectly. For example, in (c) of Figure 1, *B* depends on *A* through *D*; in (d) of Figure 1, *A* depends on *D* through I_1 while *B* depends on *D* through I_2. For some complicated situations, there can be more than one I_1 or I_2.

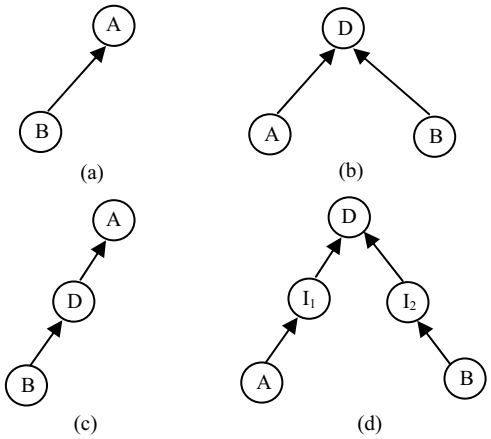


Fig.1 Different relations between A and B

Parsing indirect relations is error-prone for Web corpora. Thus we only use direct relation to extract opinion words and feature candidates in our application. For detailed extraction rules, please refer to the paper (Qiu et al., 2009).

3.2 Part-whole relation

As we discussed above, a part-whole relation is a good indicator for features if the class concept word (the "whole" part) is known. For example, the compound nominal "*car hood*" contains the part-whole relation. If we know "*car*" is the class concept word, then we can infer that "*hood*" is a feature for *car*. Part-whole patterns occur frequently in text and are expressed by a variety of lexico-syntactic structures (Girju et al, 2006; Popescu and Etzioni, 2005). There are two types of lexico-syntactic structures conveying part-whole relations: unambiguous structure and ambiguous structure. The unambiguous structure clearly indicates a part-whole relation. For example, for sentences "*the camera consists of lens, body and power cord.*" and "*the bed was made of wood*". In these cases, the detection of the patterns leads to the discovery of real part-whole relations. We can easily find features of the camera and the bed. Unfortunately, this kind of patterns is not very frequent in a corpus. However, there are many ambiguous expressions that are explicit but convey part-whole relations only in some contexts. For example, for two phrases "*valley on the mattress*" and "*toy on the mattress*", "*valley*" is a part of "*mattress*" whereas "*toy*" is not a part of "*mattress*". Our idea is to use both the unambiguous and ambiguous patterns. Although ambiguous patterns may bring some noise, we can rank them low in the ranking procedure. The following two kinds of patterns are what we have utilized for feature extraction.

3.2.1 Phrase pattern

In this case, the part-whole relation exists in a phrase.

NP + Prep + CP: noun/noun phrase (NP) contains the *part* word and the class concept phrase (CP) contains the *whole* word. They are connected by the preposition word (Prep). For example, "*battery of the camera*" is an instance of this pattern where NP (battery) is the *part* noun and CP (camera) is the *whole* noun. For our application, we only use three specific prepositions: "of", "in" and "on".

CP + with + NP: likewise, CP is the class concept phrase, and NP is the noun/noun phrase. They are connected by the word "*with*". Here NP is likely to be a feature. For example, in a phrase, "*mattress with a cover",* "*cover*" is a feature for mattress.

NP CP or CP NP: noun/noun phase (NP) and class concept phrase (CP) forms a compound word. For example, "*mattress pad*". Here "*pad*" is a feature of "*mattress*".

3.2.2 Sentence pattern

In these patterns, the part-whole relation is indicated in a sentence. The patterns contain specific verbs. The *part* word and the *whole* word can be found inside noun phrases or prepositional phrases which contain specific prepositions. We utilize the following patterns in our application.

"**CP Verb NP**": CP is the class concept phrase that contains the *whole* word, NP is the noun phrase that contains the *part* word and the verb is restricted and specific. For example, in a sentence, "*the phone has a big screen*", we can infer that "*screen*" is a feature for "*phone*", which is a class concept. In sentence patterns, verbs play an important role. We use indicative verbs to find part-whole relations in a sentence,

i.e., “has”, “have” “include” “contain” “consist”, “comprise” and so on (Girju et al, 2006).

It is worth mentioning that in order to use part-whole relations, the class concept word for a corpus is needed, which is fairly easy to find because the noun with the most frequent occurrences in a corpus is always the class concept word based on our experiments.

3.3 “no” Pattern

Besides opinion word and part-whole relation, “no” pattern is also an important pattern indicating features in a corpus. Here “no” represents word *no*. The basic form of the pattern is “no” word followed by noun/noun phrase. This simple pattern actually is very useful to feature extraction. It is a specific pattern for product reviews and forum posts. People often express their comments or opinions on features by this short pattern. For example, in a mattress domain, people always say that “*no noise*” and “*no indentation*”. Here “*noise*” and “*indentation*” are all features for the mattress. We discover that this pattern is frequently used in corpora and a very good indicator for features with a fairly high precision. But we have to take care of the some fixed “no” expression, like “*no problem*” “*no offense*”. In these cases, “*problem*” and “*offense*” should not be regarded as features. We have a list of such words, which are manually compiled.

3.4 Bipartite Graph and HITS Algorithm

Hyperlink-induced topic search (HITS) is a link analysis algorithm that rates Web pages. As discussed in the introduction section, we can apply the HITS algorithm to compute feature relevance for ranking.

Before illustrating how HITS can be applied to our scenario, let us first give a brief introduction to HITS. Given a broad search query q, HITS sends the query to a search engine system, and then collects k (k = 200 in the original paper) highest ranked pages, which are assumed to be highly relevant to the search query. This set is called the root set R; then it grows R by including any page pointed to a page in R, then forms a base set S. HITS then works on the pages in S. It assigns every page in S an **authority score** and a **hub score**. Let the number of pages to be studied be n. We use $G = (V, E)$ to denote the (directed) link graph of S. V is the set of pages (or nodes) and E is the set of directed edges (or links). We use L to denote the adjacency matrix of the graph.

$$L_{ij} = \begin{cases} 1 & if\ (i,j) \in E \\ 0 & otherwise \end{cases} \quad (1)$$

Let the authority score of the page i be $A(i)$, and the hub score of page i be $H(i)$. The mutual reinforcing relationship of the two scores is represented as follows:

$$A(i) = \sum_{(j,i)\in E} H(j) \quad (2)$$

$$H(i) = \sum_{(i,j)\in E} A(j) \quad (3)$$

We can write them in a matrix form. We use $\mathbf{A}$ to denote the column vector with all the authority scores, $\mathbf{A} = (A(1), A(2), \ldots, A(n))^T$, and use $\mathbf{H}$ to denote the column vector with all the hub scores, $\mathbf{H} = (H(1), H(2), \ldots, H(n))^T$,

$$\mathbf{A} = L^T\mathbf{H} \quad (4)$$

$$\mathbf{H} = L\mathbf{A} \quad (5)$$

To solve the problem, the widely used method is power iteration, which starts with some random values for the vectors, e.g., $A_0 = H_0 = (1, 1, 1, \ldots 1,)$. It then continues to compute iteratively until the algorithm converges.

From the formulas, we can see that the authority score estimates the importance of the content of the page, and the hub score estimates the values of its links to other pages. An authority score is computed as the sum of the scaled hub scores that point to that page. A hub score is the sum of the scaled authority scores of the pages it points to. The key idea of HITS is that a good hub points to many good authorities and a good authority is pointed by many good hubs. Thus, authorities and hubs have a mutual reinforcement relationship.

For our scenario, we have three strong clues for features in a corpus: opinion words, part-whole patterns, and the “no” pattern. Although all these three clues are not hard rules, there exist mutual enforcement relations between them. If an adjective modify many features, it is highly likely to be a good opinion word. If a feature candidate is modified by many opinion words, it is likely to be a genuine feature. The same goes with part-whole patterns, the “no” pattern, or the combination for these three clues. This kind of mutual enforcement relation can be naturally modeled in the HITS framework.

Applying the HITS algorithm: Based on the key idea of HITS algorithm and feature indicators, we can apply the HITS algorithm to obtain the feature relevance ranking. Features act as authorities and feature indicators act as hubs. Different from the general HITS algorithm, features only have authority scores and feature indicators only have hub scores in our case. They form a directed bipartite graph, which is illustrated in Figure 2. We can run the HITS algorithm on this bipartite graph. The basic idea is that if a feature candidate has a high authority score, it must be a highly-relevant feature. If a feature indicator has a high hub score, it must be a good feature indicator.

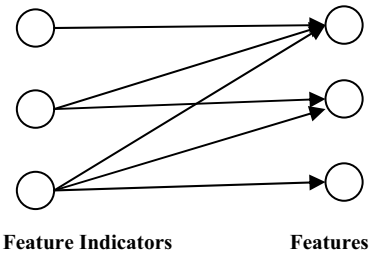


Fig. 2 Relations between feature indicators and features

3.5 Feature Ranking

Although the HITS algorithm can rank features by feature relevance, the final ranking is not only determined by relevance. As we discussed before, feature frequency is another important factor affecting the final ranking. It is highly desirable to rank those correct and frequent features at top because they are more important than the infrequent ones in opinion mining (or even other applications). With this in mind, we put everything together to present the final algorithm that we use. We use two steps:

Step 1: Compute feature score using HITS without considering frequency. Initially, we use three feature indicators to populate feature candidates, which form a directed bipartite graph. Each feature candidate acts as an authority node in the graph; each feature indicator acts as a hub node. For node s in the graph, we let H_s be the hub score and A_s be the authority score. Then, we initialize H_s and A_s to 1 for all nodes in the graph. We update the scores of H_s and A_s until they converge using power iteration. Finally, we normalize A_s and compute the score S for a feature.

Step 2: The final score function considering the feature frequency is given in Equation (6).

$$S = S(f)\log(freq(f)) \quad (6)$$

where $freq(f)$ is the frequency count of ture f, and $S(f)$ is the authority score of the candidate feature f. The idea is to push the frequent candidate features up by multiplying the log of frequency. Log is taken in order to reduce the effect of big frequency count numbers.

4 Experiments

This section evaluates the proposed method. We first describe the data sets, evaluation metrics and then the experimental results. We also compare our method with the double propagation method given in (Qiu et al., 2009).

4.1 Data Sets

We used four diverse data sets to evaluate our techniques. They were obtained from a commercial company that provides opinion mining services. Table 1 shows the domains (based on their names) and the number of sentences in each data set ("Sent." means the sentence). The data in "Cars" and "Mattress" are product reviews extracted from some online review sites. "Phone" and "LCD" are forum discussion posts extracted from some online forum sites. We split each review/post into sentences and the sentences are POS-tagged using the Brill's tagger (Brill, 1995). The tagged sentences are the input to our system.

Data Sets	Cars	Mattress	Phone	LCD
# of Sent.	2223	13233	15168	1783

Table 1. Experimental data sets

4.2 Evaluation Metrics

Besides precision and recall, we adopt the **precision@N** metric for experimental evaluation (Liu, 2006). It gives the percentage of correct features that are among the top N feature candidates in a ranked list. We compare our method's results with those of double propagation which ranks extracted candidates only by occurrence frequency.

4.3 Experimental Results

We first compare our results with double propa-

gation on recall and precision for different corpus sizes. The results are presented in Tables 2, 3, and 4 for the four data sets. They show the precision and recall of 1000, 2000, and 3000 sentences from these data sets. We did not try more sentences because manually checking the recall and precision becomes prohibitive. Note that there are less than 3000 sentences for "Cars" and "LCD" data sets. Thus, the columns for "Cars" and "LCD" are empty in Table 4. In the Tables, "DP" represents the double propagation method; "Ours" represents our proposed method; "Pr" represents precision, and "Re" represents recall.

	Cars		Mattress		Phone		LCD	
	Pr	**Re**	**Pr**	**Re**	**Pr**	**Re**	**Pr**	**Re**
DP	0.79	0.55	0.79	0.54	0.69	0.23	0.68	0.43
Ours	0.78	0.56	0.77	0.64	0.68	0.44	0.66	0.55

Table 2. Results of 1000 sentences

	Cars		Mattress		Phone		LCD	
	Pr	**Re**	**Pr**	**Re**	**Pr**	**Re**	**Pr**	**Re**
DP	0.70	0.65	0.70	0.58	0.67	0.42	0.64	0.52
Ours	0.66	0.69	0.70	0.66	0.70	0.50	0.62	0.56

Table 3. Results of 2000 sentences

	Cars	Mattress		Phone		LCD
		Pr	**Re**	**Pr**	**Re**	
DP		0.65	0.59	0.64	0.48	
Ours		0.66	0.67	0.62	0.51	

Table 4. Results of 3000 sentences

From the tables, we can see that for corpora in all domains, our method outperforms double propagation on recall with only a small loss in precision. In data sets for "Phone" and "Mattress", the precisions are even better. We also find that with the increase of the data size, the recall gap between the two methods becomes smaller gradually and the precisions of both methods also drop. However, in this case, feature ranking plays an important role in discovering important features.

Ranking comparison between the two methods is shown in Tables 5, 6, and 7, which give the precisions of top 50, 100 and 200 results respectively. Note that the experiments reported in these tables were run on the whole data sets. There were no more results for the "LCD" data beyond top 200 as there were only a limited number of features discussed in the data. So the column for "LCD" in Table 7 is empty. We rank the extracted feature candidates based on frequency for the double propagation method (DP). Using occurrence frequency is the natural way to rank features. The more frequent a feature occurs in a corpus, the more important it is. However, frequency-based ranking assumes the extracted candidates are correct features. The tables show that our proposed method (Ours) outperforms double propagation considerably. The reason is that some highly-frequent feature candidates extracted by double propagation are not correct features. Our method considers the feature relevance as an important factor. So it produces much better rankings.

	Cars	Mattress	Phone	LCD
DP	0.84	0.81	0.64	0.68
Ours	0.94	0.90	0.76	0.76

Table 5. Precision at top 50

	Cars	Mattress	Phone	LCD
DP	0.82	0.80	0.65	0.68
Ours	0.88	0.85	0.75	0.73

Table 6. Precision at top 100

	Cars	Mattress	Phone	LCD
DP	0.75	0.71	0.70	
Ours	0.80	0.79	0.76	

Table 7. Precision at top 200

5 Conclusion

Feature extraction for entities is an important task for opinion mining. The paper proposed a new method to deal with the problems of the state-of-the-art double propagation method for feature extraction. It first uses part-whole and "no" patterns to increase recall. It then ranks the extracted feature candidates by feature importance, which is determined by two factors: feature relevance and feature frequency. The Web page ranking algorithm HITS was applying to compute feature relevance. Experimental results using diverse real-life datasets show promising results. In our future work, apart from improving the current methods, we also plan to study the problem of extracting features that are verbs or verb phrases.

Acknowledgement

This work was funded by a HP Labs Innovation Research Program Award (CW165044).

References

Blair-Goldensohn, Sasha., Kerry, Hannan., Ryan, McDonald., Tyler, Neylon., George A. Reis, Jeff, Reyna. 2008. Building Sentiment Summarizer for Local Service Reviews In *Proceedings of the Workshop of NLPIX . WWW, 2008*

Brill, Eric. 1995. Transformation-Based Error-Driven Learning and Natural Language Processing: a case study in part of speech tagging. *Computational Linguistics, 1995.*

Chieu, Hai Leong and Hwee-Tou Ng. 2002. Name Entity Recognition: a Maximum Entropy Approach Using Global Information. In *Proceedings of the 6th Workshop on Very Large Corpora,* 2002.

Ding, Xiaowen., Bing Liu and Philip S. Yu. 2008. A Holistic Lexicon-Based Approach to Opinion Mining In *Proceedings of WSDM 2008*.

Girju, Roxana., Adriana Badulescu and Dan Moldovan. 2006. "Automatic Discovery of Part-Whole Relations" *Computational Linguistics ,32(1):83-135 2006*

Hu, Mingqin and Bing Liu. 2004. Mining and Summarizing Customer Reviews. In *Proceedings of KDD 2004*

Kleinberg, Jon. 1999. "Authoritative sources in hyperlinked environment" *Journal of the ACM 46 (5): 604-632 1999*

Kobayashi, Nozomi., Kentaro Inui and Yuji Matsumoto. 2007 Extracting Aspect-Evaluation and Aspect-of Relations in Opinion Mining. In *Proceedings of EMNLP*, 2007.

Lafferty, John., Andrew McCallum and Fernando Pereira. 2001 Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of ICML*, 2001.

Lin, Dekang. 1998. Dependency-based evaluation of MINIPAR. In *Proceedings of the Workshop on Evaluation of Parsing System at ICLRE* 1998.

Liu, Bing. 2006. *Web Data Mining: Exploring Hyperlinks, contents and usage data.* Springer, 2006.

Liu, Bing. 2010. Sentiment analysis and subjectivity. *Handbook of Natural Language Processing, second edition*, 2010.

Mei, Qiaozhu, Ling Xu, Matthew Wondra, Hang Su and ChengXiang Zhai. 2007. Topic Sentiment Mixture: Modeling Facets and Opinions in Weblogs. In *Proceedings of WWW, pages 171–180, 2007*.

Pang, Bo., Lillian Lee. 2008. Opinion Mining and Sentiment Analysis. *Foundations and Trends in Information Retrieval pp. 1-135 2008*

Pantel, Patrick., Eric Crestan, Arkady Borkovsky, Ana-Maria Popescu, Vishunu Vyas. 2009. Web-Scale Distributional Similarity and Entity Set Expansion. In *Proceedings of. EMNLP, 2009*

Popescu, Ana-Maria and Oren, Etzioni. 2005. Extracting product features and opinions from reviews. In *Proceedings of EMNLP, 2005.*

Qiu, Guang., Bing, Liu., Jiajun Bu and Chun Chen. 2009. Expanding Domain Sentiment Lexicon through Double Propagation. In *Proceedings of IJCAI 2009*.

Rabiner, Lawrenence. 1989. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. In *Proceedings of the IEEE, 77(2)*, *1989*.

Riloff, Ellen. 1993. Automatically Constructing a Dictionary for Information Extraction Tasks. In *Proceedings of AAAI 1993*.

Scaffidi, Christopher., Kevin Bierhoff, Eric Chang, Mikhael Felker, Herman Ng and Chun Jin. 2007. Red opal: Product-feature Scoring from Reviews. In *Proceedings of EC 2007*

Stoyanov, Veselin and Claire Cardie. 2008. Topic Identification for Fine-grained Opinion Analysis. In *Proceedings of COLING 2008*

Su, Qi., Xinying Xu., Honglei Guo, Zhili Guo, Xian Wu, Xiaoxun Zhang, Bin Swen and Zhong Su. 2008. Hidden Sentiment Association in Chinese Web Opinion Mining. In *Proceedings of WWW 2008*.

Wang, Bo., Houfeng Wang. 2008. Bootstrapping both Product Features and Opinion Words from Chinese Customer Reviews with Cross-Inducing In *Proceedings of IJCNLP 2008*

Wilson, Theresa., Janyce Wiebe and Paul Hoffmann. 2005. Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis. In *Proceedings of HLT/EMNLP 2005*

Wong, Tak-Lam., Wai Lam and Tik-Sun Wong. 2008. An Unsupervised Framework for Extracting and Normalizing Product Attributes from Multiple Web Sites In *Proceedings of SIGIR 2008*

Zhuang, Li., Feng Jing, Xiao-yan Zhu. 2006. Movie Review Mining and Summarization. In *Proceedings of CIKM 2006*

Chart Pruning for Fast Lexicalised-Grammar Parsing

Yue Zhang[a*] **Byung-Gyu Ahn**[b*] **Stephen Clark**[a*] **Curt Van Wyk**[c]
James R. Curran[d] **Laura Rimell**[a]

Computer Laboratory[a] Cambridge
Computer Science[b] Johns Hopkins
Computer Science[c] Northwestern College
School of IT[d] Sydney

{yue.zhang,stephen.clark}@cl.cam.ac.uk[a*] bahn@jhu.edu[b*]

Abstract

Given the increasing need to process massive amounts of textual data, efficiency of NLP tools is becoming a pressing concern. Parsers based on lexicalised grammar formalisms, such as TAG and CCG, can be made more efficient using supertagging, which for CCG is so effective that every derivation consistent with the supertagger output can be stored in a packed chart. However, wide-coverage CCG parsers still produce a very large number of derivations for typical newspaper or Wikipedia sentences. In this paper we investigate two forms of chart pruning, and develop a novel method for pruning complete cells in a parse chart. The result is a wide-coverage CCG parser that can process almost 100 sentences per second, with little or no loss in accuracy over the baseline with no pruning.

1 Introduction

Many NLP tasks and applications require the processing of massive amounts of textual data. For example, knowledge acquisition efforts can involve processing billions of words of text (Curran, 2004). Also, the increasing need to process large amounts of web data places an efficiency demand on existing NLP tools. TextRunner, for example, is a system that performs open information extraction on the web (Lin et al., 2009). However, the text processing that is performed by TextRunner, in particular the parsing, is rudimentary: finite-state shallow parsing technology that is now decades old. TextRunner uses this technology largely for efficiency reasons.

Many of the popular wide-coverage parsers available today operate at around one newspaper sentence per second (Collins, 1999; Charniak, 2000; Petrov and Klein, 2007). There are dependency parsers that operate orders of magnitude faster, by exploiting the fact that accurate dependency parsing can be achieved by using a shift-reduce linear-time process which makes a single decision at each point in the parsing process (Nivre and Scholz, 2004).

In this paper we focus on the Combinatory Categorial Grammar (CCG) parser of Clark and Curran (2007). One advantage of the CCG parser is that it is able to assign rich structural descriptions to sentences, from a variety of representations, e.g. CCG derivations, CCG dependency structures, grammatical relations (Carroll et al., 1998), and first-order logical forms (Bos et al., 2004). One of the properties of the grammar formalism is that it is lexicalised, associating CCG lexical categories, or CCG *supertags*, with the words in a sentence (Steedman, 2000). Clark and Curran (2004) adapt the technique of supertagging (Bangalore and Joshi, 1999) to CCG, using a standard maximum entropy tagger to assign small sets of supertags to each word. The reduction in ambiguity resulting from the supertagging stage results in a surprisingly efficient parser, given the rich structural output, operating at tens of newspaper sentences per second.

In this paper we demonstrate that the CCG parser can be made more than twice as fast, with little or no loss in accuracy. A noteworthy feature of the CCG parser is that, after the supertagging

Coling 2008: Poster Volume, pages 1471–1479,
Beijing, August 2010

stage, the parser builds a complete packed chart, storing all sentences consistent with the assigned supertags and the parser's CCG combinatory rules, *with no chart pruning whatsoever*. The use of chart pruning techniques, typically some form of beam search, is essential for practical parsing using Penn Treebank parsers (Collins, 1999; Petrov and Klein, 2007; Charniak and Johnson, 2005), as well as practical parsers based on linguistic formalisms, such as HPSG (Ninomiya et al., 2005) and LFG (Kaplan et al., 2004). However, in the CCG case, the use of the supertagger means that enough ambiguity has already been resolved to allow the complete chart to be represented.

Despite the effectiveness of the supertagging stage, the number of derivations stored in a packed chart can still be enormous for typical newspaper sentences. Hence it is an obvious question whether chart pruning techniques can be profitably applied to the CCG parser. Some previous work (Djordjevic et al., 2007) has investigated this question but with little success.

In this paper we investigate two types of chart pruning: a standard beam search, similar to that used in the Collins parser (Collins, 1999), and a more aggressive strategy in which complete cells are pruned, following Roark and Hollingshead (2009). Roark and Hollingshead use a finite-state tagger to decide which words in a sentence can end or begin constituents, from which whole cells in the chart can be removed. We develop a novel extension to this approach, in which a tagger is trained to infer the maximum length constituent that can begin or end at a particular word. These lengths can then be used in a more agressive pruning strategy which we show to be significantly more effective than the basic approach.

Both beam search and cell pruning are highly effective, with the resulting CCG parser able to process almost 100 sentences per second using a single CPU, for both newspaper and Wikipedia data, with little or no loss in accuracy.

2 The CCG Parser

The parser is described in detail in Clark and Curran (2007). It is based on CCGbank, a CCG version of the Penn Treebank developed by Hockenmaier and Steedman (2007).

The stages in the parsing pipeline are as follows. First, a POS tagger assigns a single POS tag to each word in a sentence. Second, a CCG supertagger assigns lexical categories to the words in the sentence. Third, the parsing stage combines the categories, using CCG's combinatory rules, and builds a packed chart representation containing all the derivations which can be built from the lexical categories. Finally, the Viterbi algorithm finds the highest scoring derivation from the packed chart, using the normal-form log-linear model described in Clark and Curran (2007).

Sometimes the parser is unable to build an analysis which spans the whole sentence. When this happens the parser and supertagger interact using the adaptive supertagging strategy described in Clark and Curran (2004): the parser effectively asks the supertagger to provide more lexical categories for each word. This potentially continues for a number of iterations until the parser does create a spanning analysis, or else it gives up and moves to the next sentence.

The parser uses the CKY algorithm (Kasami, 1965; Younger, 1967) described in Steedman (2000) to create a packed chart. The CKY algorithm applies naturally to CCG since the grammar is binary. It builds the chart bottom-up, starting with two-word constituents (assuming the supertagging phase has been completed), incrementally increasing the span until the whole sentence is covered. The chart is packed in the standard sense that any two equivalent constituents created during the parsing process are placed in the same equivalence class, with pointers to the children used in the creation. Equivalence is defined in terms of the category and head of the constituent, to enable the Viterbi algorithm to efficiently find the highest scoring derivation.[1] A textbook treatment of CKY applied to statistical parsing is given in Jurafsky and Martin (2000).

3 Data and Evaluation Metrics

We performed efficiency and accuracy tests on newspaper and Wikipedia data. For the newspaper data, we used the standard test sections from

[1] Use of the Viterbi algorithm in this way requires the features in the parser model to be local to a single rule application; Clark and Curran (2007) has more discussion.

```
(ncmod num hundred_1 Seven_0)
(conj and_2 sixty-one_3)
(conj and_2 hundred_1)
(dobj in_6 total_7)
(ncmod _ made_5 in_6)
(aux made_5 were_4)
(ncsubj made_5 and_2 obj)
(passive made_5)

Seven hundred and sixty-one were made in
total.
```

Figure 1: Example Wikipedia test sentence annotated with grammatical relations.

CCGbank. Following Clark and Curran (2007) we used the CCG dependencies for accuracy evaluation, comparing those output by the parser with the gold-standard dependencies in CCGbank. Unlike Clark and Curran, we calculated recall scores over all sentences, including those for which the parser did not find an analysis. For the WSJ data the parser fails on a small number of sentences (less than 1%), but the chart pruning has the effect of reducing this failure rate further, and we felt that this should be factored into the calculation of recall and hence F-score.

In order to test the parser on Wikipedia text, we created two test sets. The first, Wiki 300, for testing accuracy, consists of 300 sentences manually annotated with grammatical relations (GRs) in the style of Briscoe and Carroll (2006). An example sentence is given in Figure 1. The data was created by manually correcting the output of the parser on these sentences, with the annotation being performed by Clark and Rimell, including checks on a subset of these cases to ensure consistency across the two annotators. For the accuracy evaluation, we calculated precision, recall and balanced F-measure over the GRs in the standard way.

For testing speed on Wikipedia, we used a corpus of 2500 randomly chosen sentences, Wiki 2500. For all speed tests we measured the number of sentences per second, using a single CPU and standard hardware.

4 Beam Search

The beam search approach used in our experiments prunes all constituents in a cell having scores below a multiple (β) of the score of the highest scoring constituent for that cell.[2] The scores for a constituent are calculated using the same model used to find the highest scoring derivation. We consider two scores: the Viterbi score, which is the score of the highest scoring sub-derivation for that constituent; and the inside score, which is the sum over all sub-derviations for that constituent. We investigated the following: the trade-off between the aggressiveness of the beam search and accuracy; the comparison between the Viterbi and inside scores; and whether applying the beam to only certain cells in the chart can improve performance.

β	Speed	Gain	F-score	Gain
Baseline	43.0		85.55	
0.001	48.6	13%	85.82	0.27
0.002	54.2	26%	85.88	0.33
0.005	59.0	37%	85.73	0.18
0.01	66.7	55%	85.53	-0.02

Table 1: Accuracy and speed results using different beam values β.

δ	Speed	Gain	F-score	Gain
Baseline	43.0		85.55	
10	60.1	39%	85.55	0.00
20	70.6	64%	85.66	0.11
30	72.3	68%	85.65	0.10
40	76.4	77%	85.63	0.08
50	76.7	78%	85.62	0.07
60	74.5	73%	85.71	0.16
80	68.4	59%	85.71	0.16
100	62.0	44%	85.73	0.18
None	59.0	37%	85.73	0.18

Table 2: Accuracy and speed results for different values of δ where $\beta = 0.005$.

Table 1 shows results on Section 00 of CCGbank, using the Viterbi score to prune. As expected, the parsing speed increases as the value of β increases, since more constituents are pruned with a higher β value. The pruning is effective, with a β value of 0.01 giving a 55% speed increase with neglible loss in accuracy.[3]

[2] One restriction we apply in practice is that only constituents resulting from the application of a CCG binary rule, rather than a unary rule, are pruned.

[3] The small accuracy increase for some β values could be attributable to two factors: one, the parser may select a lower

	Speed			F-score		
Dataset	Baseline	Beam	Gain	Baseline	Beam	Gain
WSJ 00	43.0	76.4	77%	85.55	85.63	0.08
WSJ 02-21	53.4	99.4	86%	93.61	93.27	-0.34
WSJ 23	55.0	107.0	94%	87.12	86.90	-0.22
Wiki 300	35.5	80.3	126%	84.23	85.06	0.83
Wiki 2500	47.6	90.3	89%			

Table 4: Beam search results on WSJ 00, 02-21, 23 and Wikipedia texts with $\beta = 0.005$ and $\delta = 40$.

	β	δ	Speed	F-score
Baseline			24.7	85.55
inside scores	0.01		37.7	85.52
	0.001		25.3	85.79
	0.005	10	33.4	85.54
	0.005	20	39.5	85.64
	0.005	50	42.9	85.58
Viterbi scores	0.01		38.1	85.53
	0.001		28.2	85.82
	0.005	10	33.6	85.55
	0.005	20	39.4	85.66
	0.005	50	43.1	85.62

Table 3: Comparison between using Viterbi scores and inside scores as beam scores.

We also studied the effect of the beam search at different levels of the chart. We applied a selective beam in which pruning is only applied to constituents of length less than or equal to a threshold δ. For example, if $\delta = 20$, pruning is applied only to constituents spanning 20 words or less. The results are shown in Table 2. The selective beam is also highly effective, showing speed gains over the baseline (which does not use a beam) with no loss in F-score. For a δ value of 50 the speed increase is 78% with no loss in accuracy.

Note that for δ greater than 50, the speed reduces. We believe that this is due to the cost of calculating the beam scores and the reduced effectiveness of pruning for cells with longer spans (since pruning shorter constituents early in the chart-parsing process prevents the creation of many larger, low-scoring constituents later).

Table 3 shows the comparison between the inside and Viterbi scores. The results are similar, with Viterbi marginally outperforming the inside score in most cases. The interesting result from these experiments is that the summing used in calculating the inside score does not improve performance over the max operator used by Viterbi.

Table 4 gives results on Wikipedia text, compared with a number of sections from CCGbank. (Sections 02-21 provide the training data for the parser which explains the high accuracy results on these sections.) Despite the fact that the pruning model is derived from CCGbank and based on WSJ text, the speed improvements for Wikipedia were even greater than for WSJ text, with parameters $\beta = 0.005$ and $\delta = 40$ leading to almost a doubling of speed on the Wiki 2500 set, with the parser operating at 90 sentences per second.

5 Cell Pruning

Whole cells can be pruned from the chart by tagging words in a sentence. Roark and Hollingshead (2009) used a binary tagging approach to prune a CFG CKY chart, where tags are assigned to input words to indicate whether they can be the start or end of multiple-word constituents. We adapt their method to CCG chart pruning. We also show the limitation of binary tagging, and propose a novel tagging method which leads to increased speeds and accuracies over the binary taggers.

5.1 Binary tagging

Following Roark and Hollingshead (2009), we assign the binary begin and end tags separately using two independent taggers. Given the input "We like playing cards together", the pruning effects of each type of tag on the CKY chart are shown in Figure 2. In this chart, rows repre-

scoring but more accurate derivation; and two, a possible increase in recall, discussed in Section 3, can lead to a higher F-score.

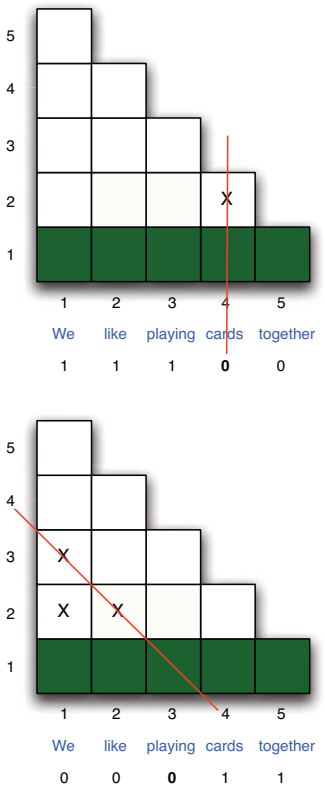


Figure 2: The pruning effect of begin (top) and end (bottom) tags; X indicates a removed cell.

sent consituent sizes and columns represent initial words of constituents. No cell in the first row of the chart is pruned, since these cells correspond to single words, and are necessary for finding a parse. The begin tag for the input word "cards" is 0, which means that it cannot begin a multi-word constituent. Therefore, no cell in column 4 can contain any constituent. The pruning effect of a binary begin tag is to cross out a column of chart cells (ignoring the first row) when the tag value is zero. Similarly, the end tag of the word "playing" is 0, which means that it cannot be the end of a multi-word constituent. Consequently cell (2, 2), which contains constituents for "like playing", and cell (1, 3), which contains constituents for "We like playing", must be empty. The pruning effect of a binary end tag is to cross out a diagonal of cells (ignoring the first row) when the tag value is zero.

We use a maximum entropy trigram tagger (Ratnaparkhi, 1996; Curran and Clark, 2003) to assign the begin and end tags. Features based on the words and POS in a 5-word window, plus the two previously assigned tags, are extracted from the trigram ending with the current tag and the five-word window with the current word in the middle. In our development experiments, both the begin and the end taggers gave a per-word accuracy of around 96%, similar to the accuracy reported in Roark and Hollingshead (2009).

Model	Speed	F-score
baseline	25.10	84.89
begin only	27.49	84.71
end only	30.33	84.56
both	33.90	84.60
oracle	33.60	85.67

Table 5: Accuracy and speed results for the binary taggers on Section 00 of CCGbank.

Table 5 shows accuracy and speed results for the binary taggers.[4] Using begin or end tags alone, the parser achieved speed increases with a small loss in accuracy. When both begin and end tags are applied, the parser achieved further speed increases, with no loss in accuracy compared to the end tag alone. Row "oracle" shows what happens using the perfect begin and end taggers, by using gold-standard constituent information from CCGbank. The F-score is higher, since the parser is being guided away from incorrect derivations, although the speed is no higher than when using automatically assigned tags.

5.2 Level tagging

A binary tag cannot take effect when there is any chart cell in the corresponding column or diagonal that contains constituents. For example, the begin tag for the word "card" in Figure 3 cannot be 0 because "card" begins a two-word constituent "card games". Hence none of the cells in the column can be pruned using the binary begin tag, even though all the cells from the third row above are empty. We propose what we call a level tagging approach to address this problem.

Instead of taking a binary value that indicates

[4]The baseline differs slightly to the previous section because gold-standard POS tags were used for the beam-search experiments.

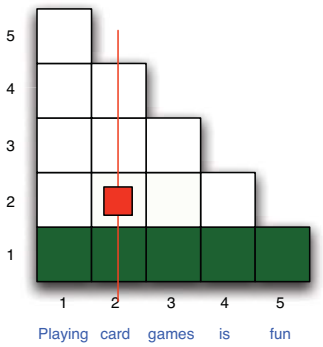


Figure 3: The limitation of binary begin tags.

whether a whole column or diagonal of cells can be pruned, a level tag (begin or end) takes an integer value which indicates the row from which a column or diagonal can be pruned in the upward direction. For example, a level begin tag with value 2 allows the column of chart cells for the word "card" in Figure 3 to be pruned from the third row upwards. A level tag (begin or end) with value 1 prunes the corresponding row or diagonal from the second row upwards; it has the same pruning effect as a binary tag with value 0. For convenience, value 0 for a level tag means that the corresponding word can be the beginning or end of any constituent, which is the same as a binary tag value 1.

A comparison of the pruning effect of binary and level tags for the sentence "Playing card games is fun" is shown in Figure 4. With a level begin tag, more cells can be pruned from the column for "card". Therefore, level tags are potentially more powerful for pruning.

We now need a method for assigning level tags to words in a sentence. However, we cannot achieve this with a straighforward classifier since level tags are related; for example, a level tag (begin or end) with value 2 implies level tags with values 3 and above. We develop a novel method for calculating the probability of a level tag for a particular word. Our mechanism for calculating these probabilities uses what we call *maxspan* tags, which can be assigned using a maximum entropy tagger.

Maxspan tags take the same values as level tags. However, the meanings of maxspan tags and level

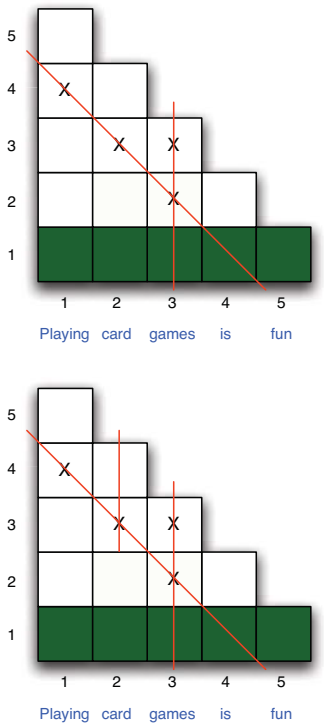


Figure 4: The pruning effect of binary (top) and level (bottom) tags.

tags are different. While a level tag indicates the row from which a column or diagonal of cells is pruned, a maxspan tag represents the size of the largest constituent a word begins or ends. For example, in Figure 3, the level end tag for the word "games" has value 3, since the largest constituent this words ends spans "playing card games".

We use the standard maximum entropy trigram tagger for maxspan tagging, where features are extracted from tag trigrams and surrounding five-word windows, as for the binary taggers. Parse trees can be turned directly into training data for a maxspan tagger. Since the level tag set is finite, we a require a maximum value N that a level tag can take. We experimented with $N = 2$ and $N = 4$, which reflects the limited range of the features used by the taggers.[5]

During decoding, the maxspan tagger uses the forward-backward algorithm to compute the probability of maxspan tag values for each word in the

[5]Higher values of N did not lead to improvements during development experiments.

Model	Speed	F-score
baseline	25.10	84.89
binary	33.90	84.60
binary oracle	33.60	85.67
level $N = 2$	32.79	84.92
level $N = 4$	34.91	84.95
level $N = 4$ oracle	47.45	86.49

Table 6: Accuracy and speed results for the level taggers on Section 00 of CCGbank.

input. Then for each word, the probability of its level tag t_l having value x is the sum of the probabilities of its maxspan t_m tag having values $1..x$:

$$P(t_l = x) = \sum_{i=1}^{x} P(t_m = i)$$

Maxspan tag values i from 1 to x represent disjoint events in which the largest constituent that the corresponding word begins or ends has size i. Summing the probabilities of these disjoint events gives the probability that the largest constituent the word begins or ends has a size between 1 and x, inclusive. That is also the probability that all the constituents the word begins or ends are in the range of cells from rows 1 to row x in the corresponding column or diagonal. And therefore that is also the probability that the chart cells above row x in the corresponding column or diagonal do not contain any constituents, which means that the column and diagonal can be pruned from row x upward. Therefore, it is also the probability of a level tag with value x.

The probability of a level tag having value x increases as x increases from 1 to N. We set a probability threshold Q and choose the smallest level tag value x with probability $P(t_l = x) \geq Q$ as the level tag for a word. If $P(t_l = N) < Q$, we set the level tag to 0 and do not prune the column or diagonal. The threshold value determines a balance between pruning power and accuracy, with a higher value pruning more cells but increasing the risk of incorrectly pruning a cell. During development we arrived at a threshold value of 0.8 as providing a suitable compromise between pruning power and accuracy.

Table 6 shows accuracy and speed results for the level tagger, using a threshold value of 0.8.

Model	Speed	F-score
baseline	36.64	84.23
binary gold	49.59	84.36
binary self 40K	48.79	83.64
binary self 200K	51.51	83.71
binary self 1M	47.78	83.75
level gold	58.23	84.12
level self 40K	54.76	83.83
level self 200K	48.57	83.39
level self 1M	52.54	83.71

Table 7: Accuracy tests on Wiki 300 comparing gold training (gold) with self training (self) for different sizes of parser output for self-training.

We compare the effect of the binary tagger and level taggers with $N = 2$ and $N = 4$. The accuracies with the level taggers are higher than those with the binary tagger; they are also higher than the baseline parsing accuracy. The parser achieves the highest speed and accuracy when pruned with the $N = 4$ level tagger. Comparing the oracle scores, the level taggers lead to higher speeds than the binary tagger, reflecting the increased pruning power of the level taggers compared with the binary taggers.

5.2.1 Final experiments using gold training and self training

In this section we report our final tests using Wikipedia data. We used two methods to derive training data for the taggers. The first is the standard method, which is to transform gold-standard parse trees into begin and end tag sequences. This method is the method that we used for all previous experiments, and we call it "gold training". In addition to gold training, we also investigate an alternative method, which is to obtain training data for the taggers from the output of the parser itself, in a form of self-training (McClosky et al., 2006). The intuition is that the tagger will learn what constituents a trained parser will eventually choose, and as long as the constituents favoured by the parsing model are not pruned, no reduction in accuracy can occur. There is the potential for an increase in speed, however, due to the pruning effect.

For gold training, we used sections 02-21 of

Model	Speed
baseline	47.6
binary gold	80.8
binary 40K	75.5
binary 200K	77.4
binary 1M	78.6
level gold	93.7
level 40K	92.8
level 200K	92.5
level 1M	96.6

Table 8: Speed tests with gold and self-training on Wiki 2500.

CCGBank (which consists of about 40K training sentences) to derive training data. For self training, we trained the parser on sections 02-21 of CCGBank, and used the parser to parse 40 thousand, 200 thousand and 1 million sentences from Wikipedia, respectively. Then we derive three sets of self training data from the three sets of parser outputs. We then used our Wiki 300 set to test the accuracy, and the Wiki 2500 set to test the speed of the parser.

The results are shown in Tables 7 and 8, where each row represents a training data set. Rows "binary gold" and "level gold" represent binary and level taggers trained using gold training. Rows "binary self X" and "level self X" represent binary and level taggers trained using self training, with the size of the training data being X sentences.

It can be seen from the Tables that the accuracy loss with self-trained binary or level taggers was not large (in the worst case, the accuracy dropped from 84.23% to 83.39%), while the speed was significantly improved. Using binary taggers, the largest speed improvement was from 47.6 sentences per second to 80.8 sentences per second (a 69.7% relative increase). Using level taggers, the largest speed improvement was from 47.6 sentences per second to 96.6 sentences per second (a 103% relative increase).

A potential advantage of self-training is the availability of large amounts of training data. However, our results are somewhat negative in this regard, in that we find training the tagger on more than 40,000 parsed sentences (the size of CCGbank) did not improve the self-training results. We did see the usual speed improvements from using the self-trained taggers, however, over the baseline parser with no pruning.

6 Conclusion

Using our novel method of level tagging for pruning complete cells in a CKY chart, the CCG parser was able to process almost 100 Wikipedia sentences per second, using both CCGbank and the output of the parser to train the taggers, with little or no loss in accuracy. This was a 103% increase over the baseline with no pruning.

We also demonstrated that standard beam search is highly effective in increasing the speed of the CCG parser, despite the fact that the supertagger has already had a significant pruning effect. In future work we plan to investigate the gains that can be achieved from combining the two pruning methods, as well as other pruning methods such as the self-training technique described in Kummerfeld et al. (2010) which reduces the number of lexical categories assigned by the supertagger (leading to a speed increase). Since these methods are largely orthogonal, we expect to achieve further gains, leading to a remarkably fast wide-coverage parser outputting complex linguistic representations.

Acknowledgements

This work was largely carried out at the Johns Hopkins University Summer Workshop and (partially) supported by National Science Foundation Grant Number IIS-0833652. Yue Zhang and Stephen Clark are supported by the European Union Seventh Framework Programme (FP7-ICT-2009-4) under grant agreement no. 247762.

References

Bangalore, Srinivas and Aravind Joshi. 1999. Supertagging: An approach to almost parsing. *Computational Linguistics*, 25(2):237–265.

Bos, Johan, Stephen Clark, Mark Steedman, James R. Curran, and Julia Hockenmaier. 2004. Wide-coverage semantic representations from a CCG parser. In *Proceedings of COLING-04*, pages 1240–1246, Geneva, Switzerland.

Briscoe, Ted and John Carroll. 2006. Evaluating the accuracy of an unlexicalized statistical parser on the PARC DepBank. In *Proceedings of the Poster Session of COLING/ACL-06*, pages 41–48, Sydney, Australia.

Carroll, John, Ted Briscoe, and Antonio Sanfilippo. 1998. Parser evaluation: a survey and a new proposal. In *Proceedings of the 1st LREC Conference*, pages 447–454, Granada, Spain.

Charniak, Eugene and Mark Johnson. 2005. Coarse-to-fine N-best parsing and maxent discriminative reranking. In *Proceedings of the 43rd Meeting of the ACL*, pages 173–180, Michigan, Ann Arbor.

Charniak, Eugene. 2000. A maximum-entropy-inspired parser. In *Proceedings of the 1st Meeting of the NAACL*, pages 132–139, Seattle, WA.

Clark, Stephen and James R. Curran. 2004. The importance of supertagging for wide-coverage CCG parsing. In *Proceedings of COLING-04*, pages 282–288, Geneva, Switzerland.

Clark, Stephen and James R. Curran. 2007. Wide-coverage efficient statistical parsing with CCG and log-linear models. *Computational Linguistics*, 33(4):493–552.

Collins, Michael. 1999. *Head-Driven Statistical Models for Natural Language Parsing*. Ph.D. thesis, University of Pennsylvania.

Curran, James R. and Stephen Clark. 2003. Investigating GIS and smoothing for maximum entropy taggers. In *Proceedings of the 10th Meeting of the EACL*, pages 91–98, Budapest, Hungary.

Curran, James R. 2004. *From Distributional to Semantic Similarity*. Ph.D. thesis, University of Edinburgh.

Djordjevic, Bojan, James R. Curran, and Stephen Clark. 2007. Improving the efficiency of a wide-coverage CCG parser. In *Proceedings of IWPT-07*, pages 39–47, Prague, Czech Republic.

Hockenmaier, Julia and Mark Steedman. 2007. CCGbank: a corpus of CCG derivations and dependency structures extracted from the Penn Treebank. *Computational Linguistics*, 33(3):355–396.

Jurafsky, Daniel and James H. Martin. 2000. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition*. Prentice Hall, New Jersey.

Kaplan, Ron, Stefan Riezler, Tracy H. King, John T. Maxwell III, Alexander Vasserman, and Richard Crouch. 2004. Speed and accuracy in shallow and deep stochastic parsing. In *Proceedings of HLT-NAACL'04*, Boston, MA.

Kummerfeld, Jonathan K., Jessika Roesner, Tim Dawborn, James Haggerty, James R. Curran, and Stephen Clark. 2010. Faster parsing by supertagger adaptation. In *Proceedings of ACL-10*, Uppsala, Sweden.

Lin, Thomas, Oren Etzioni, and James Fogarty. 2009. Identifying interesting assertions from the web. In *Proceedings of the 18th Conference on Information and Knowledge Management (CIKM 2009)*, Hong Kong.

McClosky, David, Eugene Charniak, and Mark Johnson. 2006. Effective self-training for parsing. In *Proceedings of NAACL-06*, pages 152–159, Brooklyn, NY.

Ninomiya, Takashi, Yoshimasa Tsuruoka, Yusuke Miyao, and Jun'ichi Tsujii. 2005. Efficacy of beam thresholding, unification filtering and hybrid parsing in probabilistic HPSG parsing. In *Proceedings of IWPT-05*, pages 103–114, Vancouver, Canada.

Nivre, J. and M. Scholz. 2004. Deterministic dependency parsing of English text. In *Proceedings of COLING-04*, pages 64–70, Geneva, Switzerland.

Petrov, Slav and Dan Klein. 2007. Improved inference for unlexicalized parsing. In *Proceedings of the HLT/NAACL conference*, Rochester, NY.

Ratnaparkhi, Adwait. 1996. A maximum entropy model for part-of-speech tagging. In *Proceedings of EMNLP-96*, pages 133–142, Somerset, New Jersey.

Roark, Brian and Kristy Hollingshead. 2009. Linear complexity context-free parsing pipelines via chart constraints. In *Proceedings of HLT/NAACL-09*, pages 647–655, Boulder, Colorado.

Steedman, Mark. 2000. *The Syntactic Process*. The MIT Press, Cambridge, MA.

Metaphor Interpretation and Context-based Affect Detection

Li Zhang
School of Computing
Teesside University
l.zhang@tees.ac.uk

Abstract

Metaphorical and contextual affect detection from open-ended text-based dialogue is challenging but essential for the building of effective intelligent user interfaces. In this paper, we report updated developments of an affect detection model from text, including affect detection from one particular type of metaphorical affective expression and affect detection based on context. The overall affect detection model has been embedded in an intelligent conversational AI agent interacting with human users under loose scenarios. Evaluation for the updated affect detection component is also provided. Our work contributes to the conference themes on sentiment analysis and opinion mining and the development of dialogue and conversational agents.

1 Introduction

Affect sensing from open-ended text-based natural language input is a rising research area. Zhang et al. (2008a) reported an affect detection component on detecting simple and complex emotions, meta-emotions, value judgments etc from literal expressions. Recently, metaphorical language has drawn researchers' attention since it has been widely used to provide effective vivid description. Fainsilber and Ortony (1987) commented that "an important function of metaphorical language is to permit the expression of that which is difficult to express using literal language alone". In Wallington et al's work (2008), several metaphorical affective expressions (such as animal metaphor ("X is a rat") and affects as external entities metaphor ("joy ran through me")) have been intensively studied and affect has been derived from some simple animal metaphorical expressions.

The work presented here reports developments on affect detection from one particular comparatively complex metaphorical phenomenon with affect implication, i.e. the cooking metaphor ("the lawyer grilled the witness on the stand", "I knew I was cooked when the teacher showed up at the door") (http://knowgramming.com/cooking_metaphors.htm). Since context plays an important role in the interpretation of the affect conveyed by the user during the interaction, we have used linguistic contextual analysis and cognitive emotional modeling based on Markov chain modeling and a dynamic algorithm to interpret affect from context in our application.

Our developments have been incorporated into an affect detection component, which can detect affect and emotions from literal text input and has been embedded in an intelligent conversational agent, engaged in a drama improvisation with human users under loose scenarios (school bullying and Crohn's disease). The conversational AI agent also provides appropriate responses based on the detected affect from users' input in order to stimulate the improvisation. In both scenarios, the AI agent plays a minor role in drama improvisation. E.g. it plays a close friend of the bullied victim (the leading role) in school bullying scenario, who tries to stop the bullying.

We have also analyzed affect detection performance based on previously collected (other) transcripts from user testing by calculating agreements via Cohen's Kappa between two human judges and between human judges and the AI agent with and without the new devel-

Coling 2008: Poster Volume, pages 1480–1488,
Beijing, August 2010

opment respectively in order to verify the efficiency of the metaphorical and contextual affect sensing.

The content is arranged as follows. We report relevant work in section 2 and the new developments on affect detection from the cooking metaphor in section 3. Contextual affect sensing is discussed in section 4. System evaluation and conclusion are presented in section 5.

2 Related Work

There is well-known research work in the related fields. ConceptNet (Liu and Singh, 2004) is a toolkit to provide practical textual reasoning for affect sensing for six basic emotions, text summarization and topic extraction. Shaikh et al. (2007) provided sentence-level textual affect sensing to recognize evaluations (positive and negative). They adopted a rule-based domain-independent approach, but they haven't made attempts to recognize different affective states from open-ended text input.

Although Façade (Mateas, 2002) included shallow natural language processing for characters' open-ended utterances, the detection of major emotions, rudeness and value judgements is not mentioned. Zhe and Boucouvalas (2002) demonstrated an emotion extraction module embedded in an Internet chatting environment. It used a part-of-speech tagger and a syntactic chunker to detect the emotional words and to analyze emotion intensity for the first person (e.g. 'I'). The detection focused only on emotional adjectives and first-person emotions, and did not address deep issues such as figurative expression of emotion. There is also work on general linguistic cues useful for affect detection (e.g. Craggs and Wood, 2004).

In addition, there is well-known research work on the development of emotional conversational agents. Egges et al. (2003) provided virtual characters with conversational emotional responsiveness. Aylett et al. (2006) also focused on the development of affective behavior planning for their synthetic characters. Cavazza et al. (2008) reported on a conversational agent embodied in a wireless robot to provide suggestions for users on a healthy living life-style. Hierarchical Task Networks (HTN) planner and semantic interpretation have been used in this work. The cognitive planner plays an important role in assisting with dialogue management. The user's response has also been considered for the generation of a new plan. However, the system will hesitate when open-ended user input going beyond the planner's knowledge has been used intensively during interaction. The system we present here intends to deal with such challenge.

Our work focuses on the following aspects: (1) affect detection from metaphorical expressions; (2) real-time affect sensing for basic and complex emotions in improvisational role-play situations; (3) affect detection for second and third person cases (e.g. 'you', 'she'); and (4) affect interpretation based on context profiles.

3 Further Development on Metaphorical Affect Detection

Without pre-defined constrained scripts, our original system has been developed for 14-16 year old school students to conduct creative improvisation within highly emotionally charged scenarios. Various metaphorical expressions were used to convey emotions (Kövecses, 1998), which are theoretically and practically challenging and draw our attention.

Metaphorical language can be used to convey emotions implicitly and explicitly, which also inspires cognitive semanticists (Kövecses, 1998). In our previous study (Zhang et al. 2008b; 2009), we detected affect from several comparatively simple metaphorical affective phenomena. Another type of comparatively complex metaphor has also drawn our attention, i.e. the cooking metaphor. Very often, the agent himself/herself would become the victim of slow or intensive cooking (e.g. grilled, cooked). Or one agent can perform cooking like actions towards another agent to realize punishment or torture. Examples are as follows, "he basted her with flattery to get the job", "she knew she was fried when the teacher handed back her paper".

In these examples, the suffering agents have been figuratively conceptualized as food. They bear the results of intensive or slow cooking. Thus, these agents who suffer from such cooking actions carried out by other agents tend to feel pain and sadness, while the 'cooking performing' agents may take advantage of such actions to achieve their intentions, such as persuasion, punishment or even enjoyment. The syntactic structures of some of the above exam-

ples also indicate the submissive stance of the suffering agents. E.g. in the instances, passive sentences ("he knew he was cooked when he saw his boss standing at the door") have been used to imply unwillingness and victimization of the subject agents who are in fact the objects of the cooking actions described by the verb phrases ("X + copular form + passive cooking action"). In other examples, the cooking actions have been explicitly performed by the subject agents towards the object agents to imply the former's potential willingness and enjoyment and the latter's potential suffering and pain ("A + [cooking action] + B").

Thus in our application, we focus on the above two particular types of expressions. We use Rasp (Briscoe & Carroll, 2002) to recognize user input with such syntactic structures ('A + copular form + VVN', 'A + VV0/VVD/VVZ (verb) + B'). Many sentences could possess such syntactic structures (e.g. "Lisa was bullied", "he grills Lisa", "I was hit by a car", "Lisa was given the task to play the victim role", "I steamed it" etc), but few of them are cooking metaphors. Therefore we need to resort to semantic profiles to recognize the metaphorical expressions. Rasp has also provided a syntactic label for each word in the user input. Thus the main verbs were identified by their corresponding syntactic labels (e.g. 'given' labeled as 'past participle form of lexical verbs (VVN)', 'likes' and 'grills' labeled as '-s form of lexical verbs (VVZ)') and the semantic interpretation for their base forms is discovered from WordNet (Fellbaum, 1998). Since WordNet has provided hypernyms (Y is a hypernym of X if every X is a (kind of) Y) for the general noun and verb lexicon, 'COOK' has been derived as the hypernym of the verbs' described cooking actions. E.g. 'boil', 'grill', 'steam', and 'simmer' are respectively interpreted as one way to 'COOK'. 'Toast' is interpreted as one way to 'HEAT UP' while 'cook' is interpreted as one way to 'CREAT', or 'CHEAT' etc. One verb may recover several hypernyms and in our application, we collect all of them. Another evaluation resource (Esuli and Sebastiani, 2006) is resorted to in order to recover the evaluation values of all the hypernyms for a particular verb. If some hypernyms are negative (such as 'CHEAT') and the main object of the overall input refers to first/third person cases or singular proper nouns ('him', 'her', or 'Lisa'), then the user input (e.g. "he basted her with flattery to get the job") conveys potential negative affect (e.g. pain and sadness) for the human objects and potential positive affect (e.g. persuasion or enjoyment) for the subjects. If the evaluation dictionary fails to provide any evaluation value for any hypernyms (such as 'COOK' and 'HEAT UP') of the main verbs, then we still assume that 'verbs implying COOK/HEAT UP + human objects' or 'human subjects + copular form + VVN verbs implying COOK/HEAT UP' may indicate negative emotions both for the human objects in the former and the human subjects in the latter. E.g. for the input "I was fried by the head teacher", the processing is as follows:

1. Rasp identifies the input has the following structure: 'PPIS1 (I) + copular form (was) + VVN (fried)';
2. 'Fry' (base form of the main verb) is sent to WordNet to obtain its hypernyms, which include 'COOK', 'HEAT' and 'KILL';
3. The input has the following syntactic semantic structure: 'PPIS1 (I) + copular form (was) + VVN (Hypernym: COOK)', thus it is recognized as a cooking metaphor;
4. The three hypernyms are sent to the evaluation resource to obtain their evaluation values. 'KILL' is labeled as negative while others can't obtain any evaluation values from the profile;
5. The input is transformed into: ''PPIS1 (I) + copular form (was) + VVN (KILL: negative)'
6. The subject is a first person case, then the input indicates the user who is speaking suffered from a negative action and may have a 'negative' emotional state.

Although our processing is limited to the verb metaphor examples and hasn't considered other instances like "tasty tidbits of information", it points out promising directions for figurative language processing. After our intention to improve the performance of affect sensing from individual turn-taking input, we focus on improvement of the performance using context profiles. In future work, we intend to use a metaphor ontology to recognize metaphors.

4 Affect Sensing from Context Profiles

Our previous affect detection (Zhang et al. 2008a) has been performed solely based on in-

dividual turn-taking input. Thus the context information has been ignored. However, the contextual and character profiles may influence the affect implied in the current input. In this section, we will discuss relationships between characters, linguistic contextual indicators, cognitive emotion simulation from a communication context and our approach developed based on these features to interpret affect from context.

4.1 Relationship Interpretation

Relationships between characters in drama improvisation are very crucial for the contextual affect interpretation for the emotionally ambiguous users' input. During the improvisation of each scenario, like any other drama progression, normally the recorded transcripts for creative roleplays are composed of three main improvisational sections, including the starting of the drama, the climax and the final ending. Relationships in these three drama progression stages between characters are different from one another. E.g. in the climax of the improvisation of the school bullying scenario, we normally expect very negative relationships between the bully and the bullied victim (Lisa) & her friends since the big bully is very aggressive at Lisa and her friends who try to stop the bullying. Moreover, in nearly the end of the improvisational session, sometimes the big bully feels sorry for his behavior and is cared by Lisa and her friends since he is abused by his uncle. The intense negative relationships between the big bully and Lisa & her friends are changed to those with at least less negativity or even normal relationships. Because of the creative nature of the improvisation, sometimes the bully and the victim may even have a positive relationship towards the ending of the drama improvisation.

However in our current study, we only assume consistent negative relationships between the bully and the bullied victim & her friends throughout the improvisation to simplify the processing. We will report our work on relationship interpretation using fuzzy logic to dynamically capture the changing relationships between characters as the drama progresses in the near future.

4.2 Linguistic Contextual Indicators

In our study, we noticed some linguistic indicators for contextual communication in the recorded transcripts. One useful indicator is (i) imperatives, which are often used to imply negative or positive responses to the previous speaking characters, such as "shut up", "go on then", "let's do it" and "bring it on". Other useful contextual indicators are (ii) prepositional phrases (e.g. "by who?"), semi-coordinating conjunctions (e.g. "so we are good then"), subordinating conjunctions ("because Lisa is a dog") and coordinating conjunctions ('and', 'or' and 'but'). These indicators are normally used by the current 'speaker' to express further opinions or gain further confirmation.

In addition, (iii) short phrases for questions are also used frequently in the transcripts to gain further communication based on context, e.g. "where?", "who is Dave" or "what". (iv) Character names are also normally used in the user input to indicate that the current input is intended for particular characters, e.g. "Dave go away", "Mrs Parton, say something", "Dave what has got into you?" etc. Very often, such expressions have been used to imply potential emotional contextual communication between the current speaking character and the named character. Therefore the current speaking characters may imply at least 'approval' or 'disapproval' towards the opinions/comments provided by the previous named speaking characters. Finally there are also (v) some other well known contextual indicators in Internet relay chat such as 'yeah/yes followed by a sentence ("yeah, we will see")', "I think so", 'no/nah followed by a sentence', "me too", "exactly", "thanks", "sorry", "grrrr", "hahahaha", etc. Such expressions are normally used to indicate affective responses to the previous input.

Since natural language is ambiguous and there are cases in which contextual information is required in order to appropriately interpret the affect conveyed in the input (e.g. "go on then"), our approach reported in the following integrates the above contextual linguistic indicators with cognitive contextual emotion prediction to uncover affect conveyed in emotionally ambiguous input.

4.3 Emotion Modeling in Communication Context

There are also other aspects which may influence the affect conveyed in the communication context. E.g. in our application, the affect con-

veyed by the speaking character himself/herself in the recent several turn-taking, the 'improvisational mood' that the speaking character is created, and emotions expressed by other characters, especially by the contradictory ones (e.g. the big bully), have great potential to influence the affect conveyed by the current speaking character (e.g. the bullied victim). Sometimes, the story themes or topics also have potential impact to emotions or feelings expressed by characters. For example, people tend to feel 'happy' when involved in discussions on positive topics such as harvest or raising salary, while people tend to feel 'sad' when engaged in the discussions on negative themes such as economy breakdown, tough examination etc.

In our application, although the hidden story sub-themes used in the scenarios are not that dramatic, they are still highly emotionally charged and used as the signals for potential changes of emotional context for each character. E.g. In the school bullying scenario (which is mainly about the bully, Mayid, is picking on the new comer to the school, Lisa. Lisa's friends, Elise and Dave, are trying to stop the bullying. The school teacher, Mrs Parton, also tries to find out what is going on), the director mainly provided interventions based on several main sub-themes of the story to push the improvisation forward, i.e. "Mayid starts bullying Lisa", "why Lisa is crying", "why Mayid is so nasty/a bully", "how Mayid feels when his uncle finds out about his behavior" etc. From the inspection of the recorded transcripts, when discussing the topic of "why Lisa is crying", we noticed that Mayid (the bully) tends to be really aggressive and rude, while Lisa (the bullied victim) tends to be upset and other characters (Lisa's friends and the school teacher) are inclined to show anger at Mayid. For the improvisation of the hidden story sub-theme "why Mayid is so nasty/a bully", the big bully changes from rude and aggressive to sad and embarrassed (e.g. because he is abused by his uncle), while Lisa and other characters become sympathetic (and sometimes caring) about Mayid. Usually all characters are trying to create the 'improvisational mood' according to the guidance of the hidden story sub-themes (provided via director's intervention). Therefore, the story sub-themes could be used as the indicators for potential emotional context change. The emotion patterns expressed by each character within the improvisation of each story sub-theme could be very useful for the prediction of the affect shown in a similar topic context, although the improvisation of the characters is creative within the loose scenario. It will improve the performance of the emotional context prediction if we allow more emotional profiles for each story sub-theme to be added to the training data to reflect the creative improvisation (e.g. some improvisations went deeper for a particular topic).

Therefore, a Markov chain is used to learn from the emotional context shown in the recorded transcripts for each sub-theme and for each character, and generate other possible reasonable unseen emotional context similar to the training data for each character. Markov chains are usually used for word generation. In our application, they are used to record the frequencies of several emotions showing up after one particular emotion. A matrix has been constructed dynamically for neutral and the 12 most commonly used emotions in our application (caring, arguing, disapproving, approving, grateful, happy, sad, threatening, embarrassed, angry/rude, scared and sympathetic) with each row representing the previous emotion followed by the subsequent emotions in columns. The Markov chains employ roulette wheel selection to ensure to produce a greater probability to select emotions with higher frequencies than emotions with lower occurrences. This will allow the generation of emotional context to probabilistically follow the training data, which may reflect the creative nature of the improvisation.

Then a dynamic algorithm is used to find the most resembling emotional context for any given new situation from the Markov chain's training and generated emotional contexts. I.e. by using the algorithm, a particular series of emotions for a particular story sub-theme has been regarded as the most resembling context to the test emotional situation and an emotional state is recommended as the most probable emotion for the current user input. Since the most recent affect histories of other characters and relationships between characters may also have an impact on the affect conveyed by the speaking character, the recommended affect will be further evaluated (e.g. a most recent 'insulting' input from Mayid could make Lisa 'angry').

At the training stage, first of all, the school bullying transcripts collected from previous user testing have been divided into several topic sections with each of them belonging to one of the story sub-themes. The classification of the sub-themes is mainly based on the human director's intervention which was recorded in the transcripts. Then we used two human annotators to mark up the affect of every turn-taking input in the transcripts using context inference. Thus, for each character, we have summarized a series of emotions expressed throughout the improvisation of a particular story sub-theme. Since the improvisation is creative under the loose scenario, some of the sub-themes (e.g. "why Mayid is so nasty") have been suggested for improvisation for one than once in some transcripts and some of the topics (e.g. "why Lisa is crying") are only shown in a few of the collected transcripts. We made attempts to gather as many emotional contexts as possible for each character for the improvisation of each sub-theme in order to enrich the training data.

The following is a small portion of one recorded transcript used for the training of the Markov chain. The human annotators have marked up the affect expressed in each turn-taking input.

DIRECTOR: why is Lisa crying?

Elise Brown [caring]: lisa stop cryin

Lisa Murdoch [disagree]: lisa aint crying!!!!

Dave Simons [caring]: i dunno! y u cryin lisa?

Mayid Rahim [rude]: cuz she dnt realise she is lucky to b alive

Elise Brown [angry]: beat him up! itss onlii fat..he'll go down straight away

Mayid Rahim [insulting]: lisa, y u crying? u big baby!

Mrs Parton [caring]: lisa, r u ok?

For example, the emotional context for Mayid from the above example is: 'rude' and 'insulting' (we use one letter to represent each emotional label, thus in this example, i.e. 'R I'), and in the similar way, the emotional contexts for other characters have been derived from the above example, which are used as the training data for the Markov chain for the topic "why Lisa is crying". We have summarized the emotional context for each story sub-theme for each character from 4 school bullying transcripts and used them for the training of the Markov chain. The topics considered at the training stage include: "Mayid starts bullying", "why is Lisa crying", "why is Mayid nasty/a bully" and "how does Mayid feel if his uncle knew about his behavior?"

At the test stage, our affect detection component, EMMA, is integrated with an AI agent and detects affect for each user input solely based on the analysis of individual turn-taking input itself. The above algorithms for context-based affect sensing will be activated when the affect detection component recognizes 'neutral' from the current input during the emotionally charged proper improvisation after all the characters have known each other and went on the virtual drama stage. First of all, the linguistic indicators are used to identify if the input with 'neutral' implication is a contextual-based input. E.g. we mainly focus on the checking of the five contextual implications we mentioned previously, including imperatives, prepositional phrases, conjunctions, simplified question sentences, character names, and other commonly used contextual indicators (e.g. "yeah", "I think so"). If any of the above contextual indicators exists, then we further analyze the affect embedded in the input with contextual emotion modeling reported here.

For example, we have collected the following transcript for testing. Normally the director intervened to suggest a topic change (e.g. "find out why Mayid is a bully"). Thus for a testing situation for a particular character, we use the emotion context attached with his/her user input starting right after the most recent director's intervention and ending at his/her last second input, since such a context may belong to one particular topic.

DIRECTOR: U R IN THE PLAYGROUND (indicating bullying starts)

1. Lisa Murdoch: leave me alone! [angry]
2. Mayid Rahim: WAT U GONNA DU? [neu] -> [angry]
3. Mayid Rahim: SHUT UR FAT MOUTH [angry]
4. Elise Brown: grrrrr [angry]
5. Elise Brown: im telin da dinna lady! [threatening]
6. Mayid Rahim: go on den [neutral] -> [angry]
7. Elise Brown: miss[sssssssssssssss] [neu]
8. Elise Brown: lol [happy]

9. Lisa Murdoch: mayid u gna gt banned [threatening]

10. Mayid Rahim: BY HU [neu] -> [angry]

The affect detection component detected that Lisa was 'angry' by saying "leave me alone!". It also sensed that Mayid was 'neutral' by saying "WAT U GONNA DU (what are you going to do)?" without consideration of context. From Rasp, we obtained that the input is a simplified question sentence (a linguistic contextual indicator). Thus, it implies that it could be an emotional situation caused by the previous context (e.g. previous input from Lisa) and the further processing for emotion prediction is activated. Since we don't have an emotional context yet at this stage for Mayid (the very first input from Mayid after the director's intervention), we cannot resort to the Markov chain and the dynamic algorithm currently to predict the affect. However, we could use the emotional context of other characters to predict the affect for Mayid's current input since we believe that an emotional input from a character, especially from an opponent character, has great potential to affect the emotions expressed by the current speaking character.

In the most recent chat history, there is only one input from Lisa after the director's intervention, which implied 'anger'. Since Lisa and Mayid have a negative relationship (pre-defined by character profiles), then we predict Mayid currently experiences negative emotion. Since capitalizations have been used in Mayid's input, we conclude that the affect implied in the input could be 'angry'. However, EMMA could be fooled if the affect histories of other characters fail to provide any useful indication for prediction (e.g. if Lisa implied 'neutral' in the most recent input, the interpretation of the affect conveyed by Mayid would be still 'neutral').

EMMA also detected affect for the 3rd, 4th, and 5th user input in the above example (based on individual turn-taking) until it detected 'neutral' again from the 6th input "go on den (go on then)" from Mayid. Since it is an imperative mood sentence (a linguistic contextual indicator), the input may imply a potential (emotional) response to the previous speaking character. Since we couldn't obtain the affect embedded in the imperative purely based on the analysis of the input itself, the contextual processing is required. Thus the emotional context profile for Mayid is retrieved, i.e. [angry (the 2nd input) and angry (the 3rd input)]. The Markov chain is used to produce the possible emotional context based on the training data for each sub-theme for Mayid.

The following are generated example emotional profiles for the sub-theme "Mayid starts bullying" for the Mayid character:

1. T A A N A A ['threatening, angry, angry, neutral, angry and angry']

2. N A A A ['neutral, angry, angry, and angry']

3. D A I A A A N A ['disapproval, angry, insulting, angry, angry, angry, neutral, and angry']

4. I A A N ['insulting, angry, angry and neutral']

The dynamic algorithm is used to find the smallest edit distance between the test emotional context [angry and angry] and the training and generated emotional context for the Mayid character for each sub-theme. In the above example, the second and fourth emotional sequences have the smallest edit distance (distance = 2) to the test emotional context and the former suggests 'angry' as the affect conveyed in the current input ("go on den") while the latter implies 'neutral' expressed in the current input. Thus we need to resort to the emotional context of other characters to justify the recommended affects. From the chatting log, we find that Lisa was 'angry' in her most recent input (the 1st input) while Elise was 'threatening' in her most recent input (the 5th input). Since the bully, Mayid, has a negative relationships with Lisa (being 'angry') and Elise (being 'threatening'), the imperative input ("go on den") may indicate 'angry' rather than 'neutral'. Therefore our processing adjusts the affect from 'neutral' to 'angry' for the 6th input.

In this way, by considering the linguistic contextual indicators, the potential emotional context one character was in, relationships with others and recent emotional profiles of other characters, our affect detection component has been able to inference emotion based on context to mark up the rest of the above test example transcript (e.g. Mayid being 'angry' for the 10th input). However our processing could be fooled easily by various diverse ways for affective expressions and creative improvisation (test emotional patterns not shown in the training and

generated sets). We intend to adopt better emotion simulation tools, more linguistic hints, psychological (context-based) emotional theories for further improvements. Also, our processing currently only focused on the school bullying scenario. We are on our way to extend the context-based affect sensing to the Crohn's disease scenario to further evaluate its efficiency.

5 Evaluation and Conclusion

We carried out user testing with 220 secondary school students from Birmingham and Darlington schools for the improvisation of school bullying and Crohn's disease scenarios. Generally, our previous statistical results based on the collected questionnaires indicate that the involvement of the AI character has not made any statistically significant difference to users' engagement and enjoyment with the emphasis of users' notice of the AI character's contribution throughout. Briefly, the methodology of the testing is that we had each testing subject have an experience of both scenarios, one including the AI minor character only and the other including the human-controlled minor character only. After the testing sessions, we obtained users' feedback via questionnaires and group debriefings. Improvisational transcripts were automatically recorded during the testing so that it allows further evaluation of the performance of the affect detection component.

Therefore, we produce a new set of results for the evaluation of the updated affect detection component with metaphorical and context-based affect detection based on the analysis of some recorded transcripts of school bullying scenario. Generally two human judges (not engaged in any development stage) marked up the affect of 150 turn-taking user input from the recorded another 4 transcripts from school bullying scenario (different from those used for the training of Markov chains). In order to verify the efficiency of the new developments, we provide Cohen's Kappa inter-agreements for EMMA's performance with and without the new developments for the detection of the most commonly used 12 affective states. In the school bullying scenario, EMMA played a minor bit-part character (Lisa's friend: Dave). The agreement for human judge A/B is 0.45. The inter-agreements between human judge A/B and EMMA with and without the new developments are presented in Table 1.

	Human Judge A	Human Judge B
EMMA (previous version)	0.38	0.30
EMMA (new version)	0.40	0.32

Table 1: Inter-agreements between human judges and EMMA with and without the new developments

Although further work is needed, the new developments on metaphorical and contextual affect sensing have improved EMMA's performance of affect detection in the test transcripts comparing with the previous version.

The evaluation results indicated that most of the improvements (approximately 80%) are obtained for negative affect detection based on the inference of context information. But there are still some cases: when the two human judges both believed that user inputs carried negative affective states (such as angry, threatening, disapproval etc), EMMA regarded them as neutral. One most obvious reason is that some of the previous pipeline processing (such as dealing with mis-spelling, acronyms etc, and syntactic processing from Rasp etc) failed to recover the standard user input or recognize the complex structure of the input which led to less interesting and less emotional context and may affect the performance of contextual affect sensing. (The work of Sproat et al. (2001) can point out helpful directions on this aspect.) Currently we achieved 69% average accuracy rate for the contextual affect sensing for the emotion interpretation of all the human controlled characters in school bullying scenario. We also aim to extend the evaluation of the context-based affect detection using transcripts from other scenarios. Moreover, some of the improvements (nearly 20%) in the updated affect sensing component are made by the metaphorical processing. However, since the test transcripts contained a very small number of metaphorical language phenomena comparatively, we intend to use other resources (e.g. The Wall Street Journal and other metaphorical databases (such as ATT-Meta, 2008)) to further evaluate the new development on metaphorical affect sensing.

References

ATT-Meta Project Databank: Examples of Usage of Metaphors of Mind. 2008. http://www.cs.bham.ac.uk/~jab/ATT-Meta/Databank/.

Aylett, A., Louchart, S. Dias, J., Paiva, A., Vala, M., Woods, S. and Hall, L.E. 2006. Unscripted Narrative for Affectively Driven Characters. *IEEE Computer Graphics and Applications* 26(3). 42-52.

Briscoe, E. & Carroll, J. 2002. Robust Accurate Statistical Annotation of General Text. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation*, Las Palmas, Gran Canaria. 1499-1504.

Cavazza, M., Smith, C., Charlton, D., Zhang, L., Turunen, M. and Hakulinen, J. 2008. A 'Companion' ECA with Planning and Activity Modelling. In *Proceedings of the 7th International Conference on Autonomous Agents and Multi-Agent Systems*. Portugal, 1281-1284.

Craggs, R. & Wood, M. 2004. A Two Dimensional Annotation Scheme for Emotion in Dialogue. In *Proceedings of AAAI Spring Symposium: Exploring Attitude and Affect in Text*.

Egges, A., Kshirsagar, S. & Magnenat-Thalmann, N. 2003. A Model for Personality and Emotion Simulation, In *Proceedings of Knowledge-Based Intelligent Information & Engineering Systems (KES2003)*, Lecture Notes in AI. Springer-Verlag: Berlin, 453-461.

Esuli, A. and Sebastiani, F. 2006. Determining Term Subjectivity and Term Orientation for Opinion Mining. In *Proceedings of EACL-06, 11th Conference of the European Chapter of the Association for Computational Linguistics*, Trento, IT. 193-200.

Fainsilber, L. and Ortony, A. 1987. Metaphorical uses of language in the expression of emotions. *Metaphor and Symbolic Activity*, 2(4), 239-250.

Fellbaum, C. 1998. *WordNet, an Electronic Lexical Database*. The MIT press.

Kövecses, Z. 1998. Are There Any Emotion-Specific Metaphors? In *Speaking of Emotions: Conceptualization and Expression*. Athanasiadou, A. and Tabakowska, E. (eds.), Berlin and New York: Mouton de Gruyter, 127-151.

Liu, H. & Singh, P. 2004. ConceptNet: A practical commonsense reasoning toolkit. *BT Technology Journal*, Volume 22, Kluwer Academic Publishers.

Mateas, M. 2002. Interactive Drama, Art and Artificial Intelligence. Ph.D. Thesis. School of Computer Science, Carnegie Mellon University.

Rayson, P. 2003. Matrix: A statistical method and software tool for linguistic analysis through corpus comparison. Ph.D. thesis, Lancaster University.

Shaikh, M.A.M., Prendinger, H. & Mitsuru, I. 2007. Assessing sentiment of text by semantic dependency and contextual valence analysis. In *Proceeding of ACII 2007*, 191-202.

Sproat, R., Black, A., Chen, S., Kumar, S., Ostendorf, M. and Richards, C. 2001. Normalization of Non-standard Words. *Computer Speech and Language*, 15(3), 287-333.

Wallington, A.M., Agerri, R., Barnden, J.A., Lee, M.G. & Rumbell, T. 2008. Affect Transfer by Metaphor for an Intelligent Conversational Agent. In *Procs of LREC 2008 Workshop on Sentiment Analysis: Emotion, Metaphor, Ontology and Terminology*, pp.107-113. Morocco.

Zhang, L., Barnden, J.A. Hendley, R.J. Lee, M.G. Wallington, A.M. and Wen, Z. 2008a. Affect Detection and Metaphor in E-drama. *Int. J. Continuing Engineering Education and Life-Long Learning*, Vol. 18, No. 2, 234-252.

Zhang, L., Gillies, M. & Barnden, J.A. 2008b. EMMA: an Automated Intelligent Actor in E-drama. In *Proceedings of International Conference on Intelligent User Interfaces*. 13th –16th Jan 2008. Canary Islands, Spain. pp. 409-412.

Zhang, L., Gillies, M., Dhaliwal, K., Gower, A., Robertson, D. & Crabtree, B. 2009. E-drama: Facilitating Online Role-play using an AI Actor and Emotionally Expressive Characters. *International Journal of Artificial Intelligence in Education*. Vol 19(1), pp.5-38.

Zhe, X. & Boucouvalas, A.C. 2002. Text-to-Emotion Engine for Real Time Internet Communication. In *Proceedings of International Symposium on Communication Systems, Networks and DSPs*, Staffordshire University, UK, 164-168.

Sentence Ordering with Event-Enriched Semantics and Two-Layered Clustering for Multi-Document News Summarization

Renxian Zhang **Wenjie Li** **Qin Lu**
Department of Computing, the Hong Kong Polytechnic University
{csrzhang,cswjli,csluqin}@comp.polyu.edu.hk

Abstract

We propose an event-enriched model to alleviate the semantic deficiency problem in the IR-style text processing and apply it to sentence ordering for multi-document news summarization. The ordering algorithm is built on event and entity coherence, both locally and globally. To accommodate the event-enriched model, a novel LSA-integrated two-layered clustering approach is adopted. The experimental result shows clear advantage of our model over event-agonistic models.

1 Introduction

One of the crucial steps in multi-document summarization (MDS) is information ordering, right after content selection and before sentence realization (Jurafsky and Martin, 2009:832–834). Problems with this step are the culprit for much of the dissatisfaction with automatic summaries. While textual order may guide the ordering in single-document summarization, no such guidance is available for MDS ordering.

A sensible solution is ordering sentences by enhancing coherence since incoherence is the source of disorder. Recent researches in this direction mostly focus on local coherence by studying lexical cohesion (Conroy et al., 2006) or entity overlap and transition (Barzilay and Lapata, 2008). But global coherence, i.e., coherence between sentence groups with the whole text in view, is largely unaccounted for and few efforts are made at levels higher than entity or word in measuring sentence coherence.

On the other hand, event as a high-level construct has proved useful in MDS content selection (Filatova and Hatzivassiloglou, 2004; Li et al., 2006). But the potential of event in summarization has not been fully gauged and few publications report using event in MDS information ordering. We will argue that event is instrumental for MDS information ordering, especially multi-document news summarization (MDNS). Ordering algorithms based on event and entity information outperform those based only on entity information.

After related works are surveyed in section 2, we will discuss in section 3 the problem of semantic deficiency in IR-based text processing, which motivates building event information into sentence representation. The details of such representation are provided in section 4. In section 5, we will explicate the ordering algorithms, including layered clustering and cluster-based ordering. The performance of the event-enriched model will be extensively evaluated in section 6. Section 7 will conclude the work with directions to future work.

2 Related Work

In MDS, information ordering is often realized on the sentence level and treated as a coherence enhancement task. A simple ordering criterion is the chronological order of the events represented in the sentences, which is often augmented with other ordering criteria such as lexical overlap (Conroy et al., 2006), lexical cohesion (Barzilay et al., 2002) or syntactic features (Lapata 2003).

A different way to capture local coherence in sentence ordering is the Centering Theory (CT, Grosz et al. 1995)-inspired entity-transition approach, advocated by Barzilay and Lapata (2005, 2008). In their entity grid model, syntactic roles played by entities and transitions between these syntactic roles underlie the coherence patterns between sentences and in the

Coling 2008: Poster Volume, pages 1489–1497,
Beijing, August 2010

whole text. An entity-parsed corpus can be used to train a model that prefers the sentence orderings that comply with the optimal entity transition patterns.

Another important clue to sentence ordering is the sentence positional information in a source document, or "precedence relation", which is utilized by Okazaki et al. (2004) in combination with topical clustering.

Those works are all relevant to the current work because we seek ordering clues from chronological order, lexical cohesion, entity transition, and sentence precedence. But we also add an important member to the panoply – event.

Despite its intuitive and conceptual appeal, event is not as extensively used in summarization as term or entity. Filatova and Hatzivassiloglou (2004) use "atomic events" as conceptual representations in MDS content selection, followed by Li et al. (2006) who treat event terms and named entities as graph nodes in their PageRank algorithm. Yoshioka and Haraguchi (2004) report an event reference-based approach to MDS content selection for Japanese articles. Although "sentence reordering" is a component of their model, it relies merely on textual and chronological order. Few published works report using event information in MDS sentence ordering.

Our work will represent text content at two levels: event vectors and sentence vectors. This is close in spirit to Bromberg's (2006) enriched LSA-coherence model, where both sentence and word vectors are used to compute a centroid as the topic of the text.

3 Semantic Deficiency in IR-Style Text Processing

As automatic summarization traces its root to Information Retrieval (IR), it inherits the vector space model (VSM) of text representation, according to which a sentence is treated as a bag of words or stoplist-filtered terms. The order or relation among the terms is ignored. For example,

1a) *The storm killed 120,000 people in Jamaica and five in the Dominican Republic before moving west to Mexico*.

1b) [*Dominican*, *Mexico*, *Jamaica*, *Republic*, *five*, *kill*, *move*, *people*, *storm*, *west*]

1c) [*Dominican Republic*, *Mexico*, *Jamaica*, *people*, *storm*]

1b) and 1c) are the term-based and entity-based representations of 1a) respectively. They only indicate what the sentence is about (i.e., some happening, probably a storm, in some place that affects people), but "aboutness" is a far cry from informativeness. For instance, no message about "people in which place, Mexico or Jamaica, are affected" or "what moves to where" can be gleaned from 1b) although such message is clearly conveyed in 1a). In other words, the IR-style text representation is semantically deficient.

We argue that a natural text, especially a news article, is not only about somebody or something. It also tells what happened to somebody or something in a temporal-spatial manner. A natural approach to meeting the "what happened" requirement is to introduce event.

4 Event-Enriched Sentence Representation

In summarization, an event is an activity or episode associated with participants, time, place, and manner. Conceptually, event bridges sentence and term/entity and partially fills the semantic gap in the sentence representation.

4.1 Event Structure and Extraction

Following (Li et al. 2006), we define an event E as a structured semantic unit consisting of one event term $Term(E)$ and a set of event entities $Entity(E)$. In the news domain, event terms are typically action verbs or deverbal nouns. Light verbs such as "take", "give", etc. (Tan et al., 2006) are removed.

Event entities include named entities and high-frequency entities. Named entities denote people, locations, organizations, dates, etc. High-frequency entities are common nouns or NPs that frequently participate in news events. Filatova and Hatzivassiloglou (2004) take the top 10 most frequent entities and Li et al. (2006) take the entities with frequency > 10. Rather than using a fixed threshold, we reformulate "high-frequency" as relative statistics based on (assumed) Gaussian distribution of the entities and consider those with z-score > 1 as candidate event entities.

Event extraction begins with shallow parsing and named entity recognition, analyzing each

sentence S into ordered lists of event terms $\{t_1, t_2, \ldots\}$. Low-frequency common entities are removed. If a noun is decided to be an event term, it cannot be (the head noun of) an entity.

The next step is to identify events with event terms and entities. Filatova and Hatzivassiloglou (2003) treat events as triplets with two event entities sandwiching one connector (event term). But the number restriction on entities is counterintuitive and is dropped in our method. We first identify $n + 1$ Seg_i segmented by n event terms t_j.

$$\underbrace{\ldots}_{Seg_0}\ t_1 \ldots \ldots\ t_{j-1}\ \underbrace{\ldots}_{Seg_j}\ t_j\ \underbrace{\ldots}_{Seg_j}\ t_{j+1}\ \ldots \ldots\ t_n\ \underbrace{\ldots}_{Seg_n}$$

Figure 1. Segments among Event Terms

For each t_j, the corresponding event E_j are extracted by taking t_j and the event entities in its nearest entity-containing Seg_p and Seg_q.

$E_j = [t_j, Entity(Seg_p) \cup Entity(Seg_q)]$ (Eq. 1)

where $p = \text{argmax}_{0 \le i \le j-1} Entity(Seg_i) \neq \emptyset$ and $q = \text{argmin}_{j+1 \le i \le n} Entity(Seg_i) \neq \emptyset$ if such p and q exist. 1d) is the event-extracted result of 1a).

1d) {[*killed*, [*storm*, *people*, *Jamaica*, *Dominican Republic*]], [*moving*, [*people*, *Jamaica*, *Dominican Republic*, *west*, *Mexico*]]}

From this representation, it is easy to identify the two events in sentence 1a) led by the event terms "killed" and "moving". Unlike the triplets (two named entities and one connector) in (Filatova and Hatzivassiloglou 2003), an event in our model can have an unlimited number of event entities, as is often the real case. Moreover, we can tell that the "killing" involves "people", "storm", "Jamaica", etc. and the "moving" involves "Jamaica", "Dominique Republic", etc.

The shallow parsing-based approach is admittedly coarse-grade (e.g., "storm" is missing from the "moving" event), but the extracted event-enriched representations help to alleviate the semantic deficiency problem in IR.

4.2 Event Relations

The relations between two events include event term relation and event entity relation. Two events are similar if their event terms are similar and/or their event entities are similar. Such similarities are in turn defined on the word level. For event terms, we first find the root verbs of deverbal nouns and then measure verb similarity by using the fine-grained relations provided by VerbOcean (Chklovski and Pantel, 2004), which has proved useful in summarization (Liu et al., 2007). But unlike (Liu et al., 2007), we count in all the verb relations except *antonymy* because considering two antonymous verbs as similar is counterintuitive. The other four relations – *similarity*, *strength*, *enablement*, *before* – are all considered in our measurement of verb similarity. If we denote the normalized score of two verbs on relation i as $VO_i(V_1, V_2)$ with i = 1, 2, 3, 4 corresponding to the above four relations, the term similarity of two events $\mu_t(E_1, E_2)$ is defined as in Eq. 2, where ε is a small number to suppress zeroes. ε = 0.01 if $VO_i(V_1, V_2) = 1$ and otherwise $\varepsilon = 0$.

$\mu_t(E_1, E_2) = \mu_t(Term(E_1), Term(E_2)) = 1 - \prod_{i=1}^{4}(1 - VO_i(Term(E_1), Term(E_2)) + \varepsilon)$ (Eq. 2)

Entity similarity is measured by the shared entities between two events. Li et al. (2006) define entity similarity as the number of shared entities, which may unfairly assign high scores to events with many entities in our model. So we decide to use the normalized result as shown in Eq. 3, where $\mu_e(E_1, E_2)$ denotes the event entity-based similarity between events E_1 and E_2.

$$\mu_e(E_1, E_2) = \frac{|Entity(E_1) \cap Entity(E_2)|}{|Entity(E_1) \cup Entity(E_2)|} \quad \text{(Eq. 3)}$$

$\mu(E_1, E_2)$, the score of event similarity, is a linear combination of $\mu_t(E_1, E_2)$ and $\mu_e(E_1, E_2)$.

$\mu(E_1, E_2) = \alpha_1 \times \mu_t(E_1, E_2) + (1 - \alpha_1) \times \mu_e(E_1, E_2)$ (Eq. 4)

4.3 Statistical Evidence for News Events

In this work, we introduce events as a middle-layer representation between words and sentences under the assumptions that 1) events are widely distributed in a text and that 2) they are natural clusters of salient information in a text. They guarantee the relevance of event to our task – summaries are condensed collections of salient information in source documents.

In order to confirm them, we scan the whole dataset in our experiment, which consists of 42 200w human extracts and 39 400w human extracts for the DUC 02 multi-document extract task. Detailed information about the dataset can be found in Section 6. Table 1 lists the statistics.

	200w	**400w**	**200w + 400w**	**Source Docs**
Entity/Sent	8.78	8.48	8.47	6.01
Entity/Word	0.34	0.33	0.33	0.30
Event/Sent	2.43	2.26	2.28	1.42

Event/Word	0.09	0.09	0.09	0.07
Sents with events/Sents	86.9%	85.1%	84.6%	71.3%

Table 1. Statistics from DUC 02 Dataset

There are on average 1.42 events per sentence in the source documents, and more than 70% of all the sentences contain events. The high event density confirms our first assumption about the distribution of events. For the 200w+400w category consisting of all the human-selected sentences, there are on average 2.28 events per sentence, a 60% increase from the same ratio in the source documents. The proportion of event-containing sentences reaches 84.6%, 13% higher than that in the source documents. Such is evidence that events count into the extract-worthiness of sentences, which confirms our second assumption about the relevance of events to summarization. The data also show higher entity density in the extracts than in the source documents. As entities are still reliable and domain-independent clues of salient content, we will consider both event and entity in the following ordering algorithm.

5 MDS Sentence Ordering with Event and Entity Coherence

In this section, we discuss how event can facilitate MDS sentence ordering with layered clustering on the event and sentence levels and then how event and entity information can be integrated in a coherence-based algorithm to order sentences based on sentence clusters.

5.1 Two-layered Clustering

After sentences are represented as collections of events, we need to vectorize events and sentences to facilitate clustering and cluster-based sentence ordering.

For a document set, event vectorization begins with aggregating all the event terms and entities in a set of **event units** (*eu*). Given m distinct event terms, n distinct named entities, and p distinct high-frequency common entities, the $m + n + p$ *eu*'s are a concatenation of the event terms and entities such that eu_i is an event term for $1 \le i \le m$, a named entity for $m + 1 \le i \le m + n$, and a high-frequency entity for $m + n + 1 \le i \le m + n + p$. The *eu*'s define the $m + n + p$ dimensions of an event vector in an eu-by-event matrix $\boldsymbol{E} = [e_{ij}]$, as shown in Figure 2.

$$\begin{array}{l} \\ eu_1 \\ \cdots \\ eu_m \\ \cdots \\ eu_{m+n} \\ \cdots \\ eu_{m+n+p} \end{array} \overbrace{\begin{bmatrix} e_{11} & \cdots & e_{1q} \\ \vdots & \ddots & \vdots \\ e_{m1} & \cdots & e_{mq} \\ \vdots & & \vdots \\ e_{m+n,1} & \cdots & e_{m+n,q} \\ \vdots & & \vdots \\ e_{m+n+p,1} & \cdots & e_{m+n+p,q} \end{bmatrix}}^{E_1, E_2, \ldots E_q}$$

Figure 2. eu-by-Event Matrix

We further define $Entity_N(E_j)$ and $Entity_H(E_j)$ to be the set of named entities and set of high-frequency entities of E_j. Then,

$$e_{ij} = \begin{cases} \mu_t(eu_i, Term(E_j)) & 1 \le i \le m \\ \dfrac{\sum_{e \in Entity_N(E_j)} v_n(eu_i, e)}{|Entity_N(E_j)|} & m + 1 \le i \le m + n \\ \dfrac{\sum_{e \in Entity_H(E_j)} v_h(eu_i, e)}{|Entity_H(E_j)|} & m + n + 1 \le i \le m + n + p \end{cases} \quad \text{(Eq. 5)}$$

$$v_n(w_1, w_2) = \begin{cases} 2 & w_1 \text{ is identical to } w_2 \\ 1 & w_1\ (w_2) \text{ is a part of } w_2\ (w_1) \text{ or they are in a hypernymy / holonymy relationship} \\ 0 & \text{otherwise} \end{cases} \quad \text{(Eq. 6)}$$

$$v_h(w_1, w_2) = \begin{cases} 1 & w_1 \text{ is identical to } w_2 \\ 0.5 & w_1 \text{ are } w_2 \text{ are synonyms} \\ 0 & \text{otherwise} \end{cases} \quad \text{(Eq. 7)}$$

In Eq. 5, $\mu_t(w_1, w_2)$ is defined as in Eq. 2. Both the entity-based $v_n(w_1, w_2)$ and $v_h(w_1, w_2)$ are measured in terms of total equivalence (identity) and partial equivalence. For named entities, partial equivalence applies to structural subsumption (e.g., "Britain" and "Great Britain") and hypernymy/holonymy (e.g., "South Africa" and "Zambia"). For common entities, it applies to synonymy (e.g., "security" and "safety"). Partial equivalence is considered because of the lexical variations frequently employed in journalist writing. The named entity scores are doubled because they represent the essential elements of a news story.

Since the events are represented as vectors, sentence vectorization based on events is not as straightforward as on entities or terms. In this work we propose a novel approach of **two-layered clustering** for the purpose. The basic idea is clustering events at the first layer and then using event clusters as a feature to vectorize and cluster sentences at the second

layer. Hard clustering of events, such as K-means, not only results in binary values in event vectors and data sparseness but also is inappropriate. For example, if EC_1 clusters events all with event terms similar to t^* and EC_2 clusters events all with event entity sets similar to e^* (a set), what about event $\{t^*, e^*\}$? Assigning it to either EC_1 or EC_2 is problematic as it is partially similar to both. So we decide to do soft clustering at the first layer.

A well-studied soft clustering technique is the Expectation-Maximization (EM) algorithm which iteratively estimates the unknown parameters in a probability mixture model. We assume a Gaussian mixture model for the q event vectors $\boldsymbol{V}_1$, $\boldsymbol{V}_2$, …, $\boldsymbol{V}_q$, with hidden variables $\boldsymbol{H}_i$, initial means $\boldsymbol{M}_i$, priors $\boldsymbol{\pi}_i$, and covariance matrix $\boldsymbol{C}_i$. The E-step is to calculate the hidden variables $\boldsymbol{H}_i^t$ for each $\boldsymbol{V}_t$ and the M-step re-estimates the new priors $\boldsymbol{\pi}_i'$, means $\boldsymbol{M}_i'$, and covariance matrix $\boldsymbol{C}_i'$. We iterate the two steps until the log-likelihood converges within a threshold = 10^{-6}. The performance of the EM algorithm is sensitive to the initial means, which are pre-computed by a conventional K-means.

In a preliminary study, we found that the event vectors display pronounced sparseness. A solution to this problem in an effort to leverage the latent "event topics" among *eu*'s is the Latent Semantic Analysis (LSA, Landauer and Dumais, 1997) approach. We apply LSA-style dimensionality reduction to the eu-by-event matrix $\boldsymbol{E}$ by doing Singular Value Decomposition (SVD). A problem is with the number h of the largest singular values, which affects the performance of dimensionality reduction. In this work, we adopt a utility-based metric to find the best h^* by maximizing intra-cluster similarity (Φ_h) and minimizing inter-cluster similarity (Ψ_h) corresponding to the h-dimensionality reduction

$$h^* = \text{argmax}_h \ \Phi_h/\Psi_h \qquad \text{(Eq. 8)}$$

Φ_h is defined as the mean of average cluster similarities measured by cosine distance and Ψ_h is the mean of cluster centroid similarities. Because the EM clustering assigns a probability to every event vector, we also take those probabilities into account when calculating Φ_h and Ψ_h.

Based on the EM clustering of events, we vectorize a sentence by summing up the probabilities of its constituent event vectors over all event clusters (ECs) and obtaining an EC-by-sentence (S_n) matrix $\boldsymbol{S} = [s_{ij}]$.

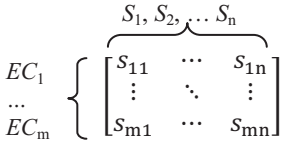


Figure 3. EC-by-Sentence Matrix

$s_{ij} = \sum_{E_r \in S_j} P(\overrightarrow{E_r}|EC_i))$ where $\overrightarrow{E_r}$ is E_r's vector.

At the sentence layer, hard clustering is sufficient because we need definitive, not probabilistic, membership information for the next step – sentence ordering. We use K-means for the purpose. The LSA-style dimensionality reduction is still in order as possible performance gain is expected from the discovery of latent EC "topics". The decision of the best dimensionality is the same as before, except that no probabilities are included.

5.2 Coherence-Based Sentence Ordering

Our ordering algorithm is based on sentence clusters, which is designed on the observation that human writers and summarizers organize sentences by blocks (paragraphs). Sentences within a block are conceptually close to each other and adjacent sentences cohere with each other. Local coherence is thus realized within blocks. On the other hand, blocks are not randomly ordered. Two blocks are put next to each other if their contents are close enough to ensure text-level coherence. So text-level, or global coherence is realized among blocks.

We believe in MDNS, the block-style organization is a sensible strategy taken by human extractors to sort sentences from different sources. Sentence clusters are simulations of such blocks and our ordering algorithm will be based on local coherence and global coherence described above.

First we have to pinpoint the leading sentence for an extract. Using the heuristic of time and textual precedence, we first generate a set of possible leading sentences $L = \{L_i\}$ as the intersection of the document-leading extract sentence set L_{Doc} and the time-leading sentence set L_{Time}. Note that $|L_{Doc}|$ = the number of documents, L_{Time} is in fact a sentence collection of time-leading documents, and $L_{Doc} \cap L_{Time} \neq \emptyset$.

If L is a singleton, finding the leading sentence S_L is trivial. If not, S_L is decided to be the sentence in L most similar to all the other sentences in the extract sentence set P so that it qualifies as a good topic sentence.

$$S_L = \text{argmax}_{L_i \in L} \sum_{L' \in P \setminus \{L_i\}} Sim_{\mu+\nu}(L_i, L') \quad \text{(Eq. 9)}$$

where $Sim_{\mu+\nu}(S_1, S_2)$ is the similarity between S_1 and S_2 in terms of their event similarity $\mu(S_1, S_2)$ and entity similarity $\nu(S_1, S_2)$. $\mu(S_1, S_2)$ is an extended version of $\mu(E_1, E_2)$ (Eq. 4) by averaging the $\mu_t(E_i, E_j)$ and $\mu_e(E_i, E_j)$ for all (E_i, E_j) pairs in $S_1 \times S_2$.

$$\mu(S_1, S_2) = \alpha_2 \times \frac{\sum_{E_i \in S_1, E_j \in S_2} \mu_t(E_i, E_j)}{|Event(S_1) \times Event(S_2)|} + (1 - \alpha_2) \times \frac{\sum_{E_i \in S_1, E_j \in S_2} \mu_e(E_i, E_j)}{|Event(S_1) \times Event(S_2)|} \quad \text{(Eq. 10)}$$

where $Event(S)$ is the set of all events in S. Next, $\nu(S_1, S_2)$ is the cosine similarity between their entity vectors $\overrightarrow{S_1}$ and $\overrightarrow{S_2}$ with entity weights constructed according to Eq. 6 and 7. Then,

$$Sim_{\mu+\nu}(S_1, S_2) = \alpha_3 \times \mu(S_1, S_2) + (1 - \alpha_3) \times \nu(S_1, S_2) \quad \text{(Eq. 11)}$$

After the leading sentence is determined, we identify the leading cluster it belongs to and our local coherence-based ordering starts with this cluster. We adopt a greedy algorithm, which selects each time from the unordered sentence set a sentence that best coheres with the sentence just selected, called **anchor sentence**.

Matching each candidate sentence with the anchor sentence only in terms of $Sim_{\mu+\nu}$ would assume that the sentences are isolated and decontextualized. But the anchor sentence did not come from nowhere and in order to find its best successor, we should also seek clues from its source context, which is inspired by the "sentence precedence" by Okazaki et al. (2004).

More formally, given an anchor sentence S_i at the end of the ordered sentence list, we select the next best sentence S_{i+1} according to their **associative similarity** and **substitutive similarity**, two crucial measures invented by us.

Associative similarity $Sim_{ASS}(S_i, S_j)$ measures how S_i and S_j associate with each other in terms of their event and entity coherence, which almost is $Sim_{\mu+\nu}(S_i, S_j)$. But to better capture the transition between entities and the flow of topic, we also consider a topic-continuity score $tc(S_i, S_j)$ according to the Centering Theory. If the topic continuity is measured in terms of entity change, local coherence can be captured by the centering transitions (*CB* and *CP*) in adjacent sentences. Based on (Taboada and Wiesemann, 2009), we assign 0.2 to the *Establish* and *Continue* transitions, 0.1 to *Smooth Shift* and *Retain*, and 0 to other centering transitions.

Since $tc(S_i, S_j)$ only applies to entities, it is treated as a bonus affiliated to $\nu(S_i, S_j)$.

$$Sim_{ASS}(S_i, S_j) = \alpha_4 \times \mu(S_i, S_j) + (1 - \alpha_4) \times \nu(S_i, S_j) \times (1 + tc(S_i, S_j)) \quad \text{(Eq. 12)}$$

Substitutive similarity accommodates what we earlier emphasized about the "source context" of the extracted sentences by measuring to what degree S_i and S_j resemble each other's relevant source context. More formally, let $LC(S_i)$ and $RC(S_i)$ be the left and right source contexts of S_i respectively, and the substitutive similarity $Sim_{SUB}(S_i, S_j)$ is defined as follows.

$$Sim_{SUB}(S_i, S_j) = Sim_{\mu+\nu}(S_i, LC(S_j)) + Sim_{\mu+\nu}(RC(S_i), S_j) \quad \text{(Eq. 13)}$$

In this work, we simply take $LC(S_i)$ and $RC(S_i)$ to be the left adjacent sentence and right adjacent sentence of S_i in the source document. Note that $tc(S_i, S_j)$ does not apply here. In view of the chronological order widely accepted in MDS ordering, a time penalty, $tp(S_i, S_j)$, is used to discount the score by 0.8 if S_i's document date is later than S_j's document date. Finally, Eq. 14 summarizes our intra-cluster ordering method in a sentence cluster SC_k.

$$S_{i+1} = \text{argmax}_{S_j \in SC_k \setminus \{S_i\}} \left(\alpha_5 \times Sim_{ASS}(S_i, S_j) + (1 - \alpha_5) \times Sim_{SUB}(S_i, S_j)\right) \times tp(S_i, S_j) \quad \text{(Eq. 14)}$$

After all the sentences in the current sentence cluster are ordered, we move on by considering the similarity of sentence clusters. Given a processed sentence cluster SC_i, the next best sentence cluster SC_{i+1} is the one that maximizes the cluster similarity $Sim_{CLU}(SC_i, SC_j)$ among the set of all clusters U. Since clusters are collections of sentences, their similarity is the mean of cross-cluster pairwise sentence similarities, each calculated according to Eq. 14. Eq. 15 shows how SC_{i+1} is computed.

$$SC_{i+1} = \text{argmax}_{SC_j \in U \setminus \{SC_i\}} Sim_{CLU}(SC_i, SC_j) \quad \text{(Eq. 15)}$$

This is how we incorporate (block-style) global coherence into MDS sentence ordering. Starting from the second chosen sentence cluster, we choose the first sentence in the current cluster with reference to the last sentence in the previous processed cluster and apply Eq. 14. We continue the whole process until all the extract sentences are ordered.

6 Evaluation

In this section, we report the experimental result on the DUC 02 dataset.

6.1 Data

We use the dataset of the DUC 02 summarization track for MDS because it includes an extraction task for which model extracts are provided. For every document set, 2 model extracts are provided each for the 200w and 400w length categories. We use 1 randomly chosen model extract per document set per length category as the gold standard.

We intended to use all the 59 document sets on DUC 02 but found that for some categories, both model extracts contain material from sections such as the *title*, *lead*, or even *byline*. Those extracts are incompatible with our design tailored for news body extracts. Therefore we have to filter them and retain only those extracts with all units selected from the news body. As a result, we collect 42 200w extracts and 39 400w extracts as our experimental dataset.

6.2 Peer Orderings

We evaluate the role played by various key elements in our approach, including event, topic continuity, time penalty, and LSA-style dimensionality reduction. In addition, we produce a random ordering and a baseline ordering according to chronological and textual order only. Table 2 lists the 9 peer orderings to be evaluated, with their codes.

A	Random
B	Baseline (time order + textual order)
C	Entity only (no LSA)
D	Event only (no LSA)
E	Entity + Event – topic continuity (no LSA)
F	Entity + Event – time penalty (no LSA)
G	Entity + Event (no LSA)
H	Entity + Event (event clustering LSA)
I	Entity + Event (event + sentence clustering LSA)

Table 2. Peer Orderings

6.3 Metrics

A popular metric used in sequence evaluation is Kendall's τ (Lapata, 2006), which measures ordering differences in terms of the number of adjacent sentence inversions necessary to convert a test ordering to the reference ordering.

$$\tau = 4m/(n(n-1)) \qquad \text{(Eq. 16)}$$

where m is the number of inversions described above and n is the total number of sentences.

The second metric we use is the Average Continuity (AC) developed by Bollegala et al. (2006), which captures the intuition that the ordering quality can be estimated by the number of correctly arranged continuous sentences.

$$AC = \exp\left(\frac{1}{k-1}\sum_{n=2}^{k}\log(P_n + \varepsilon)\right) \qquad \text{(Eq. 17)}$$

where k is the maximum number of continuous sentences, ε is a small value in case $P_n = 1$. P_n, the proportion of continuous sentences of length n in an ordering, is defined as $m/(N - n + 1)$ where m is the number of continuous sentences of length n in both the test and reference orderings and N is the total number of sentences. We set $k = 4$ and $\varepsilon = 0.01$.

6.4 Result

We empirically determine all the parameters (α_i) and produce all the peer orderings. Table 3 lists the result, where we also show the statistical significance between the full model peer ordering "I" and all other versions, marked by * ($p < .05$) and ** ($p < .01$) on a two-tailed t-test.

Peer Code	200w		400w	
	Kendall's τ	*AC*	Kendall's τ	*AC*
A	0.014**	0.009**	-0.019**	0.004**
B	0.387	0.151*	0.259**	0.151*
C	0.369*	0.128*	0.264*	0.156*
D	0.380	0.163	0.270*	0.158*
E	0.375*	0.156*	0.267*	0.157*
F	0.388	0.159*	0.264*	0.157*
G	0.385	0.158*	0.269*	0.162
H	0.384	0.164	0.292*	0.170
I	**0.395**	**0.170**	**0.350**	**0.176**

Table 3. Evaluation Result

Almost all versions with entity and event information outperform the baseline. The LSA-style dimensionality reduction proves effective for our task, as the full model (Peer I) ranks first and significantly beats versions without event information, topic continuity, or LSA. Applying LSA to both event and sentence clustering is better than applying it only to event clustering (Peer H), which produces unstable results and is sometimes outperformed by no-LSA versions (Peer G).

Event (Peer D) proves to be more valuable than entity (Peer C) as the event-only versions outperform the entity-only version in all categories, which is predicable because events

1) Thursday's **acquittals** in the McMartin Pre-School **molestation** case outraged parents who said prosecutors botched it, while those on the defense side proclaimed a triumph of justice over hysteria and hype.
2) Originally, there were seven defendants, including Raymond Buckey's sister, Peggy Ann Buckey, and Virginia McMartin, the founder of the school, mother of Mrs. Buckey and grandmother of Raymond Buckey.
3) Seven jurors who spoke with reporters in a joint news conference after **acquitting** Raymond Buckey and his mother, Peggy McMartin Buckey, on 52 **molestation** charges Thursday said they felt some children who testified may have been **molested** _ but not at the family-run McMartin Pre-School.
4) ``The children were never allowed to say in their own words what happened to them,'' said juror John Breese.
5) Ray Buckey and his mother, Peggy McMartin Buckey, were found not guilty Thursday of **molesting** children at the family-run McMartin Pre-School in Manhattan Beach, a verdict which brought to a close the longest and costliest criminal trial in history .
6) As it becomes apparent that McMartin cases will stretch out for years to come, parents and the former criminal defendants alike are trying to **resign** themselves to the inevitability that the matter may be one they can never leave behind.

Figure 4. Extract sentences of d80ae, 200w

are high-level constructs that incorporate most of the document-level important entities.

When entity is used, extra bonus can be gained from topic continuity concerns from CT (Peer E vs. Peer G) because the centering transition effectively captures the coherence pattern between adjacent sentences. The effect of the chronological order seems less clear (Peer F vs. P) as removing it hurts longer extracts rather than short extracts. Therefore chronological clues are more valuable for arranging more sentences from the same source document.

Our ordering algorithm achieves even better result with long extracts because the importance of order and coherence grows with text length. Measured by Kendall's τ, the full model ordering in the 400w category is significantly better than all other orderings.

For a qualitative evaluation, we select the 200w extract d080ae and list all the sentences in Figure 4. The event terms are boldfaced and the event entities are underlined.

Limited by space, let's focus on the baseline (1 2 3 4 5 6), entity-only (3 5 2 4 6 1), and full-model versions (3 5 4 2 1 6). The news extract is about the acquitting of child molesters. Both the "acquitting" and "molesting" events are found in 1) and 3) but only the latter qualifies as the topic sentence because it contains important event entities. Choosing 3) instead of 1) as the leading sentence shows the advantage of our event-enriched model over the baseline. The same choice is made by the entity-only version because 3) happens to be also entity-intensive. In order to see the advantage of the full model over the entity-only model, let's consider 2) and 4). 2) is chosen by the entity-only model after 5) because of the heavy entity overlap between 5) and 2). But semantically, 2) is not as close to 5) as 4) because only 4) contains entities for both the "acquitting" ("juror") and "molesting" ("children") events and intuitively, 4) continues the main trial-acquittal event topic but 2) supplies only secondary information. We examined the sentence clusters before the ordering and found that 3), 5), and 4) are clustered together only by the full model, leading to better coherence, locally and globally.

7 Conclusion and Future Work

We set out by realizing the semantic deficiency of IR and propose a low-cost approach of building event semantics into sentence representation. Event extraction relies on shallow parsing and external knowledge sources. Then we propose a novel approach of two-layered clustering to use event information, coupled with LSA-style dimensionality reduction. MDS sentence ordering is guided by local and global coherence to simulate the block-style writing and is realized by a greedy algorithm. The evaluation shows clear advantage of our event-enriched model over baseline and event-agonistic models, quantitatively and qualitatively.

The extraction approach can be refined by deep parsing and rich verb (frame) semantics. In a follow-up project, we will expand our dataset and experiment with more data and incorporate human evaluation in comparative tasks.

Acknowledgment

The work described in this paper was partially supported by a grant from the HK RGC (Project Number: PolyU5217/07E).

References

Barzilay, R., Elhadad, N., and McKeown, K. 2002. Inferring Strategies for Sentence Ordering in Multidocument News Summarization. *Journal of Artificial Intelligence Research*, 17:35–55.

Barzilay, R., and Lapata, M. 2005. Modeling Local Coherence: An Entity-based Approach. In *Proceedings of the 43rd Annual Meeting of the ACL*, 141-148. Ann Arbor.

Barzilay, R., and Lapata, M. 2008. Modeling Local Coherence: An Entity-Based Approach. *Computational Linguistics*, 34:1–34.

Bollegala, D, Okazaki, N., and Ishizuka, M. 2006. A Bottom-up Approach to Sentence Ordering for Multi-document Summarization. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL*, 385–392. Sydney, Australia.

Bromberg, I. 2006. Ordering Sentences According to Topicality. Presented at the Midwest Computational Linguistics Colloquium.

Chklovski, T., and Pantel, P. 2004. VerbOcean: Mining the Web for Fine-Grained Semantic Verb Relations. In *Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP-04)*. 11–13. Barcelona, Spain.

Conroy, J. M., Schlesinger, J. D., and Goldstein, J. 2006. CLASSY Tasked Based Summarization: Back to Basics. In *proceedings of the Document Understanding Conference (DUC-06)*.

Filatova, E., and Hatzivassiloglou, V. 2003. Domain-independent detection, extraction, and labeling of atomic events. In *Proceedings of RANLP*, 145–152, Borovetz, Bulgaria.

Filatova, E., and Hatzivassiloglou, V. 2004. Event-Based Extractive Summarization. In *Proceedings of the ACL-04*, 104–111.

Grosz, B. J., Aravind K. J., and Scott W. 1995. Centering: A framework for Modeling the Local Coherence of Discourse. *Computational Linguistics*, 21(2):203–225.

Jurafsky D., and Martin, J. H. 2009. *Speech and Language Processing, Second Edition*. Upper Saddle River, NJ: Pearson Education International.

Landauer, T., and Dumais, S. 1997. A solution to Plato's problem: The latent semantic analysis theory of the acquisition, induction, and representation of knowledge. *Psychological Review*, 104.

Lapata, M. 2003. Probabilistic Text Structuring: Experiments with Sentence Ordering. In *Proceedings of the Annual Meeting of ACL*, 545-552. Sapporo, Japan.

Li, W., Wu, M., Lu, Q., Xu, W., and Yuan, C. 2006. Extractive Summarization Using Inter- and Intra-Event Relevance. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL*, 369–376. Sydney.

Liu, M., Li, W., Wu, M., and Lu, Q. 2007. Extractive Summarization Based on Event Term Clustering. In *Proceedings of the ACL 2007 Demo and Poster Sessions*, 185–188. Prague.

Okazaki, N., Matsuo, Y., and Ishizuka, M. 2004. Improving Chronological Ordering by Precedence Relation. In *Proceedings of 20th International Conference on Computational Linguistics (COLING 04)*, 750–756.

Taboada, M., and Wiesemann, L., Subjects and topics in conversation. *Journal of Pragmatics* (2009), doi:10.1016/j.pragma.2009.04.009.

Tan, Y. F., Kan, M., and Cui, H. 2006. Extending corpus-based identification of light verb constructions using a supervised learning framework. In *Proceedings of the EACL 2006 Workshop on Multi-word-expressions in a multilingual context*, 49–56, Trento, Italy.

Yoshioka, M., and Haraguchi, M. 2004. Multiple News Articles Summarization Based on Event Reference Information. In *Working Notes of NTCIR-4*, Tokyo.

Automatic Temporal Expression Normalization with Reference Time Dynamic-Choosing

Xujian Zhao, Peiquan Jin, and Lihua Yue
School of Computer Science and Technology
University of Science and Technology of China
nonozxj@mail.ustc.edu.cn, {jpq,llyue}@ustc.edu.cn

Abstract

Temporal expressions in texts contain significant temporal information. Understanding temporal information is very useful in many NLP applications, such as information extraction, documents summarization and question answering. Therefore, the temporal expression normalization which is used for transforming temporal expressions to temporal information has absorbed many researchers' attentions. But previous works, whatever the hand-crafted rules-based or the machine-learnt rules-based, all can not address the actual problem about temporal reference in real texts effectively. More specifically, the reference time choosing mechanism employed by these works is not adaptable to the universal implicit times in normalization. Aiming at this issue, we introduce a new reference time choosing mechanism for temporal expression normalization, called reference time dynamic-choosing, which assigns the appropriate reference times to different classes of implicit temporal expressions dynamically when normalizing. And then, the solution to temporal expression defuzzification by scenario dependences among temporal expressions is discussed. Finally, we evaluate the system on a substantial corpus collected by Chinese news articles and obtained more promising results than compared methods.

1 Introduction

Temporal expression normalization is very important for temporal information processing because it is in charge of transforming temporal expressions in surface texts to temporal information behind surface texts. Temporal information is defined as the knowledge about time or duration, which can be abstracted into some objects defined as temporal attributes in TIMEX2 Standard [Ferro et al., 2005]. Human being can take temporal relation reasoning and anchor events on the time line with this information. Meanwhile, temporal expressions are defined as chunks of texts which convey explicit or implicit temporal information. So TERN evaluation plan[1] gives the task of temporal expression normalization that is annotating the appropriate temporal attributes for each temporal expression in texts. For example, a simple temporal expression, "May 1, 2009", can be normalized as <TIMEX2 VAL = "2009-05-01"> May 1, 2009 </TIMEX2>.

Unfortunately, temporal expressions in real texts are more complicated because they contain a large number of Implicit Times besides Explicit Times. Here,

(1) *Explicit Time*: Explicit Time can directly be laid in the timeline. Basically, it is a direct entry in the timeline and need not to be transformed. E.g., "May 1, 2009".

(2) *Implicit Time*: Implicit Time can be mapped as an entry in the timeline with help of real contexts and some predefined knowledge and need to be transformed. E.g., "May 1", "tomorrow" and "two day ago".

Consequently, temporal expression normalization is mainly aiming at Implicit Times that

[1] http://timex2.mitre.org/tern.html

Coling 2008: Poster Volume, pages 1498–1506,
Beijing, August 2010

need to be transformed with referring to some specific times. However, the previous works on temporal expression normalization which basically adopt two mechanisms for choosing reference time, static time-value [Mani and Wilson, 2000; Wu et al., 2005; Wu et al., 2005] and static choosing-rules [Vozov, 2001; Jang et al., 2004; Lin et al., 2008], are not compatible with the real texts. The static time-value mechanism refers to taking the report time or publication time of the document as the fixed reference time for the whole text when normalizing. And the static choosing-rules mechanism means that the machine always uses fixed rules by contexts to choose reference time for each Implicit Time whatever its temporal semantics is. The rule based on the nearest narrative time [Lin et al., 2008] is the most typical and effective one, which uses the nearest narrative time in text above as the reference time all the while. But actually the context-free assumption or the rote operation is unsuitable for universal Implicit Times. For example, a news report is as Figure 1 shows:

(*Beijing, May 6, 2009*) B company took over A company totally on March 8, 2000. After one week, B company listed in Hong Kong, and became the first listed company in that industry. However, owing to the decision-making mistakes in the leadership and the company later poor management, B company got into debt for several hundred million dollars, and was forced to announce bankruptcy this Monday.

Figure 1. Example of news reports

For these two Implicit Times in the text, "after one week" and "this Monday", obviously there will be critical conflicts when using these two mechanisms referred above to choose reference time. The static time-value is unsuited for the "after one week", and "this Monday" makes mistakes when taking the nearest narrative time (i.e., "after one week") as the reference time to normalize according to the static choosing-rules.

Motivated by this issue, we propose a new reference time choosing mechanism for temporal expression normalization. Firstly, we segment the Implicit Time into two parts, modifier and temporal noun, and then train a classifier with referential features of these two parts to classify Implicit Times. As a result, we choose the corresponding reference time for each temporal expression depending on its class when normalizing. Meanwhile an acceptable defuzzification solution is introduced to normalize fuzzy times in our method. And the contributions of this paper are:

(1) We introduce a simple but effective reference time choosing method, called dynamic-choosing mechanism, which can choose the appropriate reference times automatically for universal Implicit Times as well as be compatible with the dynamically changeable contexts.

(2) Going beyond traditional normalization approaches, we develop a new way to deal with the defuzzification in order to figure out the *fuzzy reference time* (the reference time is vague or has imprecise start and end in timeline), which makes the normalization robust and improve the accuracy of reference times.

The rest of this paper is organized as follows. Section 2 discusses related works. In section 3 we describe the reference time dynamic-choosing mechanism. The temporal expression normalization is presented in section 4. Section 5 gives the description about experiments and evaluations. Finally, conclusion and future work are presented in section 6.

2 Related Work

In general, several research works on normalizing temporal expressions, which are involved in English [Mani and Wilson, 2000], French [Vozov, 2001], Spanish [Saquete et al., 2002], Korean [Jang et al., 2004] and Chinese [Wu et al., 2005; Lin et al., 2008], have been reported in recent years. Among them, the hand-crafted rules-based methods [Saquete et al., 2002; Schilder and Habel, 2001; Mani and Wilson, 2000] can deal with various temporal expressions, but the procedure to build a robust rules system is quite time-consuming. With regard to the machine learning for normalization [Jang et al., 2004; Wu et al., 2005; Vicente-Diez et al., 2008], the potential task is the classification which is deciding one explanation of a temporal expression from several alternatives.

However, these works on temporal expression normalization do not give an effective reference time choosing method for Implicit Times in real texts. More specifically, the pioneer work by Lacarides [1992] investigated various contextual effects on different temporal-reference relations. Then Hitzeman et al. [1995] discussed the refer-

ence-choosing taking into account the effects of tense, aspect, temporal adverbials and rhetorical relations. Dorr and Gaasterland [2002] presented the enhanced one in addition considering the connecting words. But they are theoretical in nature and heavily dependent on languages. Currently, the static time-value mechanism [Mani and Wilson, 2000; Wu et al., 2005; Wu et al., 2005] and the static choosing-rules mechanism [Vozov, 2001; Jang et al., 2004; Lin et al., 2008] for reference time choosing are applied into some systems widely. Nevertheless, as the discussion in section 1, these two ways are not adaptable to universal Implicit Times. In addition, Vicente-Diez et al. [2008; 2009] discussed the reference date for relative times, but the alternative rules are not effective in experiments. Lin et al. [2008] considered the condition that there is no report time or publication time when choosing reference time.

Referring to the defuzzification, TIMEX2 Standard [Ferro et al., 2005] takes the X placeholder to express fuzzy times' value, so the related works [Jang et al., 2004; Lin et al., 2008; Vicente-Diez and Martinez, 2009] almost follow this vague expressing way. However, this method can not address the actual situation that the fuzzy time is referred to by other times. Based on the human cognitive psychology, Anderson et al. [1983] presented a classical scenario-time shifting model that discussed the time includes the fuzzy time is the clue to scenario shifting when people reading. Inspired by this issue and based on our experiments, we find all times in a same scenario own strong dependences in temporal granularity, which can effectively help us determine granularity in defuzzification. And more details are discussed in section 4.2.

Aiming at solving these challenges above, we establish a temporal expression normalization system for real texts, which improves the accuracy of temporal reference normalization remarkably by the dynamic-choosing mechanism.

3 Reference Time Dynamic-choosing Mechanism

3.1 Referential feature in Implicit Time

In this paper, we define the Implicit Time consists of the modifier and the temporal noun which is modified by modifiers. And here we extend the modifier based on the TIMEX2 Standard, which include verb, conjunction, adverb and preposition that quantify or modify temporal nouns. For example, "ten days" is a temporal noun, but "ten days ago" is modified after adding the modifier "ago".

Meanwhile we find no matter how long or how many modifiers modify the temporal noun, the whole temporal expression holds the original temporal reference inferred from the temporal noun. Moreover, the key point of normalizing temporal expressions is choosing the appropriate reference time according to the real context rather than deciding the right direction or computing the measurable offset. For instance, with regard to these two Implicit Times in Figure 1, "after one week" and "this Monday", we can achieve the referential direction easily from the modifiers through some mapping rules. Meanwhile, the offsets are able to be understood directly by machine with pattern matching. But for the reference time, we must build the context-depending reference reasoning to trace it. The reference link is described as Figure 2 shows.

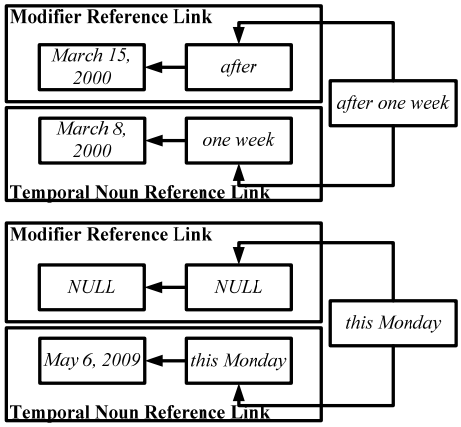


Figure 2. Example of reference link

From the reference reasoning, we can see the full temporal reference comes from two parts: modifier reference and temporal noun reference. Because the former is inferred from the latter, the temporal noun reference reasoning plays more important roles in normalization. In other words, the reference reasoning of the whole Implicit Time strongly depends on the temporal noun. Furthermore, in the practical operation, we indeed take the report time or the nearest narrative time in text above as the reference time of the temporal noun when normalizing a whole Implicit Time. Therefore, we consider the classi-

fication of the Implicit Time based on the classes of temporal noun's reference time. Basically we tag the Implicit Time as the same class as its temporal noun's under classifying temporal nouns into two classes according to the referential feature.

(1) *Global Temporal Noun*: Global Temporal Noun takes the report time or publication time of the document as the reference time when normalizing. Basically, it is independent of the local context.

(2) *Local Temporal Noun*: Local Temporal Noun makes reference to the nearest narrative time in text above in normalization due to depending on the local context.

Table 1 and 2 give some examples of Global Temporal Noun and Local Temporal Noun in real texts.

Consequently, here we denote the Implicit Time consists of the Global Temporal Noun and the modifier(s) by *Global Time* or GT, and accordingly the Local Temporal Noun corresponds to *Local Time* or LT.

Class	Sub-class	Examples
Global Temporal Noun	year	*last year*
	month	*next month*
	day	*this Friday*
	hour	*tonight*
	fuzzy	*lately*[2]

Table 1. Common Global Temporal Noun expressions

Class	Sub-class	Examples
Local Temporal Noun	year	*that year*
	month	*October*
	day	*the second day*
	hour	*morning*
	fuzzy	*then*
	duration	*one month*

Table 2. Common Local Temporal Noun expressions

3.2 Naïve Bayesian Classifier

A variety of machine learning classifiers are designed to resolve the classification problem, such as SVM classifier, ME classifier and the Decision Tree family. But the performance of these classifiers is greatly depending on the features selection. Based on the observation and analysis in our experiments, we find the referential feature holds in the temporal noun is hard to express with some explicit denotations. For example, "that year" and "this year" are nearly identical in surface feature, but the former is locally context-depending while the latter is locally context-free. So the Naïve Bayesian Classifier that assumes independence among feature denotations is suitable to be applied to our method.

We take the single word in the temporal noun as the object attribute x_i after removing the Explicit Time in the whole text. Given the class label c, the classifier learns the conditional probability of each attribute x_i from training data. Meanwhile, achieving the practical instance of X, classification is then performed by applying Bayes rules to compute the probability of c, and then predicting the class with the highest posterior probability.

$$c_o = \arg\max_c grade(c \mid x_1, x_2, \cdots, x_n) \quad x_i \in X \quad (1)$$

$$grade(c \mid x_1, x_2, \cdots, x_n) = \frac{p(c \mid x_1, x_2, \cdots, x_n)}{p(\bar{c} \mid x_1, x_2, \cdots, x_n)} \quad (2)$$

Applying Bayes rules to (2), we have:

$$grade(c \mid x_1, x_2, \cdots, x_n) = \frac{p(x_1, x_2, \cdots, x_n \mid c)p(c)}{p(x_1, x_2, \cdots, x_n \mid \bar{c})p(\bar{c})} = \frac{\prod_{i=1}^{n} p(x_i \mid c)p(c)}{\prod_{i=1}^{n} p(x_i \mid \bar{c})p(\bar{c})} \quad (3)$$

Actually, we estimate $p(x_i \mid c)$ and $p(x_i \mid \bar{c})$ by Maximum Likelihood Estimation (MLE) from training data with Dirichlet Smoothing method [Li et al., 2004].

$$p(x_i \mid c) = \frac{num(x_i, c) + \mu}{\sum_{j=1}^{n} num(x_j, c) + \mu \cdot n} \quad (4)$$

$$p(x_i \mid \bar{c}) = \frac{num(x_i, \bar{c}) + \mu}{\sum_{j=1}^{n} num(x_j, \bar{c}) + \mu \cdot n} \quad (5)$$

3.3 Reference Time Choosing

In our approach, there is a reference time table is used to hold full reference times for the whole text, and we need to update and maintain it dynamically after each normalizing processing.

[2] Some single temporal adverbs are taken as temporal noun, e.g. recently, currently and so on.

The time table consists of two parts: Global Reference Time and Local Reference Time.

(1) *Global Reference Time*: Global Reference Time (GRT) is a type of reference time which is referred to by the Global Time. Specifically, it is the report time or the publication time of the document.

(2) *Local Reference Time*: Local Reference Time (LRT) is made reference to by the Local Time. It will be updated dynamically after each normalizing.

Figure 3 shows a sample of the interaction between reference times and target times.

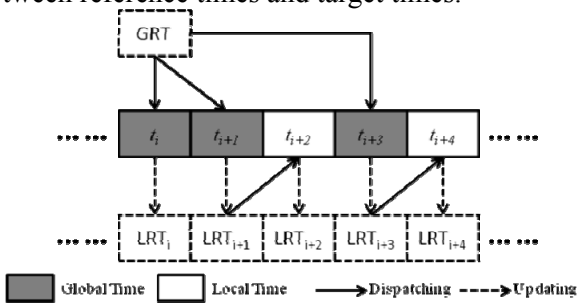


Figure 3. Interaction between reference times and target times

In Figure 3, we notice that different classes of time dynamically and automatically choose references based on their respective classes rather than do it using the fixed value or the inconsiderate rule under the static mechanism. And the reference time table is updated in real time finishing each normalizing, which makes the temporal situation compliable with dynamically changeable contexts.

4 Temporal Expression Normalization

4.1 Basic Normalizing Algorithm

In the beginning, we need to achieve the report time (RT) or the publication time (PT) of the document to initialize the GRT and LRT. Additionally, the fuzzy time can be referred to by other times in the normalization, but we must solve the defuzzification problem before taking it as the reference time. With respect to this issue, we will discuss it in the next section. Consequently, the practical normalizing algorithm is as follows.

```
Algorithm: TimeNormalize
Input: temporal expression t_i in text
Output: regular time list TList
Begin
  //initialize the GRT and LRT with RT or PT of this
  document
  GRT ← Initialize (RT|PT)
  LRT ← Initialize (RT|PT)
  for each t_i in text do
    //segment t_i into modifier and temporal noun
    t_i' ← SegmentTemporal (t_i)
    if IsExplicitTime (t_i) is true
      //update the time table with t_i
      LRT ← UpdateTime (t_i)
      //insert t_i into regular time list directly
      TList ← InsertList (t_i)
    else
      if IsLocalTime (t_i') is true
       //retrieve the latest LRT from time table and then
        normalize t_i'
        T_i ← RegularizeTemporal (t_i' , LRT)
      else
        //retrieve GRT from time table and then
         normalize t_i'
        T_i ← RegularizeTemporal (t_i' , GRT)
      LRT ← UpdateTime (T_i)
      end if
    TList ← InsertList (T_i)
    end if
  return TList
End Begin
```

4.2 Temporal Expression Defuzzification

In general, the defuzzification for fuzzy times faces two problems: deciding granularity and choosing offset. Here we introduce some knowledge on the human cognitive psychology and the empirical method to figure out these two issues respectively. Based on the scenario-time shifting model referred in related works, we get the conclusion that once the scenario is shifting, the time is shifting. More specifically, the time shifting is reflected in the temporal granularity between two different scenarios. So referring to writers, they will choose a few temporal expressions own the same granularity to render the coherent temporal dimensionality in one scenario in order to avoid generating improper scenario shifting for readers. Figure 4 describes the variation process of the temporal granularity between two different scenarios through scenario-time shifting.

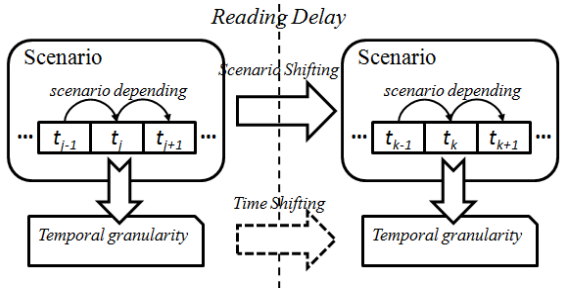


Figure 4. Variation process of temporal granularity

As conveyed in Figure 4, temporal expressions in the same scenario are constrained by the scenario depending. Hence fuzzy times should keep pace with scenario-correlative times in granularity. For example, two sentences in different scenarios:

"He was in Hong Kong *yesterday*, but now he is in Beijing."

"He was in Hong Kong *last year*, but now he is in Beijing."

Obviously, the first "now" means "today" in that scenario, and it has the same temporal granularity with "yesterday". Meanwhile it will be more appropriate for the second "now" choosing "year" as the temporal granularity than choosing "day" because of the dependence to the scenario-correlative time. In narrative, the paragraph is normally considered as the minimum unit of the scenario, so scenario-correlative relations should stand on the one paragraph at least.

But for the first temporal expression in the paragraph, we need to think about two specific conditions: when it appears in the first paragraph and in the non-first paragraph if it is a fuzzy time. Because there is no scenario shifting to the first paragraph, we employ a dictionary to initialize the algorithm when the first time included in the first paragraph is fuzzy time. The defuzzification process is outlined as follows.

Algorithm: *TempGranuarityDefuzzify*
Input: temporal expression t_i in text
Output: precise-granularity time t_i'
Begin
  //obtain the granularity of t_i
  granularity ← `GetGranularity` (t_i)
  //decide whether t_i is a fuzzy time
  if *granularity* is **not null**
    t_i' ← t_i
  else
    if t_i is **not** first temporal expression in current paragraph
      //assign the former's granularity to the t_i
      granularity ← `GetGranularity` (t_{i-1})
    else
      if t_i is **not** included in first paragraph
        //decide which granularity is assigned to t_i between t_{i-1} and t_{i+1}
        if `IsSameGranularity` (`GetGranularity` (t_{i-1}), `GetGranularity` (t_{i+1})) is **not true**
          granularity ← `GetGranularity` (`CoarserCompare` (t_{i-1}, t_{i+1}))
        else
          granularity ← `GetGranularity` (t_{i+1})
        end if
      else
        //retrieve default granularity from dictionary
        granularity ← `FindGranularityInDict` (t_i)
      end if
    end if
  //retrieve default offset from dictionary
  offset ← `FindOffsetInDict` (t_i)
  //update and intact all temporal attributes of t_i
  t_i' ← `ModifyTimeAttribute` (t_i, *granularity*, *offset*)
  end if
  return t_i'
End Begin

It's possible for the first temporal expression to correlate with forenamed times in last paragraph in real texts, so we choose the coarser granularity for the fuzzy time when appearing conflicts in granularity between the last temporal expression and the next temporal expression. Additionally, an empirical fuzzy time dictionary is constructed as the default in order to figure out the offset problem. For example, "lately" is denoted in dictionary as below.

Lately
Common synonyms: *recently*, *latterly*, *late*, *of late*
Default granularity: *day*
Default offset: *7 units*

Finishing the defuzzification for the whole text, the basic normalizing algorithm is evoked then. In the experiments, we find that the temporal expression defuzzified can clearly improve the accuracy of reference times besides discovering the implicit temporal information much more.

5 Evaluation

5.1 Setup

Because the normalization for temporal expressions is independent of the language [Wilson et al., 2001], we take the formal Chinese news[3] as

[3] People's Daily news corpus (January, 1998), supported by Institute of Computational Linguistics (ICL), Peking University.

the experimental corpus, which consist of 3148 Chinese news articles. The data collection contains 2,816,612 characters/967,884 words and 21,176 manually annotated temporal nouns. Among this corpus, 2518 articles (80%) include 13,835 temporal expressions are used as training data for the classification, and the rest (20%) as test data. Then the whole corpus is tested for the normalization. Event-anchored expressions are relevant with a specific event and it is hard to represent the exact meaning of them, so in our system, event-anchored expressions are not normalized.

5.2 Results

Results on Implicit Times classification: We firstly choose some temporal expressions classified in advance by crafted, and manually extend them in expressing patterns as the original training samples. For example, "last month" will extend to "this month" and "next month", which all belong to Global Times. Actually there are only 16,104 temporal expressions in our experiment because integrated temporal expressions in corpus are segmented into several parts, and we combine them together again before operating. Using classifier trained by training data, we get 2,264 Global Times and 998 Local Times from testing collections, where there are 1,705 Global Times and 804 Local Times are correct respectively by manual statistics. Table 3 gives the details of classification.

From the experiment data, we find the precision and the recall almost below 80%, and the classification performance is not expected. The reason is that we do not consider some special application situations beforehand, which result in classifying errors. For example, the Global Time should be taken as the Local Time when it appears in the dialog or speech that marks boundaries by a pair of quotation marks. So we introduce some revising patches shown in Table 4 to deal with this issue. Here the second and the fourth patches make corresponding temporal expressions be treated as non-target times that need not be processed. In addition, Time Set and Non-Specific are taken as the other classes except the Implicit Time and the Explicit Time. The final results with revising patches are shown in Table 5. Obviously, revising patches make the classification be more adapted for the real texts, and the performance evaluation is promising.

Class	#Correct	Precision (%)	Recall (%)	F-measure (%)
Global Time	1705	75.31	78.64	76.94
Local Time	804	80.56	79.45	80.00
Sum/Average	2509	77.94	79.05	78.47

Table 3. Results of classification

ID	Patch Type	Patterns	Operations
1	Dialog/Speech	"XXX"	Time → LT
2	Book/Movie	《XXX》 *XXX*	Be omitted
3	Time Set	quantifier + XXX	Time → others
4	Proper Noun	e.g. October Revolution	Be omitted
5	Non-Specific	e.g. childhood	Time → others

Table 4. Revising patches for classification

Class	#Correct	Precision (%)	Recall (%)	F-measure (%)
Global Time	1879	88.69	86.67	87.67
Local Time	918	90.55	90.71	90.63
Sum/Average	2797	89.62	88.69	89.15

Table 5. Results of classification with revising patches

Results on temporal expression normalization: For evaluating our algorithm objectively, we compare the experiment result with other two methods on the same testing corpus. The first compared method which is adopted in many traditional systems [Li et al., 2004; Wu et al., 2005] applies the static time-value mechanism to determine the reference time. The nearest narrative time [Lin et al., 2008; Vicente-Diez and Martinez, 2009] that represents the static choosing-rules mechanism is taken as the second compared method. Table 6 presents the results.

Method	Average referent updating/ article	Accuracy (%)	Errors	
			Referent (%)	Others (%)
STVM	0	68.42	22.84	8.74
SCRM	7.8	76.19	11.25	12.56
Our method	4.2	83.55	7.33	9.12

*STVM: Static Time-Value Mechanism
SCRM: Static Choosing-Rules Mechanism

Table 6. Results of normalization

The data shows that our method exceeds the compared ones evidently. The accuracy increases by 15.13% at most, and the errors by referent decreases by 3.92% at least. In contrast to the SCRM, we avoid the limitation that SCRM only concentrates on the nearest distance for choosing referent. Meanwhile, because the SCRM pays no attention to the normalization for fuzzy temporal expressions, the error by others (e.g. granularity) is greater than ours. Additionally, the STVM method applies the report time or the publication time of the document as the reference time for the whole text, so there is no referent updating in process. We mark all errors as referent errors as long as they involve with false reference time in results analysis, therefore, the STVM gets the highest referent errors ratio.

With respect to the defuzzification, we evaluate it on fuzzy times separately. All defuzzified fuzzy times are assessed by human, and then decided whether they are acceptable to the context. The evaluation results are shown in Table 7.

Type	#Acceptable	Acceptable ratio (%)	As referent (%)
Global Time	687	80.14	18.39
Local Time	159	92.61	6.43
Sum/Average	846	86.38	12.41

Table 7. Evaluations on temporal expression defuzzification

For the fuzzy temporal expression in Local Time, it is much fewer and easier than the one in Global Time in number and expression respectively, so the defuzzification in Local Times achieves more expected results. On the other hand, the fuzzy time in Global Time is often the first temporal expression in the first paragraph, and the corresponding dictionary-based method certainly affects the experiment results. According to the percentages that the temporal expressions defuzzified successfully account for in the all reference times, it demonstrates that the defuzzification makes contributions to the referential normalization besides discovering the internal temporal information in the fuzzy time.

6 Conclusion

In this paper, we present an approach to automatically normalizing temporal expressions under the reference time dynamic-choosing mechanism. The referential feature in temporal nouns is applied to classify Implicit Times. Based on this, different classes of times can be normalized according to their respective classes. Meanwhile, we introduce the scenario-time shifting model to deal with the defuzzification problem. The experiment shows that our approach achieves more promising evaluation results, and makes the automatic normalization more adaptable to real texts than the prior works. However, the neglect on the event-anchored expression certainly restricts the whole system in applications, so the event-anchored expression will be our research focus in future.

Acknowledgement This work is supported by the National Natural Science Foundation of China under the grant no. 60776801 and 70803001, the Open Projects Program of National Laboratory of Pattern Recognition (no.20090029), the Key Laboratory of Advanced Information Science and Network Technology of Beijing (no. xdxx1005), the National High Technology Research and Development Program ("863" Program) of China (Grant No. 2009AA12Z204).

References

Anderson, A., Garrod, S.C. and Sandford, A.J. 1983. The Accessibility of Pronominal Antecedents As a Function of Episode Shifts in Narrative Text. *Quarterly Journal of Experimental Psychology*, 35a, pp. 427-440.

Dorr, B. and Gaasterland, T. 2002. Constraints on the Generation of Tense, Aspect, and Connecting Words from Temporal Expressions. *Technical Report CS-TR-4391,UMIACS-TR-2002-71, LAMP-TR-091*, University of Maryland, College Park, MD, 2002.

Ferro, L., Gerber, L., Mani, I., et al. 2005. TIDES 2005 Standard for the Annotation of Temporal Expressions [EB/O L]. (2005-09) http: //timex2.mitre.org.

Hitzeman, J., Moens, M. and Grover, C. 1995. Algorithms for Analyzing the Temporal Structure of Discourse. In *Proceedings of the 7th European Meeting of the Association for Computational Linguistics*, pp. 253-260.

Jang, S.B., Baldwin, J. and Mani, I. 2004. Automatic TIMEX2 Tagging of Korean News. *ACM Transactions on Asian Language Information processing* 3(1), 51-65.

Lascarides, A., Asher, N. and Oberlander, J. 1992. Inferring Discourse Relations in Context. In *Proceedings of the 30th Meeting of the Association for Computational Linguistics*, pp. 1-8.

Li, W.J., Wong, K.F., Cao, G.H. et al. 2004. Applying Machine Learning to Chinese Temporal Relation Resolution. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, pp. 582-588.

Lin, J., Cao, D.F. and Yuan, C.F. 2008. Automatic TIMEX2 tagging of Chinese temporal information. *Journal of Tsinghua University* 48(1), 117-120.

Mani, I. and Wilson, G. 2000. Robust Temporal Processing of News. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pp. 69-76.

Saquete, E., Martinez-Barco, Patricio and Munoz, R. 2002. Recognizing and Tagging Temporal Expressions in Spanish, in *Proceedings of Workshop on Annotation Standards for Temporal Information in Natural Language*, pp. 44-51.

Schilder, F. and Habel, C. 2001. From Temporal Expressions to Temporal Information: Semantic Tagging of News Messages. In *Proceedings of the ACL 2001 Workshop on Temporal and Spatial Information Processing*, pp. 65-72.

Vazov, N. 2001. A System for Extraction of Temporal Expressions from French Texts based on Syntactic and Semantic Constraints. In *Proceedings of the ACL Workshop on Temporal and Spatial Information Processing*, pp. 96-103.

Vicente-Díez, M.T., Samy, D. and Martínez, P. 2008. An Empirical Approach to a Preliminary Successful identification and Resolution of Temporal Expressions in Spanish News Corpora. In *Proceedings of the Sixth International Language Resources and Evaluation*, pp. 2153-2158.

Vicente-Díez, M.T., Martinez, P. 2009. Temporal Semantics Extraction for Improving Web Search. In *Proceedings of the Workshop on Database and Expert Systems Application*, pp. 69-73.

Wilson, G., Mani, I., Sundheim, B. et al. 2001. A Multilingual Approach to Annotating and Extracting Temporal Information. In *Proceedings of the Workshop on Temporal and Spatial Information Processing*, pp. 1-7.

Wu, M.L., Li, W.J., Lu, Q. and Li, B.L. 2005. CTEMP: A Chinese temporal parser for extracting and normalizing temporal information. In *Proceedings of International Joint Conference on Natural Language Processing*, pp. 694-706.

Wu, M.L., Li, W.J., Chen, Q. and Lu, Q. 2005. Normalizing Chinese Temporal Expressions with Multi-label Classification. In *Proceedings of Natural Language Processing and Knowledge Engineering*, pp. 318-323.

Predicting Discourse Connectives for Implicit Discourse Relation Recognition

Zhi-Min Zhou and **Yu Xu**
East China Normal University
51091201052@ecnu.cn

Zheng-Yu Niu
Toshiba China R&D Center
zhengyu.niu@gmail.com

Man Lan and **Jian Su**
Institute for Infocomm Research
sujian@i2r.a-star.edu.sg

Chew Lim Tan
National University of Singapore
tancl@comp.nus.edu.sg

Abstract

Existing works indicate that the absence of explicit discourse connectives makes it difficult to recognize implicit discourse relations. In this paper we attempt to overcome this difficulty for implicit relation recognition by automatically inserting discourse connectives between arguments with the use of a language model. Then we propose two algorithms to leverage the information of these predicted connectives. One is to use these predicted implicit connectives as additional features in a supervised model. The other is to perform implicit relation recognition based only on these predicted connectives. Results on Penn Discourse Treebank 2.0 show that predicted discourse connectives help implicit relation recognition and the first algorithm can achieve an absolute average f-score improvement of 3% over a state of the art baseline system.

1 Introduction

Discourse relation analysis is to automatically identify discourse relations (e.g., explanation relation) that hold between arbitrary spans of text. This analysis may be a part of many natural language processing systems, e.g., text summarization system, question answering system. If there are discourse connectives between textual units to explicitly mark their relations, the recognition task on these texts is defined as *explicit discourse relation recognition*. Otherwise it is defined as *implicit discourse relation recognition*.

Previous study indicates that the presence of discourse connectives between textual units can greatly help relation recognition. In Penn Discourse Treebank (PDTB) corpus (Prasad et al., 2008), the most general senses, i.e., Comparison (Comp.), Contingency (Cont.), Temporal (Temp.) and Expansion (Exp.), can be disambiguated in explicit relations with more than 90% f-scores based only on the discourse connectives explicitly used to signal the relation (Pitler and Nenkova., 2009b).

However, for implicit relations, there are no connectives to explicitly mark the relations, which makes the recognition task quite difficult. Some of existing works attempt to perform relation recognition without hand-annotated corpora (Marcu and Echihabi, 2002), (Sporleder and Lascarides, 2008) and (Blair-Goldensohn, 2007). They use unambiguous patterns such as [Arg1, *but* Arg2] to create synthetic examples of implicit relations and then use [Arg1, Arg2] as an training example of an implicit relation. Another research line is to exploit various linguistically informed features under the framework of supervised models, (Pitler et al., 2009a) and (Lin et al., 2009), e.g., polarity features, semantic classes, tense, production rules of parse trees of arguments, etc.

Our study on PDTB test data shows that the average f-score for the most general 4 senses can reach 91.8% when we simply mapped the ground truth implicit connective of each test instance to its most frequent sense. It indicates the importance of connective information for implicit relation recognition. However, so far there is no previous study attempting to use such kind of connective information for implicit relation. One possi-

Coling 2008: Poster Volume, pages 1507–1514,
Beijing, August 2010

ble reason is that implicit connectives do not exist in unannotated real texts. Another evidence of the importance of connectives for implicit relations is shown in PDTB annotation. The PDTB annotation consists of inserting a connective expression that best conveys the inferred relation by the readers. Connectives inserted in this way to express inferred relations are called *implicit connectives*, which do not exist in real texts. These evidences inspire us to consider two interesting research questions:
(1) Can we automatically predict implicit connectives between arguments?
(2) How to use the predicted implicit connectives to build an automatic discourse relation analysis system?

In this paper we address these two questions as follows: (1) We insert appropriate discourse connectives between two textual units with the use of a language model. Here we train the language model on large amount of raw corpora without the use of any hand-annotated data. (2) Then we present two algorithms to use these predicted connectives for implicit relation recognition. One is to use these connectives as additional features in a supervised model. The other is to perform relation recognition based only on these connectives.

We performed evaluation of the two algorithms and a baseline system on PDTB 2.0 corpus. Experimental results showed that using predicted discourse connectives as additional features can significantly improve the performance of implicit discourse relation recognition. Specifically, the first algorithm achieved an absolute average f-score improvement of 3% over a state of the art baseline system.

The rest of this paper is organized as follows. Section 2 describes the two algorithms for implicit discourse relation recognition. Section 3 presents experiments and results on PDTB data. Section 4 reviews related work. Section 5 concludes this work.

2 Our Algorithms for Implicit Discourse Relation Recognition

2.1 Prediction of implicit connectives

Explicit discourse relations are easily identifiable due to the presence of discourse connectives between arguments. (Pitler and Nenkova., 2009b) showed that in PDTB corpus, the most general senses, i.e., Comparison (Comp.), Contingency (Cont.), Temporal (Temp.) and Expansion (Exp.), can be disambiguated in explicit relations with more than 90% f-scores based only on discourse connectives.

But for implicit relations, there are no connectives to explicitly mark the relations, which makes the recognition task quite difficult. PDTB data provides *implicit connectives* that are inserted between paragraph-internal adjacent sentence pairs not marked by any of explicit connectives. The availability of ground-truth implicit connectives makes it possible to evaluate the contribution of these connectives for implicit relation recognition. Our initial study on PDTB data show that the average f-score for the most general 4 senses can reach 91.8% when we obtained the sense of each test example by mapping each ground truth implicit connective to its most frequent sense. We see that connective information is an important knowledge source for implicit relation recognition. However these implicit connectives do not exist in real texts. In this paper we overcome this difficulty by inserting a connective between two arguments with the use of a language model.

Following the annotation scheme of PDTB, we assume that each implicit connective takes two arguments, denoted as *Arg1* and *Arg2*. Typically, there are two possible positions for most of implicit connectives[1], i.e., the position before Arg1 and the position between Arg1 and Arg2. Given a set of possible implicit connectives $\{c_i\}$, we generate two synthetic sentences, c_i+Arg1+Arg2 and Arg1+c_i+Arg2 for each c_i, denoted as $S_{c_i,1}$ and $S_{c_i,2}$. Then we calculate the perplexity (an intrinsic score) of these sentences with the use of a language model, denoted as $PPL(S_{c_i,j})$. According

[1]For parallel connectives, e.g., *if*...*then*..., the two connectives will take the two arguments together, so there is only one possible combination for connectives and arguments.

to the value of $PPL(S_{c_i,j})$ (the lower the better), we can rank these sentences and select the connectives in top N sentences as implicit connectives for this argument pair. The language model may be trained on large amount of unannotated corpora that can be cheaply acquired, e.g., North American News corpus.

2.2 Using predicted implicit connectives as additional features

We predict implicit connectives on both training set and test set. Then we can use the predicted implicit connectives as additional features for supervised implicit relation recognition. Previous works exploited various linguistically informed features under the framework of supervised models. In this paper, we include 9 types of features in our system due to their superior performance in previous studies, e.g., polarity features, semantic classes of verbs, contextual sense, modality, inquirer tags of words, first-last words of arguments, cross-argument word pairs, ever used in (Pitler et al., 2009a), production rules of parse trees of arguments used in (Lin et al., 2009), and intra-argument word pairs inspired by the work of (Saito et al., 2006).

Here we provide the details of the 9 features, shown as follows:

Verbs: Similar to the work in (Pitler et al., 2009a), the verb features consist of the number of pairs of verbs in Arg1 and Arg2 if they are from the same class based on their highest Levin verb class level (Dorr, 2001). In addition, the average length of verb phrase and the part of speech tags of main verb are also included as verb features.

Context: If the immediately preceding (or following) relation is an explicit, its relation and sense are used as features. Moreover, we use another feature to indicate if Arg1 leads a paragraph.

Polarity: We use the number of positive, negated positive, negative and neutral words in arguments and their cross product as features. For negated positives, we locate the negated words in text span and then define the closely behind positive word as negated positive.

Modality: We look for modal words including their various tenses or abbreviation forms in both arguments. Then we generate a feature to indicate the presence or absence of modal words in both arguments and their cross product.

Inquirer Tags: Inquirer Tags extracted from General Inquirer lexicon (Stone et al., 1966) contains positive or negative classification of words. In fact, its fine-grained categories, such as Fall versus Rise, or Pleasure versus Pain, can indicate the relation between two words, especially for verbs. So we choose the presence or absence of 21 pair categories with complementary relation in Inquirer Tags as features. We also include their cross production as features.

FirstLastFirst3: We choose the first and last words of each argument as features, as well as the pair of first words, the pair of last words, and the first 3 words in each argument. In addition, we apply Porter's Stemmer (Porter, 1980) to each word before preparation of these features.

Production Rule: According to (Lin et al., 2009), we extract all the possible production rules from arguments, and check whether the rules appear in Arg1, Arg2 and both arguments. We remove the rules occurring less than 5 times in training data.

Cross-argument Word Pairs: We perform the Porter's stemming (Porter, 1980), and then group all words from Arg1 and Arg2 into two sets W_1 and W_2 respectively. Then we generate any possible word pair (w_i, w_j) ($w_i \in W_1$, $w_j \in W_2$). We remove the word pairs with less than 5 times.

Intra-argument Word Pairs: Let $Q_1 = (q_1, q_2, \ldots, q_n)$ be the word sequence of Arg1. The intra-argument word pairs for Arg1 is defined as $WP_1 = ((q_1, q_2), (q_1, q_3), \ldots, (q_1, q_n), (q_2, q_3), \ldots, (q_{n-1}, q_n))$. We extract all the intra-argument word pairs from Arg1 and Arg2 and remove word pairs appearing less than 5 times in training data.

2.3 Relation recognition based only on predicted implicit connectives

After the prediction of implicit connectives, we can address the implicit relation recognition task with the methods for explicit relation recognition due to the presence of implicit connectives, e.g., sense classification based only on connectives (Pitler and Nenkova., 2009b). The work of (Pitler and Nenkova., 2009b) showed that most

of connectives are unambiguous and it is possible to obtain high performance in prediction of discourse sense due to the simple mapping relation between connectives and senses. Given two examples:
(E1) She paid less on her dress, ***but*** it is very nice.
(E2) We have to harry up ***because*** the raining is getting heavier and heavier.
The two connectives, i.e., *but* in E1 and *because* in E2, convey Comparison and Contingency sense respectively. In most cases, we can easily recognize the relation sense by the appearance of discourse connective since it can be interpreted in only one way. That means, the ambiguity of the mapping between sense and connective is quite few.

We count the frequency of sense tags for each possible connective on PDTB training data for implicit relation. Then we build a sense recognition model by simply mapping each connective to its most frequent sense. Here we do not perform connective prediction on training data. During testing, we use the language model to insert implicit connectives into each test argument pair. Then we perform relation recognition by mapping each implicit connective to its most frequent sense.

3 Experiments and Results

3.1 Experiments

3.1.1 Data sets

In this work we used the PDTB 2.0 corpus for evaluation of our algorithms. Following the work of (Pitler et al., 2009a), we used sections 2-20 as training set, sections 21-22 as test set, and sections 0-1 as development set for parameter optimization. For comparison with the work of (Pitler et al., 2009a), we ran four binary classification tasks to identify each of the main relations (Cont., Comp., Exp., and Temp.) from the rest. For each relation, we used equal numbers of positive and negative examples as training data[2]. The negative examples were chosen at random from sections 2-20. We used all the instances in sections 21 and 22 as test set, so the test set is representative of the natural distribution. The numbers of positive and negative instances for each sense in different data sets are listed in Table 1.

Table 1: Statistics of positive and negative samples in training, development and test sets for each relation.

Relation	Train	Dev	Test
	Pos/Neg	Pos/Neg	Pos/Neg
Comp.	1927/1927	191/997	146/912
Cont.	3375/3375	292/896	276/782
Exp.	6052/6052	651/537	556/502
Temp.	730/730	54/1134	67/991

In this work we used LibSVM toolkit to construct four linear SVM models for a baseline system and the system in Section 2.2.

3.1.2 A baseline system

We first built a baseline system, which used 9 types of features listed in Section 2.2.

We tuned the numbers of firstLastFirst3, cross-argument word pair, intra-argument word pair on development set. Finally we set the frequency threshold at 3, 5 and 5 respectively.

3.1.3 Prediction of implicit connectives

To predict implicit connectives, we adopt the following two steps:(1) train a language model; (2) select top N implicit connectives.

Step 1: We used SRILM toolkit to train the language models on three benchmark news corpora, i.e., New York part in the BLLIP North American News, Xin and Ltw parts of English Gigaword (4th Edition). We also tried different values for n in n-gram model. The parameters were tuned on the development set to optimize the accuracy of prediction. In this work we chose 3-gram language model trained on NY corpus.

Step 2: We combined each instance's Arg1 and Arg2 with connectives extract from PDTB2 (100 in all). There are two types of connectives, single connective (e.g. *because* and *but*) and parallel connective (such as "*not only* ..., *but also*"). Since discourse connectives may appear not only ahead of the Arg1, but also between Arg1 and Arg2, we considered this case. Given a set of possible implicit connectives $\{c_i\}$, for single connective $\{c_i\}$, we constructed two synthetic sentences, c_i+Arg1+Arg2 and Arg1+c_i+Arg2. In case of

[2]Here the numbers of training and test instances for Expansion relation are different from those in (Pitler et al., 2009a). The reason is that we do not include instances of EntRel as positive examples.

parallel connective, we constructed one synthetic sentence like c_{i1}+Arg1+c_{i2}+Arg2.

As a result, we can get 198 synthetic sentences for each argument pair. Then we converted all words to lower cases and used the language model trained in the above step to calculate perplexity on sentence level. The perplexity scores were ranked from low to high. For example, we got the perplexity (*ppl*) for two sentences as follows:
(1) *but* this is an old story, we're talking about years ago before anyone heard of asbestos having any questionable properties.
ppl= 652.837
(2) this is an old story, *but* we're talking about years ago before anyone heard of asbestos having any questionable properties.
ppl= 583.514

We considered the combination of connectives and their position as final features like mid_*but*, first_*but*, where the features are binary, that is, the presence and absence of the specific connective.

According to the value of $PPL(S_{c_i,j})$ (the lower the better), we selected the connectives in top N sentences as implicit connectives for this argument pair. In order to get the optimal N value, we tried various values of N on development set and selected the minimum value of N so that the ground-truth connectives appeared in top N connectives. The final N value is set to 60 based on the trade-off between performance and efficiency.

3.1.4 Using predicted connectives as additional features

This system combines the predicted implicit connectives as additional features and the 9 types of features in an supervised framework. The 9 types of features are listed as shown in Section 2.2 and tuned on development set.

We combined predicted connectives with the best subset features from the development data set with respect to f-score. In our experiment of selecting best subset features, single features rather than the combination of several features achieved much higher scores. So we combine single features with predicted connectives as final features.

3.1.5 Using only predicted connectives for implicit relation recognition

We built two variants for the algorithm in Section 2.3. One is to use the data for explicit relations in PDTB sections 2-20 as training data. The other is to use the data for implicit relations in PDTB sections 2-20 as training data. Given training data, we obtained the most frequent sense for each connective appearing in the training data. Then given test data, we recognized the sense of each argument pair by mapping each predicted connective to its most frequent sense. In this work we conducted another experiment to see the upper-bound performance of this algorithm. Here we performed recognition based on ground-truth implicit connectives and used the data for implicit relations as training data.

3.2 Results

3.2.1 Result of baseline system

Table 2 summarizes the best performance achieved by the baseline system in comparison with previous state-of-the-art performance achieved in (Pitler et al., 2009a). The first two lines in the table show their best results using single feature and using combined feature subset. It indicates that the performance of using combined feature subset is higher than that using single feature alone.

From this table, we can find that our baseline system has a comparable result on Contingency and Temporal. On Comparison, our system achieved a better performance around 9% f-score higher than their best result. However, for Expansion, they expanded both training and testing sets by including EntRel relation as positive examples, which makes it impossible to perform direct comparison. Generally, our baseline system is reasonable and thus the consequent experiments on it are reliable.

3.2.2 Result of algorithm 1: using predicted connectives as additional features

Table 3 summarizes the best performance achieved by the baseline system and the first algorithm (i.e., baseline + Language Model) on test set. The second and third column show the best performance achieved by the baseline system and

Table 2: Performance comparison of the baseline system with the system of (Pitler et al., 2009a) on test set.

System	Comp. vs. Not F_1 (Acc)	Cont. vs. Other F_1 (Acc)	Exp. vs. Other F_1 (Acc)	Temp. vs. Other F_1 (Acc)
Using the best single feature (Pitler et al., 2009a)	21.01(52.59)	36.75(62.44)	71.29(59.23)	15.93(61.20)
Using the best feature subset (Pitler et al., 2009a)	21.96(56.59)	47.13(67.30)	76.42(63.62)	16.76(63.49)
The baseline system	30.72(78.26)	45.38(40.17)	65.95(57.94)	16.46(29.96)

the first algorithm using predicted connectives as additional features.

Table 3: Performance comparison of the algorithm in Section 2.2 with the baseline system on test set.

Relation	Features	Baseline F_1 (Acc)	Baseline+LM F_1 (Acc)
Comp.	Production Rule	**30.72**(78.26)	31.08(68.15)
	Context	24.66(42.25)	27.64(53.97)
	InquirerTags	23.31(73.25)	27.87(55.48)
	Polarity	21.11(40.64)	23.64(52.36)
	Modality	17.25(80.06)	26.17(55.20)
	Verbs	25.00(53.50)	**31.79**(58.22)
Cont.	Prodcution Rule	**45.38**(40.17)	**47.16**(48.96)
	Context	37.61(44.70)	34.74(48.87)
	Polarity	35.57(50.00)	43.33(33.74)
	InquirerTags	38.04(41.49)	42.22(36.11)
	Modality	32.18(66.54)	35.26(55.58)
	Verbs	40.44(54.06)	42.04(32.23)
Exp.	Context	48.34(54.54)	68.32(53.02)
	FirstLastFirst3	**65.95**(57.94)	68.94(53.59)
	InquirerTags	61.29(52.84)	68.49(53.21)
	Modality	64.36(56.14)	68.9(52.55)
	Polarity	49.95(50.38)	68.62(53.40)
	Verbs	52.95(53.31)	**70.11**(54.54)
Temp.	Context	13.52(64.93)	16.99(79.68)
	FirstLastFirst3	15.75(66.64)	19.70(64.56)
	InquirerTags	8.51(83.74)	19.20(56.24)
	Modality	**16.46**(29.96)	19.97(54.54)
	Polarity	16.29(51.42)	**20.30**(55.48)
	Verbs	13.88(54.25)	13.53(61.34)

From this table, we found that this additional feature obtained from language model showed significant improvements in almost four relations. Specifically, the top two improvements are on Expansion and Temporal relations, which improved 4.16% and 3.84% in f-score respectively. Although on Comparison relation there is only a slight improvement (+1.07%), our two best systems both got around 10% improvements of f-score over a state-of-the-art system in (Pitler et al., 2009a). As a whole, the first algorithm achieved 3% improvement of f-score over a state of the art baseline system. All these results indicate that predicted implicit connectives can help improve the performance.

3.2.3 Result of algorithm 2: using only predicted connectives for implicit relation recognition

Table 4 summarizes the best performance achieved by the second algorithm in comparison with the baseline system on test set.

The experiment showed that the baseline system using just gold-truth implicit connectives can achieve an f-score of 91.8% for implicit relation recognition. It once again proved that implicit connectives make significant contributions for implicit relation recognition. This also encourages our future work on finding the most suitable connectives for implicit relation recognition.

From this table, we found that, using only predicted implicit connectives achieved an comparable performance to (Pitler et al., 2009a), although it was still a bit lower than our best baseline. But we should bear in mind that this algorithm only uses 4 features for implicit relation recognition and these 4 features are easy computable and fast run, which makes the system more practical in application. Furthermore, compared with other algorithms which require hand-annotated data for training, the performance of this second algorithm is acceptable if we take into account that no labeled data is used for model training.

3.3 Analysis

Experimental results on PDTB showed that using the predicted implicit connectives significantly improves the performance of implicit discourse relation recognition. Our first algorithm achieves an average f-score improvement of 3% over a state of the art baseline system. Specifically, for the relations: Comp., Cont., Exp., Temp., our first algorithm can achieve 1.07%, 1.78%, 4.16%, 3.84% f-score improvements over a state of the art baseline system. Since (Pitler et al., 2009a)

Table 4: Performance comparison of the algorithm in Section 2.3 with the baseline system on test set.

System	Comp. vs. Other F_1 (Acc)	Cont. vs. Other F_1 (Acc)	Exp. vs. Other F_1 (Acc)	Temp. vs. Other F_1 (Acc)
The baseline system	30.72(78.26)	45.38(40.17)	65.95(57.94)	16.46(29.96)
Our algorithm with training data for explicit relation	26.02(52.17)	35.72(51.70)	64.94(53.97)	13.76(41.97)
Our algorithm with training data for implicit relation	24.55(63.99)	16.26(70.79)	60.70(53.50)	14.75(70.51)
Sense recognition using gold-truth implicit connectives	94.08(98.30)	98.19(99.05)	97.79(97.64)	77.04(97.07)

used different selection of instances for Expansion sense[3], we cannot make a direct comparison. However, we achieve the best f-score around 70%, which provide 5% improvements over our baseline system. On the other hand, the second proposed algorithm using only predicted connectives still achieves promising results for each relation. Specifically, the model for the Comparison relation achieves an f-score of 26.02% (5% over the previous work in (Pitler et al., 2009a)). Furthermore, the models for Contingency and Temporal relation achieve 35.72% and 13.76% f-score respectively, which are comparable to the previous work in (Pitler et al., 2009a). The model for Expansion relation obtains an f-score of 64.95%, which is only 1% less than our baseline system which consists of ten thousands of features.

4 Related Work

Existing works on automatic recognition of discourse relations can be grouped into two categories according to whether they used hand-annotated corpora.

One research line is to perform relation recognition without hand-annotated corpora.

(Marcu and Echihabi, 2002) used a pattern-based approach to extract instances of discourse relations such as Contrast and Elaboration from unlabeled corpora. Then they used word-pairs between two arguments as features for building classification models and tested their model on artificial data for implicit relations.

There are other efforts that attempt to extend the work of (Marcu and Echihabi, 2002). (Saito et al., 2006) followed the method of (Marcu and Echihabi, 2002) and conducted experiments with combination of cross-argument word pairs and phrasal patterns as features to recognize implicit relations between adjacent sentences in a Japanese corpus. They showed that phrasal patterns extracted from a text span pair provide useful evidence in the relation classification. (Sporleder and Lascarides, 2008) discovered that Marcu and Echihabi's models do not perform as well on implicit relations as one might expect from the test accuracies on synthetic data. (Blair-Goldensohn, 2007) extended the work of (Marcu and Echihabi, 2002) by refining the training and classification process using parameter optimization, topic segmentation and syntactic parsing.

(Lapata and Lascarides, 2004) dealt with temporal links between main and subordinate clauses by inferring the temporal markers linking them. They extracted clause pairs with explicit temporal markers from BLLIP corpus as training data.

Another research line is to use human-annotated corpora as training data, e.g., the RST Bank (Carlson et al., 2001) used by (Soricut and Marcu, 2003), adhoc annotations used by (**?**), (Baldridge and Lascarides, 2005), and the Graph-Bank (Wolf et al., 2005) used by (Wellner et al., 2006).

Recently the release of the Penn Discourse TreeBank (PDTB) (Prasad et al., 2008) benefits the researchers with a large discourse annotated corpora, using a comprehensive scheme for both implicit and explicit relations. (Pitler et al., 2009a) performed implicit relation classification on the second version of the PDTB. They used several linguistically informed features, such as word polarity, verb classes, and word pairs, showing performance increases over a random classification baseline. (Lin et al., 2009) presented an implicit discourse relation classifier in PDTB with the use of contextual relations, constituent Parse Features, dependency parse features and cross-argument word pairs.

[3]They expanded the Expansion data set by adding randomly selected EntRel instances by 50%, which is considered to significantly change data distribution.

In comparison with existing works, we investigated a new knowledge source, implicit connectives, for implicit relation recognition. Moreover, our two models can exploit both labeled and unlabeled data by training a language model on unlabeled data and then using this language model to generate implicit connectives for recognition models trained on labeled data.

5 Conclusions

In this paper we use a language model to automatically generate implicit connectives and then present two methods to use these connectives for recognition of implicit relations. One method is to use these predicted implicit connectives as additional features in a supervised model and the other is to perform implicit relation recognition based only on these predicted connectives. Results on Penn Discourse Treebank 2.0 show that predicted discourse connectives help implicit relation recognition and the first algorithm achieves an absolute average f-score improvement of 3% over a state of the art baseline system.

Acknowledgments

This work is supported by grants from National Natural Science Foundation of China (No.60903093), Shanghai Pujiang Talent Program (No.09PJ1404500) and Doctoral Fund of Ministry of Education of China (No.20090076120029).

References

J. Baldridge and A. Lascarides. 2005. *Probabilistic head-driven parsing for discourse structure*. Proceedings of the Ninth Conference on Computational Natural Language Learning.

L. Carlson, D. Marcu, and Ma. E. Okurowski. 2001. *Building a discourse-tagged corpus in the framework of rhetorical structure theory*. Proceedings of the Second SIG dial Workshop on Discourse and Dialogue.

B. Dorr. LCS Verb Database. *Technical Report Online Software Database, University of Maryland, College Park, MD*,2001.

R. Girju. 2003. *Automatic detection of causal relations for question answering*. In ACL 2003 Workshops.

S. Blair-Goldensohn. 2007. *Long-Answer Question Answering and Rhetorical-Semantic Relations*. Ph.D. thesis, Columbia Unviersity.

M. Lapata and A. Lascarides. 2004. *Inferring Sentence-internal Temporal Relations*. Proceedings of the North American Chapter of the Assocation of Computational Linguistics.

Z.H. Lin, M.Y. Kan and H.T. Ng. 2009. *Recognizing Implicit Discourse Relations in the Penn Discourse Treebank*. Proceedings of the 2009 Conference on EMNLP.

D. Marcu and A. Echihabi. 2002. *An Unsupervised Approach to Recognizing Discourse Relations*. Proceedings of the 40th ACL.

E. Pitler, A. Louis, A. Nenkova. 2009. *Automatic sense prediction for implicit discourse relations in text*. Proceedings of the 47th ACL.

E. Pitler and A. Nenkova. 2009. *Using Syntax to Disambiguate Explicit Discourse Connectives in Text*. Proceedings of the ACL-IJCNLP 2009 Conference Short Papers.

M. Porter. 1980. An algorithm for suffix stripping. In *Program*, vol. 14, no. 3, pp.130-137.

R. Prasad, N. Dinesh, A. Lee, E. Miltsakaki, L. Robaldo, A. Joshi, B. Webber. 2008. *The Penn Discourse TreeBank 2.0*. Proceedings of LREC'08.

M. Saito, K.Yamamoto, S.Sekine. 2006. *Using Phrasal Patterns to Identify Discourse Relations*. Proceeding of the HLTCNA Chapter of the ACL.

R. Soricut and D. Marcu. *Sentence Level Discourse Parsing using Syntactic and Lexical Information*. Proceedings of HLT/NAACL 2003.

C. Sporleder and A. Lascarides. 2008. *Using automatically labelled examples to classify rhetorical relations: an assessment*. Natural Language Engineering, Volume 14, Issue 03.

P.J. Stone, J. Kirsh, and Cambridge Computer Associates. 1966. *The General Inquirer: A Computer Approach to Content Analysis*. MIT Press.

B. Wellner , J. Pustejovsky, C. H. R. S., A. Rumshisky. 2006. *Classification of discourse coherence relations: An exploratory study using multiple knowledge sources*. Proceedings of the 7th SIGDIAL Workshop on Discourse and Dialogue.

F. Wolf, E. Gibson, A. Fisher, M. Knight. 2005. *The Discourse GraphBank: A database of texts annotated with coherence relations*. Linguistic Data Consortium.

Active Deep Networks for Semi-Supervised Sentiment Classification

Shusen Zhou, Qingcai Chen and Xiaolong Wang
Shenzhen Graduate School, Harbin Institute of Technology
zhoushusen@hitsz.edu.cn, qincai.chen@hitsz.edu.cn,
wangxl@insun.hit.edu.cn

Abstract

This paper presents a novel semi-supervised learning algorithm called Active Deep Networks (ADN), to address the semi-supervised sentiment classification problem with active learning. First, we propose the semi-supervised learning method of ADN. ADN is constructed by Restricted Boltzmann Machines (RBM) with unsupervised learning using labeled data and abundant of unlabeled data. Then the constructed structure is fine-tuned by gradient-descent based supervised learning with an exponential loss function. Second, we apply active learning in the semi-supervised learning framework to identify reviews that should be labeled as training data. Then ADN architecture is trained by the selected labeled data and all unlabeled data. Experiments on five sentiment classification datasets show that ADN outperforms the semi-supervised learning algorithm and deep learning techniques applied for sentiment classification.

1 Introduction

In recent years, sentiment analysis has received considerable attentions in Natural Language Processing (NLP) community (Blitzer et al., 2007; Dasgupta and Ng, 2009; Pang et al., 2002). Polarity classification, which determine whether the sentiment expressed in a document is positive or negative, is one of the most popular tasks of sentiment analysis (Dasgupta and Ng, 2009). Sentiment classification is a special type of text categorization, where the criterion of classification is the attitude expressed in the text, rather than the subject or topic. Labeling the reviews with their sentiment would provide succinct summaries to readers, which makes it possible to focus the text mining on areas in need of improvement or on areas of success (Gamon, 2004) and is helpful in business intelligence applications, recommender systems, and message filtering (Pang, et al., 2002).

While topics are often identifiable by keywords alone, sentiment classification appears to be a more challenge task (Pang, et al., 2002). First, sentiment is often conveyed with subtle linguistic mechanisms such as the use of sarcasm and highly domain-specific contextual cues (Li et al., 2009). For example, although the sentence "The thief tries to protect his excellent reputation" contains the word "excellent", it tells us nothing about the author's opinion and in fact could be well embedded in a negative review. Second, sentiment classification systems are typically domain-specific, which makes the expensive process of annotating a large amount of data for each domain and is a bottleneck in building high quality systems (Dasgupta and Ng, 2009). This motivates the task of learning robust sentiment models from minimal supervision (Li, et al., 2009).

Recently, semi-supervised learning, which uses large amount of unlabeled data together with labeled data to build better learners (Raina et al., 2007; Zhu, 2007), has drawn more attention in sentiment analysis (Dasgupta and Ng, 2009; Li, et al., 2009). As argued by several researchers (Bengio, 2007; Salakhutdinov and Hinton, 2007), deep architecture, composed of multiple levels of non-linear operations (Hinton et al., 2006), is expected to perform well in semi-supervised learning because of its capability of modeling hard artificial intelligent tasks. Deep Belief Networks (DBN) is a representative

Coling 2008: Poster Volume, pages 1515–1523,
Beijing, August 2010

deep learning algorithm achieving notable success for semi-supervised learning (Hinton, et al., 2006). Ranzato and Szummer (2008) propose an algorithm to learn text document representations based on semi-supervised auto-encoders that are combined to form a deep network.

Active learning is another way that can minimize the number of required labeled data while getting competitive result. Usually, the training set is chosen randomly. However, active learning choose the training data actively, which reduce the needs of labeled data (Tong and Koller, 2002). Recently, active learning had been applied in sentiment classification (Dasgupta and Ng, 2009).

Inspired by the study of semi-supervised learning, active learning and deep architecture, this paper proposes a novel semi-supervised polarity classification algorithm called Active Deep Networks (ADN) that is based on a representative deep learning algorithm Deep Belief Networks (DBN) (Hinton, et al., 2006) and active learning (Tong and Koller, 2002). First, we propose the ADN architecture, which utilizes a new deep architecture for classification, and an exponential loss function aiming to maximize the separability of the classifier. Second, we propose the ADN algorithm. It firstly identifies a small number of manually labeled reviews by an active learner, and then trains the ADN classifier with the identified labeled data and all of the unlabeled data.

Our paper makes several important contributions: First, this paper proposes a novel ADN architecture that integrates the abstraction ability of deep belief nets and the classification ability of backpropagation strategy. It improves the generalization capability by using abundant unlabeled data, and directly optimizes the classification results in training dataset using back propagation strategy, which makes it possible to achieve attractive classification performance with few labeled data. Second, this paper proposes an effective active learning method that integrates the labeled data selection ability of active learning and classification ability of ADN architecture. Moreover, the active learning is also based on the ADN architecture, so the labeled data selector and the classifier are based on the same architecture, which provides an unified framework for semi-supervised classification task. Third, this paper applies semi-supervised learning and active learning to sentiment classification successfully and gets competitive performance. Our experimental results on five sentiment classification datasets show that ADN outperforms previous sentiment classification methods and deep learning methods.

The rest of the paper is organized as follows. Section 2 gives an overview of sentiment classification. The proposed semi-supervised learning method ADN is described in Section 3. Section 4 shows the empirical validation of ADN by comparing its classification performance with previous sentiment classifiers and deep learning methods on sentiment datasets. The paper is closed with conclusion.

2 Sentiment Classification

Sentiment classification can be performed on words, sentences or documents, and is generally categorized into lexicon-based and corpus-based classification method (Wan, 2009). The detailed survey about techniques and approaches of sentiment classification can be seen in the book (Pang and Lee, 2008). In this paper we focus on corpus-based classification method.

Corpus-based methods use a labeled corpus to train a sentiment classifier (Wan, 2009). Pang et al. (2002) apply machine learning approach to corpus-based sentiment classification firstly. They found that standard machine learning techniques outperform human-produced baselines. Pang and Lee (2004) apply text-categorization techniques to the subjective portions of the sentiment document. These portions are extracted by efficient techniques for finding minimum cuts in graphs. Gamon (2004) demonstrate that using large feature vectors in combination with feature reduction, high accuracy can be achieved in the very noisy domain of customer feedback data. Xia et al. (2008) propose the sentiment vector space model to represent song lyric document, assign the sentiment labels such as light-hearted and heavy-hearted.

Supervised sentiment classification systems are domain-specific and annotating a large scale corpus for each domain is very expensive (Dasgupta and Ng, 2009). There are several solutions for this corpus annotation bottleneck.

The first type of solution is using old domain labeled examples to new domain sentiment clas-

sification. Blitzer et al. (2007) investigate domain adaptation for sentiment classifiers, which could be used to select a small set of domains to annotate and their trained classifiers would transfer well to many other domains. Li and Zong (2008) study multi-domain sentiment classification, which aims to improve performance through fusing training data from multiple domains.

The second type of solution is semi-supervised sentiment classification. Sindhwani and Melville (2008) propose a semi-supervised sentiment classification algorithm that utilizes lexical prior knowledge in conjunction with unlabeled data. Dasgupta and Ng (2009) firstly mine the unambiguous reviews using spectral techniques, and then exploit them to classify the ambiguous reviews via a novel combination of active learning, transductive learning, and ensemble learning.

The third type of solution is unsupervised sentiment classification. Zagibalov and Carroll (2008) describe an automatic seed word selection for unsupervised sentiment classification of product reviews in Chinese.

However, unsupervised learning of sentiment is difficult, partially because of the prevalence of sentimentally ambiguous reviews (Dasgupta and Ng, 2009). Using multi-domain sentiment corpus to sentiment classification is also hard to apply, because each domain has a very limited amount of training data, due to annotating a large corpus is difficult and time-consuming (Li and Zong, 2008). So in this paper we focus on semi-supervised approach to sentiment classification.

3 Active Deep Networks

In this part, we propose a semi-supervised learning algorithm, Active Deep Networks (ADN), to address the sentiment classification problem with active learning. Section 3.1 formulates the ADN problem. Section 3.2 proposes the semi-supervised learning of ADN without active learning. Section 3.3 proposes the active learning method of ADN. Section 3.4 gives the ADN procedure.

3.1 Problem Formulation

There are many review documents in the dataset. We preprocess these reviews to be classified, which is similar with Dasgupta and Ng (2009). Each review is represented as a vector of unigrams, using binary weight equal to 1 for terms present in a vector. Moreover, the punctuations, numbers, and words of length one are removed from the vector. Finally, we sort the vocabulary by document frequency and remove the top 1.5%. It is because that many of these high document frequency words are stopwords or domain specific general-purpose words.

After preprocess, every review can be represented by a vector. Then the dataset can be represented as a matrix:

$$\mathbf{X}=\left[\mathbf{x}^1,\mathbf{x}^2,\ldots,\mathbf{x}^{R+T}\right]=\begin{bmatrix} x_1^1, x_1^2,\ldots, x_1^{R+T} \\ x_2^1, x_2^2,\ldots, x_2^{R+T} \\ \vdots,\vdots,\ldots,\vdots \\ x_D^1, x_D^2,\ldots, x_D^{R+T} \end{bmatrix} \quad (1)$$

where R is the number of training samples, T is the number of test samples, D is the number of feature words in the dataset. Every column of $\mathbf{X}$ corresponds to a sample $\mathbf{x}$, which is a representation of a review. A sample that has all features is viewed as a vector in $\mathbb{R}^D$, where the i^{th} coordinate corresponds to the i^{th} feature.

The L labeled samples are chosen randomly from R training samples, or chosen actively by active learning, which can be seen as:

$$\mathbf{X}^L=\mathbf{X}^R(\mathbf{S}),\mathbf{S}=[s_1,s_2,...,s_L]\ 1\le s_i\le R \quad (2)$$

where $\mathbf{S}$ is the index of selected training reviews to be labeled manually.

Let $\mathbf{Y}$ be a set of labels corresponds to L labeled training samples and is denoted as:

$$\mathbf{Y}^L=\left[\mathbf{y}^1,\mathbf{y}^2,\ldots,\mathbf{y}^L\right]=\begin{bmatrix} y_1^1, y_1^2,\ldots, y_1^L \\ y_2^1, y_2^2,\ldots, y_2^L \\ \vdots,\vdots,\ldots,\vdots \\ y_C^1, y_C^2,\ldots, y_C^L \end{bmatrix} \quad (3)$$

where C is the number of classes. Every column of $\mathbf{Y}$ is a vector in $\mathbb{R}^C$, where the j^{th} coordinate corresponds to the j^{th} class.

$$y_j^i=\begin{cases}1 & \text{if } \mathbf{x}^i\in j^{\text{th}}\text{ class} \\ -1 & \text{if } \mathbf{x}^i\notin j^{\text{th}}\text{ class}\end{cases} \quad (4)$$

For example, if a review $\mathbf{x}$ is positive, y=[1, -1]'; else, $\mathbf{y}$ = [-1, 1]'.

We intend to seek the mapping function $\mathbf{X}^L\to\mathbf{Y}^L$ using the L labeled data and R+T-L unlabeled data. After training, we can determine $\mathbf{y}$ by the trained ADN while a new sample $\mathbf{x}$ is fed.

3.2 Semi-Supervised Learning

To address the problem formulated in section 3.1, we propose a novel deep architecture for ADN method, as show in Figure 1. The deep architecture is a fully interconnected directed belief nets with one input layer $\mathbf{h}^0$, N hidden layers $\mathbf{h}^1$, $\mathbf{h}^2$, …, $\mathbf{h}^N$, and one label layer at the top. The input layer $\mathbf{h}^0$ has D units, equal to the number of features of sample data $\mathbf{x}$. The label layer has C units, equal to number of classes of label vector $\mathbf{y}$. The numbers of units for hidden layers, currently, are pre-defined according to the experience or intuition. The seeking of the mapping function $\mathbf{X}^L \to \mathbf{Y}^L$, here, is transformed to the problem of finding the parameter space $\mathbf{W}=\{\mathbf{w}^1, \mathbf{w}^2,\ldots,\mathbf{w}^N\}$ for the deep architecture.

The semi-supervised learning method based on ADN architecture can be divided into two stages: First, AND architecture is constructed by greedy layer-wise unsupervised learning using RBMs as building blocks. All the unlabeled data together with L labeled data are utilized to find the parameter space $\mathbf{W}$ with N layers. Second, ADN architecture is trained according to the exponential loss function using gradient descent method. The parameter space $\mathbf{W}$ is retrained by an exponential loss function using L labeled data.

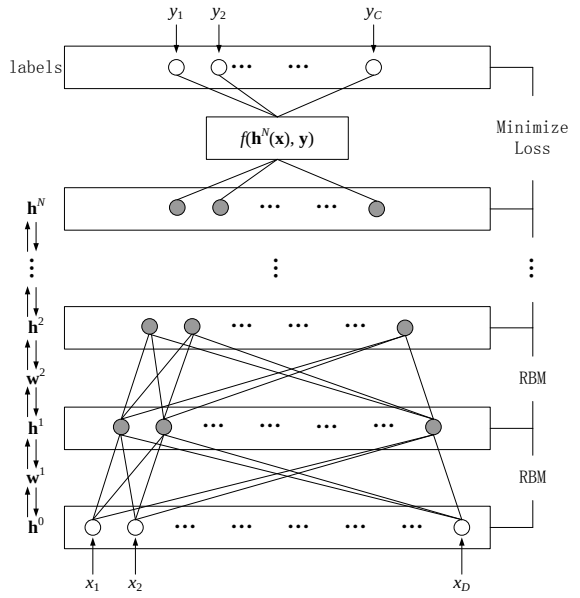


Figure 1. Architecture of Active Deep Networks

For unsupervised learning, we define the energy of the state ($\mathbf{h}^{k-1}$, $\mathbf{h}^k$) as:

$$E\left(\mathbf{h}^{k-1},\mathbf{h}^k;\theta\right)=-\sum_{s=1}^{D_{k-1}}\sum_{t=1}^{D_k} w_{st}^k h_s^{k-1} h_t^k - \sum_{s=1}^{D_{k-1}} b_s^{k-1} h_s^{k-1} - \sum_{t=1}^{D_k} c_t^k h_t^k \quad (5)$$

where $\theta=(\mathbf{w},\mathbf{b},\mathbf{c})$ are the model parameters: w_{st}^k is the symmetric interaction term between unit s in the layer $\mathbf{h}^{k-1}$ and unit t in the layer $\mathbf{h}^k$, k =1,…, N-1. b_s^{k-1} is the s^{th} bias of layer $\mathbf{h}^{k-1}$ and c_t^k is the t^{th} bias of layer $\mathbf{h}^k$. D_k is the number of unit in the k^{th} layer.

The probability that the model assigns to $\mathbf{h}^{k-1}$ is:

$$P\left(\mathbf{h}^{k-1};\theta\right)=\frac{1}{Z(\theta)}\sum_{\mathbf{h}^k}\exp\left(-E\left(\mathbf{h}^{k-1},\mathbf{h}^k;\theta\right)\right) \quad (6)$$

$$Z(\theta)=\sum_{\mathbf{h}^{k-1}}\sum_{\mathbf{h}^k}\exp\left(-E\left(\mathbf{h}^{k-1},\mathbf{h}^k;\theta\right)\right) \quad (7)$$

where $Z(\theta)$ denotes the normalizing constant. The conditional distributions over $\mathbf{h}^k$ and $\mathbf{h}^{k-1}$ are:

$$p\left(\mathbf{h}^k\mid\mathbf{h}^{k-1}\right)=\prod_t p\left(h_t^k\mid\mathbf{h}^{k-1}\right) \quad (8)$$

$$p\left(\mathbf{h}^{k-1}\mid\mathbf{h}^k\right)=\prod_s p\left(h_s^{k-1}\mid\mathbf{h}^k\right) \quad (9)$$

the probability of turning on unit t is a logistic function of the states of $\mathbf{h}^{k-1}$ and w_{st}^k:

$$p\left(h_t^k=1\mid\mathbf{h}^{k-1}\right)=\text{sigm}\left(c_t^k+\sum_s w_{st}^k h_s^{k-1}\right) \quad (10)$$

the probability of turning on unit s is a logistic function of the states of $\mathbf{h}^k$ and w_{st}^k:

$$p\left(h_s^{k-1}=1\mid\mathbf{h}^k\right)=\text{sigm}\left(b_s^{k-1}+\sum_t w_{st}^k h_t^k\right) \quad (11)$$

where the logistic function is:

$$\text{sigm}(\eta)=1/1+e^{-\eta} \quad (12)$$

The derivative of the log-likelihood with respect to the model parameter $\mathbf{w}^k$ can be obtained by the CD method (Hinton, 2002):

$$\frac{\partial \log p(\mathbf{h}^{k-1})}{\partial w_{st}}=\left\langle h_s^{k-1}h_t^k\right\rangle_{P_0}-\left\langle h_s^{k-1}h_t^k\right\rangle_{P_M} \quad (13)$$

where $\langle\cdot\rangle_{P_0}$ denotes an expectation with respect to the data distribution and $\langle\cdot\rangle_{P_M}$ denotes a distribution of samples from running the Gibbs sampler initialized at the data, for M full steps.

The above discussion is based on the training of the parameters between two hidden layers with one sample data $\mathbf{x}$. For unsupervised learning, we construct the deep architecture using all labeled data with unlabeled data by inputting them one by one from layer $\mathbf{h}^0$, train the parameter between $\mathbf{h}^0$ and $\mathbf{h}^1$. Then $\mathbf{h}^1$ is constructed, we

can use it to construct the up one layer $\mathbf{h}^2$. The deep architecture is constructed layer by layer from bottom to top, and in each time, the parameter space $\mathbf{w}^k$ is trained by the calculated data in the k-1th layer.

According to the $\mathbf{w}^k$ calculated above, the layer $\mathbf{h}^k$ can be got as below when a sample $\mathbf{x}$ is fed from layer $\mathbf{h}^0$:

$$h_t^k(\mathbf{x}) = \mathrm{sigm}(c_t^k + \sum_{s=1}^{D_{k-1}} w_{st}^k h_s^{k-1}(\mathbf{x})) \quad t = 1,\ldots,D_k \quad k = 1,\ldots,N-1 \quad (14)$$

The parameter space $\mathbf{w}^N$ is initialized randomly, just as backpropagation algorithm. Then ADN architecture is constructed. The top hidden layer is formulated as:

$$h_t^N(\mathbf{x}) = c_t^N + \sum_{s=1}^{D_{N-1}} w_{st}^N h_s^{N-1}(\mathbf{x}) \quad t = 1,\ldots,D_N \quad (15)$$

For supervised learning, the ADN architecture is trained by L labeled data. The optimization problem is formulized as:

$$\arg\min_{\mathrm{h}^N} \quad \mathrm{f}\left(\mathrm{h}^N\left(\mathbf{X}^L\right), \mathbf{Y}^L\right) \quad (16)$$

where

$$\mathrm{f}\left(\mathrm{h}^N\left(\mathbf{X}^L\right), \mathbf{Y}^L\right) = \sum_{i=1}^{L}\sum_{j=1}^{C} \mathrm{T}\left(\mathrm{h}_j^N\left(\mathbf{x}^i\right) y_j^i\right) \quad (17)$$

and the loss function is defined as

$$\mathrm{T}(r) = \exp(-r) \quad (18)$$

In the supervised learning stage, the stochastic activities are replaced by deterministic, real valued probabilities. We use gradient-descent through the whole deep architecture to retrain the weights for optimal classification.

3.3 Active Learning

Semi-supervised learning allows us to classify reviews with few labeled data. However, annotating the reviews manually is expensive, so we want to get higher performance with fewer labeled data. Active learning can help to choose those reviews that should be labeled manually in order to achieving higher classification performance with the same number of labeled data. For such purpose, we incorporate pool-based active learning with the ADN method, which accesses to a pool of unlabeled instances and requests the labels for some number of them (Tong and Koller, 2002).

Given an unlabeled pool $\mathbf{X}^R$ and a initial labeled data set $\mathbf{X}^L$ (one positive, one negative), the ADN architecture $\mathbf{h}^N$ will decide which instance in $\mathbf{X}^R$ to query next. Then the parameters of $\mathbf{h}^N$ are adjusted after new reviews are labeled and inserted into the labeled data set. The main issue for an active learner is the choosing of next unlabeled instance to query. In this paper, we choose the reviews whose labels are most uncertain for the classifier. Following previous work on active learning for SVMs (Dasgupta and Ng, 2009; Tong and Koller, 2002), we define the uncertainty of a review as its distance from the separating hyperplane. In other words, reviews that are near the separating hyperplane are chosen as the labeled training data.

After semi-supervised learning, the parameters of ADN are adjusted. Given an unlabeled pool $\mathbf{X}^R$, the next unlabeled instance to be queried are chosen according to the location of $\mathbf{h}^N(\mathbf{X}^R)$. The distance of a point $\mathbf{h}^N(\mathbf{x}^i)$ and the classes separation line $h_1^N = h_2^N$ is:

$$\mathbf{d}^i = \left|h_1^N\left(\mathbf{x}^i\right) - h_2^N\left(\mathbf{x}^i\right)\right| / \sqrt{2} \quad (19)$$

The selected training reviews to be labeled manually are given by:

$$s = \left\{ j : \mathbf{d}^j = \min\left(\mathbf{d}\right)\right\} \quad (20)$$

We can select a group of most uncertainty reviews to label at each time.

The experimental setting is similar with Dasgupta & Ng (2009). We perform active learning for five iterations and select twenty of the most uncertainty reviews to be queried each time. Then the ADN is re-trained on all of labeled and unlabeled reviews so far with semi-supervised learning. At last, we can decide the label of reviews $\mathbf{x}$ according to the output $\mathbf{h}^N(\mathbf{x})$ of the ADN architecture as below:

$$y_j = \begin{cases} 1 & \text{if } h_j^N(\mathbf{x}) = \max\left(\mathbf{h}^N(\mathbf{x})\right) \\ -1 & \text{if } h_j^N(\mathbf{x}) \neq \max\left(\mathbf{h}^N(\mathbf{x})\right) \end{cases} \quad (21)$$

As shown by Tong and Koller (2002), the BalanceRandom method, which randomly sample an equal number of positive and negative instances from the pool, has much better performance than the regular random method. So we incorporate this "Balance" idea with ADN method. However, to choose equal number of positive and negative instances without labeling the entire pool of instances in advance may not be practicable. So we present a simple way to approximate the balance of positive and negative reviews. At first, count the number of positive and negative labeled data respectively. Second,

for each iteration, classify the unlabeled reviews in the pool and choose the appropriate number of positive and negative reviews to let them equally.

3.4 ADN Procedure

The procedure of ADN is shown in Figure 2. For the training of ADN architecture, the parameters are random initialized with normal distribution. All the training data and test data are used to train the ADN with unsupervised learning. The training set $\mathbf{X}^R$ can be seen as an unlabeled pool. We randomly select one positive and one negative review in the pool to input as the initial labeled training set that are used for supervised learning. The number of units in hidden layer $D_1 \cdots D_N$ and the number of epochs Q are set manually based on the dimension of the input data and the size of training dataset. The iteration times I and the number G of active choosing data for each iteration can be set manually based on the number of labeled data in the experiment.

For each iteration, the ADN architecture is trained by all the unlabeled data and labeled data in existence with unsupervised learning and supervised learning firstly. Then we choose G reviews from the unlabeled pool based on the distance of these data from the separating line. At last, label these data manually and add them to the labeled data set. For the next iteration, the ADN architecture can be trained on the new labeled data set. At last, ADN architecture is retrained by all the unlabeled data and existing labeled data. After training, the ADN architecture is tested based on Equation (21).

The proposed ADN method can active choose the labeled data set and classify the data with the same architecture, which avoid the barrier between choosing and training with different architecture. More importantly, the parameters of ADN are trained iteratively on the label data selection process, which improve the performance of ADN. For the ADN training process: in unsupervised learning stage, the reviews can be abstracted; in supervised learning stage, ADN is trained to map the samples belong to different classes into different regions. We combine the unsupervised and supervised learning, and train parameter space of ADN iteratively. The proper data that should be labeled are chosen in each iteration, which improves the classification performance of ADN.

Active Deep Networks Procedure

Input: data $\mathbf{X}$
number of units in every hidden layer $D_1 \cdots D_N$
number of epochs Q
number of training data R
number of test data T
number of iterations I
number of active choose data for every iteration G
Initialize: $\mathbf{W}$ = normally distributed random numbers
$\mathbf{X}^L$ = one positive and one negative reviews

for i = 1 to I
Step 1. Greedy layer-wise training hidden layers using RBM
for n = 1 to N-1
for q = 1 to Q
for k = 1 to $R+T$
Calculate the non-linear positive and negative phase according to (10) and (11).
Update the weights and biases by (13).
end for
end for
end for
Step 2. Supervised learning the ADN with gradient descent
Minimize $f(h^N(\mathbf{X}),\mathbf{Y})$ on labeled data set $\mathbf{X}^L$, update the parameter space $\mathbf{W}$ according to (16).
Step 3. Choose instances for labeled data set
Choose G instances which near the separating line by (20)
Add G instances into the labeled data set $\mathbf{X}^L$
end
Train ADN with Step 1 and Step 2.

Output: ADN $h^N(\mathbf{x})$

Figure 2. Active Deep Networks Procedure.

4 Experiments

4.1 Experimental Setup

We evaluate the performance of the proposed ADN method using five sentiment classification datasets. The first dataset is MOV (Pang, et al., 2002), which is a widely-used movie review dataset. The other four dataset contain reviews of four different types of products, including books (BOO), DVDs (DVD), electronics (ELE), and kitchen appliances (KIT) (Blitzer, et al., 2007; Dasgupta and Ng, 2009). Each dataset includes 1,000 positive and 1,000 negative reviews.

Similar with Dasgupta and Ng (2009), we divide the 2,000 reviews into ten equal-sized folds randomly and test all the algorithms with cross-validation. In each folds, 100 reviews are random selected as training data and the remaining 100 data are used for test. Only the reviews in the training data set are used for the selection of labeled data by active learning.

The ADN architecture has different number of hidden units for each hidden layer. For greedy

layer-wise unsupervised learning, we train the weights of each layer independently with the fixed number of epochs equal to 30 and the learning rate is set to 0.1. The initial momentum is 0.5 and after 5 epochs, the momentum is set to 0.9. For supervised learning, we run 10 epochs, three times of linear searches are performed in each epoch.

We compare the classification performance of ADN with five representative classifiers, i.e., Semi-supervised spectral learning (Spectral) (Kamvar et al., 2003), Transductive SVM (TSVM), Active learning (Active) (Tong and Koller, 2002), Mine the Easy Classify the Hard (MECH) (Dasgupta and Ng, 2009), and Deep Belief Networks (DBN) (Hinton, et al., 2006). Spectral learning, TSVM, and Active learning method are three baseline methods for sentiment classification. MECH is a new semi-supervised method for sentiment classification (Dasgupta and Ng, 2009). DBN (Hinton, et al., 2006) is the classical deep learning method proposed recently.

4.2 ADN Performance

For MOV dataset, the ADN structure used in this experiment is 100-100-200-2, which represents the number of units in output layer is 2, in 3 hidden layers are 100, 100, and 200 respectively. For the other four data sets, the ADN structure is 50-50-200-2. The number of unit in input layer is the same as the dimensions of each datasets. All theses parameters are set based on the dimension of the input data and the scale of the dataset. Because that the number of vocabulary in MOV dataset is more than other four datasets, so the number of units in previous two hidden layers for MOV dataset are more than other four datasets. We perform active learning for 5 iterations. In each iteration, we select and label 20 of the most uncertain points, and then re-train the ADN on all of the unlabeled data and labeled data annotated so far. After 5 iterations, 100 labeled data are used for training.

The classification accuracies on test data in cross validation for five datasets and six methods are shown in Table 1. The results of previous four methods are reported by Dasgupta and Ng (2009). For ADN method, the initial two labeled data are selected randomly, so we repeat thirty times for each fold and the results are averaged. For the randomness involved in the choice of labeled data, all the results of other five methods are achieved by repeating ten times for each fold and then taking average on results.

Through Table 1, we can see that the performance of DBN is competitive with MECH. Since MECH is the combination of spectral clustering, TSVM and Active learning, DBN is just a classification method based on deep neural network, this result proves the good learning ability of deep architecture. ADN is a combination of semi-supervised learning and active learning based on deep architecture, the performance of ADN is better than all other five methods on five datasets. This could be contributed by: First, ADN uses a new architecture to guide the output vector of samples belonged to different regions of new Euclidean space, which can abstract the useful information that are not accessible to other learners; Second, ADN use an exponential loss function to maximize the separability of labeled data in global refinement for better discriminability; Third, ADN fully exploits the embedding information from the large amount of unlabeled data to improve the robustness of the classifier; Fourth, ADN can choose the useful training data actively, which also improve the classification performance.

Type	MOV	KIT	ELE	BOO	DVD
Spectral	67.3	63.7	57.7	55.8	56.2
TSVM	68.7	65.5	62.9	58.7	57.3
Active	68.9	68.1	63.3	58.6	58.0
MECH	76.2	74.1	70.6	62.1	62.7
DBN	71.3	72.6	73.6	64.3	66.7
ADN	**76.3**	**77.5**	**76.8**	**69.0**	**71.6**

Table 1. Test Accuracy with 100 Labeled Data for Five Datasets and Six Methods.

4.3 Effect of Active Learning

To test the performance of our proposed active learning method, we conduct following additional experiments.

Passive learning: We random select 100 reviews from the training fold and use them as labeled data. Then the proposed semi-supervised

learning method of ADN is used to train and test the performance. Because of randomness, we repeat 30 times for each fold and take average on results. The test accuracies of passive learning for five datasets are shown in Table 2. In comparison with ADN method in Table 1, we can see that the proposed active learning method yields significantly better results than randomly chosen points, which proves effectiveness of proposed active learning method.

Fully supervised learning: We train a fully supervised classifier using all 1,000 training reviews based on the ADN architecture, results are also shown in Table 2. Comparing with the ADN method in Table 1, we can see that employing only 100 active learning points enables us to almost reach fully-supervised performance for three datasets.

Type	MOV	KIT	ELE	BOO	DVD
Passive	72.2	75.0	75.0	66.0	67.9
Supervised	77.2	79.4	79.1	69.3	72.1

Table 2. Test Accuracy of ADN with different experiment setting for Five Datasets.

4.4 Semi-Supervised Learning with Variance of Labeled Data

To verify the performance of semi-supervised learning with different number of labeled data, we conduct another series of experiments on five datasets and show the results on Figure 3. We run ten-fold cross validation for each dataset. Each fold is repeated ten times and the results are averaged.

We can see that ADN can also get a relative high accuracy even by using just 20 labeled reviews for training. For most of the sentiment datasets, the test accuracy is increasing slowly while the number of labeled review is growing. This proves that ADN reaches good performance even with few labeled reviews.

5 Conclusions

This paper proposes a novel semi-supervised learning algorithm ADN to address the sentiment classification problem with a small number of labeled data. ADN can choose the proper training data to be labeled manually, and fully exploits the embedding information from the large amount of unlabeled data to improve the robustness of the classifier. We propose a new architecture to guide the output vector of samples belong to different regions of new Euclidean space, and use an exponential loss function to maximize the separability of labeled data in global refinement for better discriminability. Moreover, ADN can make the right decision about which training data should be labeled based on existing unlabeled and labeled data. By using unsupervised and supervised learning iteratively, ADN can choose the proper training data to be labeled and train the deep architecture at the same time. Finally, the deep architecture is re-trained using the chosen labeled data and all the unlabeled data. We also conduct experiments to verify the effectiveness of ADN method with different number of labeled data, and demonstrate that ADN can reach very competitive classification performance just by using few labeled data. This results show that the proposed ADN method, which only need fewer manual labeled reviews to reach a relatively higher accuracy, can be used to train a high performance sentiment classification system.

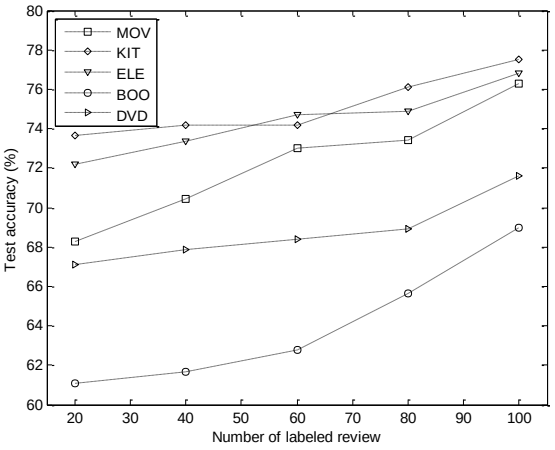


Figure 3. Test Accuracy of ADN with Different Number of Labeled Reviews for Five Datasets.

Acknowledgement

This work is supported in part by the National Natural Science Foundation of China (No. 60703015 and No. 60973076).

References

Bengio, Yoshua. 2007. *Learning deep architectures for AI*. Montreal: IRO, Universite de Montreal.

Blitzer, John, Mark Dredze, and Fernando Pereira. 2007. Biographies, Bollywood, Boom-boxes and Blenders: Domain Adaptation for Sentiment Classification. In *45th Annual Meeting of the Association of Computational Linguistics*.

Dasgupta, Sajib, and Vincent Ng. 2009. Mine the Easy, Classify the Hard: A Semi-Supervised Approach to Automatic Sentiment Classification. In *Joint Conference of the 47th Annual Meeting of the Association for Computational Linguistics and 4th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing*.

Gamon, Michael. 2004. Sentiment classification on customer feedback data: noisy data, large feature vectors, and the role of linguistic analysis. In *International Conference on Computational Linguistics*.

Hinton, Geoffrey E. . 2002. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8): 1771-1800.

Hinton, Geoffrey E. , Simon Osindero, and Yee-Whye Teh. 2006. A Fast Learning Algorithm for Deep Belief Nets. *Neural Computation*, 18: 1527-1554.

Kamvar, Sepandar, Dan Klein, and Christopher Manning. 2003. Spectral Learning. In *International Joint Conferences on Artificial Intelligence*.

Li, Shoushan, and Chengqing Zong. 2008. Multi-domain Sentiment Classification. In *46th Annual Meeting of the Association of Computational Linguistics*.

Li, Tao, Yi Zhang, and Vikas Sindhwani. 2009. A Non-negative Matrix Tri-factorization Approach to Sentiment Classification with Lexical Prior Knowledge. In *Joint Conference of the 47th Annual Meeting of the Association for Computational Linguistics and 4th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing*.

Pang, Bo, and Lillian Lee. 2004. A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts. In *42th Annual Meeting of the Association of Computational Linguistics*.

Pang, Bo, and Lillian Lee. 2008. *Opinion mining and sentiment analysis* (Vol. 2).

Pang, Bo, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up? Sentiment Classification using Machine Learning Techniques. In *Conference on Empirical Methods in Natural Language Processing*.

Raina, Rajat, Alexis Battle, Honglak Lee, Benjamin Packer, and Andrew Y. Ng. 2007. Self-taught learning: transfer learning from unlabeled data. In *International conference on Machine learning*.

Ranzato, Marc'Aurelio, and Martin Szummer. 2008. Semi-supervised learning of compact document representations with deep networks. In *International Conference on Machine learning*.

Salakhutdinov, Ruslan, and Geoffrey E. Hinton. 2007. Learning a Nonlinear Embedding by Preserving Class Neighbourhood Structure. In *Proceedings of Eleventh International Conference on Artificial Intelligence and Statistics*.

Sindhwani, Vikas, and Prem Melville. 2008. Document-Word Co-regularization for Semi-supervised Sentiment Analysis. In *International Conference on Data Mining*.

Tong, Simon, and Daphne Koller. 2002. Support vector machine active learning with applications to text classification. *Journal of Machine Learning Research*, 2: 45-66.

Wan, Xiaojun. 2009. Co-Training for Cross-Lingual Sentiment Classification. In *Joint Conference of the 47th Annual Meeting of the Association for Computational Linguistics and 4th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing*.

Xia, Yunqing, Linlin Wang, Kam-Fai Wong, and Mingxing Xu. 2008. Lyric-based Song Sentiment Classification with Sentiment Vector Space Model. In *46th Annual Meeting of the Association of Computational Linguistics*.

Zagibalov, Taras, and John Carroll. 2008. Automatic Seed Word Selection for Unsupervised Sentiment Classification of Chinese Text. In *International Conference on Computational Linguistics*.

Zhu, Xiaojin. 2007. *Semi-supervised learning literature survey*. University of Wisconsin Madison.

Dual-Space Re-ranking Model for Document Retrieval

Dong Zhou[1], **Seamus Lawless**[1], **Jinming Min**[2], **Vincent Wade**[1]

1. Center for Next Generation Localisation, University of Dublin, Trinity College
2. Center for Next Generation Localisation, Dublin City University

dongzhou1979@hotmail.com, seamus.lawless@scss.tcd.ie,
jinming.min@googlemail.com, Vincent.Wade@sccs.tcd.ie

Abstract

The field of information retrieval still strives to develop models which allow semantic information to be integrated in the ranking process to improve performance in comparison to standard bag-of-words based models. A conceptual model has been adopted in general-purpose retrieval which can comprise a range of concepts, including linguistic terms, latent concepts and explicit knowledge concepts. One of the drawbacks of this model is that the computational cost is significant and often intractable in modern test collections. Therefore, approaches utilising concept-based models for re-ranking initial retrieval results have attracted a considerable amount of study. This method enjoys the benefits of reduced document corpora for semantic space construction and improved ranking results. However, fitting such a model to a smaller collection is less meaningful than fitting it into the whole corpus. This paper proposes a dual-space model which incorporates external knowledge to enhance the space produced by the latent concept method. This model is intended to produce global consistency across the semantic space: similar entries are likely to have the same re-ranking scores with respect to the latent and manifest concepts. To illustrate the effectiveness of the proposed method, experiments were conducted using test collections across different languages. The results demonstrate that the method can comfortably achieve improvements in retrieval performance.

1 Introduction

Information retrieval often suffers from the so called *"vocabulary mismatch"* problem. A document may be semantically relevant to a query despite the fact that the specific query terms used and the terms found in the document completely or partially differ (Furnas et al., 1987). Consequently, overlap with respect to linguistic terms should not be a necessary condition in query-document similarity and methods relying on the bag-of-words model can display poor performance as a result. In order to overcome the vocabulary mismatch problem, several solutions have been suggested which exploit semantic relations between text units. Among these methods, the latent model, the explicit model and the mixed model are commonly employed.

The latent model (Landauer et al., 1998; Blei et al., 2003) tries to directly model the internal structure of "topics" or "concepts" in the text data, thus building meaningful groups beyond single words. Typically some form of dimension reduction (Fodor, 2002) is applied to the data matrix to find such latent dimensions which correspond to concepts. In contrast, the explicit model (Gabrilovich and Markovitch, 2007) indexes texts according to an external knowledge base. Typically the meaning of a piece of text is represented as a weighted vector of knowledge-based concepts derived from ex-

Coling 2008: Poster Volume, pages 1524–1532,
Beijing, August 2010

ternal resources such as ODP[1] or Wikipedia[2] articles. The mixed model (Serban et al., 2005) extends the bag-of-words vector by adding external categories derived from WordNet or similar thesaurus. Based upon these definitions, the explicit model and the mixed model are similar in nature but differ in their use of external knowledge sources.

Models such as those described above, however, have well documented drawbacks. Firstly, these methods are very computationally complex. In the latent model, complexity grows linearly with the number of dimensions and the number of documents. For example, the computational cost of singular value decomposition (SVD) is significant; no successful experiment has been reported with over one million documents (Manning et al., 2008). This has been the biggest obstacle to the widespread adoption of this kind of method. For the explicit and mixed model, the dimensions of projecting documents into the external knowledge space are often limited to ten thousand (Potthast et al., 2008) in order to facilitate the large size of the test collections used. Another problem with the explicit model is that the documents are often distributed over thousands of dimensions in which the semantic relatedness will degrade dramatically. For example, in (Sorg and Cimiano, 2008) when the whole Wikipedia collection is adopted to build the space, one document is mapped to ten thousand dimensions, in which it may only have very few truly semantically related dimensions. The means of identifying these dimensions is not reported and this may significantly influence the retrieval performance.

Therefore, researchers started to consider integrating the aforementioned models into smaller, controlled document collections to overcome these shortcomings and assist the retrieval process. (Zhou and Wade, 2009b) proposed a Latent Dirichlet Allocation (LDA)-based method to model the latent structure of "topics" deduced from the initial retrieval results. The scores obtained from this process are then combined with initial ranking scores to produce a re-ranked list of results that are superior to original ordering. The method also enjoys the benefits of fast and tractable latent semantic computation and successfully avoids the incremental build problem (Landauer et al., 1998) which commonly exists in latent semantic analysis (LSA) techniques.

There is an important factor, however, that needs to be taken into account when applying this method. Due to the smaller corpus size, fitting a latent model into this corpus is less meaningful than fitting the same model into a large, web-scale corpus. This means that some form of justification has to be applied to achieve better performance. A simple approach to address this problem is to directly apply the explicit or mixed model into a controlled corpus to improve ranking performance. A similar problem will arise in the latent model in this single semantic space, resulting in limited improvements.

To address the challenges described above, this paper proposes a dual-space model which incorporates external knowledge to enhance the semantic space produced by the latent concept method. This model is intended to produce global consistency across the semantic space: *similar entries are likely to have the same re-ranking scores with respect to the latent and manifest concepts.* In other words: in this model, if a group of documents deal with the same topic induced from a dual semantic space which shares a strong similarity with a query, the documents will get allocated similar ranking as they are more likely to be relevant to the query.

In the experiments carried out in this paper, the dual-space model is applied to ad-hoc document retrieval and compared with the initial language model-based ranker and single-space model exploiting latent and explicit features. The results show that the explicit model could only bring minor improvements over the initial ranker. The latent model delivered more significant improvements than the explicit model. Both, however, are outperformed by the dual-space model.

The main contribution of this paper is to propose a dual-space semantic model for the re-ranking problem, which aims to improve precision, especially of the most highly ranked results. Other contributions of the paper include proposing a novel way of applying the explicit model to the re-ranking problem, and performing a systematic comparison between different models.

[1] http://www.dmoz.org/
[2] http://www.wikipedia.org/

The rest of this paper is organised as follows. Related work on re-ranking and concept-based methods is briefly summarised in Section 2. Section 3 describes the latent space model and explicit space model used in the framework developed by this research, Section 4 presents details of how to build the dual-space model. In Section 5 a report is provided on a series of experiments performed over three different test collections written in English, French and German. This report includes details of the results obtained. Finally, Section 6 concludes the paper and speculates on future work.

2 Related Work

There exist several strands of related work in the areas of re-ranking and concept-based document retrieval.

A family of work on the structural re-ranking paradigm over different sized document corpora was proposed to refine initial ranking scores. Kurland and Lee performed re-ranking based on measures of centrality in the graph formed by the generation of links induced by language model scores, through a weighted version of the PageRank algorithm (Kurland and Lee, 2005) and a HITS-style cluster-based approach (Kurland and Lee, 2006). Zhang et al. (Zhang et al., 2005) proposed a similar method to improve web search based on a linear combination of results from text search and authority ranking. The graph, which they named an "affinity graph", shares strong similarities with Kurland and Lee's work where the links are induced by a modified version of cosine similarity using the vector space model. Diaz (Diaz, 2005) used score regularisation to adjust document retrieval rankings from an initial retrieval by a semi-supervised learning method. Deng et al. (Deng et al., 2009) further developed this method by building a latent space graph based on content and explicit link information. Unlike their approach this research attempts to model the explicit information directly.

The latent concept retrieval model has a long history in information retrieval. (Dumais, 1993; Dumais, 1995) conducted experiments with latent semantic indexing (LSI) on TREC[3] documents and tasks. These experiments achieved precision at, or above, that of the median TREC participant. On about 20% of TREC topics this system was the top scorer, and reportedly slightly better than average results in comparison to standard vector spaces for LSI at about 350 dimensions. (Hofmann, 1999) provides an initial probabilistic extension of the basic latent semantic indexing technique. A more satisfactory formal basis for a probabilistic latent variable model for dimensionality reduction is the LDA model (Blei et al., 2003), which is generative and assigns probabilities to documents outside of the training set. Wei and Croft (Wei and Croft, 2006) presented the first large-scale evaluation of LDA, finding it to significantly outperform the query likelihood model. (Zhou and Wade, 2009b; Zhou and Wade, 2009a) successfully applied this method to document re-ranking and achieved significant improvement over language model-based ranking and various graph-based re-ranking methods.

The explicit concept model has recently attracted much attention in the information retrieval community. Notably, explicit semantic analysis (ESA) has been proposed as an approach to computing semantic relatedness between words and thus, has a natural application in this field (Gabrilovich and Markovitch, 2007). In essence, ESA indexes documents with respect to the Wikipedia article space, indicating how strongly a given word in the document is associated to a specific Wikipedia article. In this model, each article is regarded as a concept, an analogical unit used in the latent model. As in the latent model, two words or texts can be semantically related in spite of not having any words in common. Specifically, this method has been widely adopted in cross-language information retrieval (CLIR) as an approach to resolving an extreme case of the vocabulary mismatch problem, where queries and documents are written in different languages (Potthast et al., 2008). (Anderka et al., 2009) showed that this approach has comparable performance to linguistic matching methods. (Cimiano et al., 2009) compared this method with a latent concept model based on LSI/LDA and concluded that it will outperform the latent model if trained on Wikipedia articles.

[3] http:// trec.nist.gov

3 Latent and Explicit Models

In this section, an overview of the problem addressed by this paper is presented and the latent and explicit document re-ranking models are described in more detail. This section also demonstrates how these models can be used in a re-ranking setting.

3.1 Problem Definition

Let $\mathbb{D} = \{d_1, d_2, \dots, d_n\}$ denote the set of documents to be retrieved. Given a query q, a set of initial results $\mathbb{D}_{init} \in \mathbb{D}$ of top documents are returned by a standard information retrieval model (initial ranker). However, typically the performance of the initial ranker can be improved upon. The purpose of the re-ranking method developed by this research is to re-order a set of documents $\mathbb{D}'_{init}$ so as to improve retrieval accuracy at the most highly ranked results.

3.2 Latent Concept Model

The specific method used here is borrowed from (Zhou and Wade, 2009b), which is based on the LDA model. The topic mixture is drawn from a conjugate Dirichlet prior that remains the same for all documents. The process of generating a document corpus is as follows:

1) Pick a multinomial distribution $\vec{\varphi}_z$ for each topic k from a Dirichlet distribution with hyperparameter $\vec{\beta}$.
2) For each document d, pick a multinomial distribution $\vec{\theta}_d$, from a Dirichlet distribution with hyperparameter $\vec{\alpha}$.
3) For each word token w in document d, pick a topic $z \in \{1 \dots k\}$ from the multinomial distribution $\vec{\theta}_d$.
4) Pick word w from the multinomial distribution $\vec{\varphi}_z$.

LDA possesses fully consistent generative semantics by treating the topic mixture distribution as a k-parameter hidden random variable. LDA offers a new and interesting framework to model a set of documents. The documents and new text sequences (for example, queries) can easily be connected by "mapping" them to the topics in the corpus.

In a re-ranking setting, the probability that a document d generates w is estimated using a mixture model LDA. It uses a convex combination of a set of component distributions to model observations. In this model, a word w is generated from a convex combination of some hidden topics z:

$$LDA_d(w) = \sum_{z=1}^{k} p(w|z)p(z|d)$$

where each mixture model $p(w|z)$ is a multinomial distribution over terms that correspond to one of the latent topics z. This could be generated to give a distribution on a sequence of text:

$$LDA_d(w_1 w_2 \dots w_n) \stackrel{\text{def}}{=} \prod_{j=1}^{n} LDA_d(w_j)$$

Then the distance between a query and a document based on this model can be obtained. The method used here adopts the KL divergence (Baeza-Yates and Ribeiro-Neto, 1999) between the query terms and document terms to compute a Re-Rank score RS_{LDA}^{KL}:

$$RS_{LDA}^{KL} = -D(MLE_q(\cdot)||LDA_d(\cdot))$$

The final score is then obtained through a linear combination of the re-ranking scores based on the initial ranker and the latent document re-ranker, shown as follows:

$$RS_{Latent}^{LDA} = \lambda \cdot OS + (1 - \lambda) \cdot RS_{LDA}^{KL}$$

where OS denotes original scores returned by the initial ranker and λ is a parameter that can be tuned with $\lambda = 1$ meaning no re-ranking is performed.

Another well-known approach to the latent model is the LSI method. It is based on SVD, a technique from linear algebra. This method has not been reported anywhere previously for re-ranking purposes. It has been included here to compare the effectiveness of different latent approaches. As a full SVD is a loss-free decomposition of a matrix M, which is decomposed into two orthogonal matrices U and V and a diagonal matrix Σ. Estimating less singular values and their corresponding singular vectors leads to reduced dimensions resembling latent concepts so that documents are no longer represented by terms but by concepts. New documents (queries) are represented in terms of concepts by folding them into the LSI model. Next, cosine similarities may be used to compute the similarity between a query and a document to obtain RS_{LSI}^{COS} and combine it with the original score to produce the final re-ranking score:

Collection	Contents	Language	Num of docs	Size	Queries
BL (CLEF2009)	British Library Data	English (Main)	1,000,100	1.2 GB	50
BNF (CLEF2009)	Bibliothèque Nationale de France	French (Main)	1,000,100	1.3 GB	50
ONB (CLEF2009)	Austrian National Library	German (Main)	869,353	1.3 GB	50

Table 1. Statistics of test collections

$$RS_{latent}^{LSI} = \lambda' \cdot OS + (1-\lambda') \cdot RS_{LSI}^{COS}$$

3.3 Explicit Concept Model

As an example of explicit concept model (Gabrilovich and Markovitch, 2007), explicit semantic analysis attempts to index or classify a given text t with respect to a set of explicitly given external categories. The basic idea is to take as input a document d and map it to a high-dimensional, real-valued vector space. This space is spanned by a Wikipedia database $W_l = \{a_1, \ldots, a_n\}$.This mapping is given by the following function:

$$\Phi_l : T \to \mathbb{R}^{|W_l|}$$
$$\Phi_l(t) \coloneqq \langle v_1, \ldots, v_{|W_l|} \rangle$$

Where $|W_l|$ is the number of articles in Wikipedia W_l corresponding to language l. The value v_i in the vector t expresses the strength of association between t and the Wikipedia article a_i and is defined as the cosine similarity:

$$RS_{ESA}^{COS} = \frac{\langle t, a_i \rangle}{\| t \| \| a_i \|}$$

As pointed out in section 1, documents are often distributed over thousands of dimensions in which the semantic relatedness will degrade dramatically. The main purpose is to find the most relevant dimensions with respect to queries. To apply this method to re-ranking, W_l is limited to the number of highly relevant documents for a given query. In other words, the entire set of Wikipedia articles in language l is retrieved, and only return a specific number of documents as in W_l. This modification will also lead to fast computation of scores compared to scanning through the whole Wikipedia collection.

Similar to the latent model described above, the final ranking score is defined as:

$$RS_{Explicit}^{ESA} = \mu \cdot OS + (1-\mu) \cdot RS_{ESA}^{COS}$$

4 Dual space model

Armed with the latent and explicit models, the dual-space model proposed by this paper is now described. In order to make a direct connection between the two models, the key point is to make the dimensions comparable across different models. The detail presented on the latent and explicit concept models in the previous section did not describe how to define a specific number of dimensions. A simple assumption is taken here in the dual-space model: the number of dimensions produced by the explicit model has to correspond to the number of dimensions induced by the latent model. As the same group of documents are being mapped into two different semantic spaces, it is assumed that the concepts induced by the latent model reflect the hidden structures in this document collection. Therefore, the same phenomenon should be observed when applying the explicit model and vice-versa. Based on this assumption, the dual-space model could be conducted so as to make a constraint:

$$|W_l| = k$$

and the final ranking score for this dual space is:

$$RS_{dual}^{LDA} = \varsigma \cdot OS + (1-\varsigma-\tau) \cdot RS_{LDA}^{KL} + \tau \cdot RS_{ESA}^{COS}$$

or

$$RS_{dual}^{LSI} = \varsigma \cdot OS + (1-\varsigma-\tau) \cdot RS_{LSI}^{COS} + \tau \cdot RS_{ESA}^{COS}$$

4 Experiments and Results

In this section, an empirical study of the effectiveness of the dual-space model over three data collections written in English, French and German is presented.

4.1 Experimental Setup

The text corpus used in the experiment described below consisted of elements of the CLEF-2008[4] and CLEF-2009 European Library (TEL) collections[5] written in English, French and German. These collections are described in greater detail in Table 1. All of the documents in the experiment were indexed using the Terrier toolkit[6]. Prior to indexing, Porter's stemmer and a stopword list[7] were used for the English documents. A French and German analyser[8] is used to analyse French and German documents.

It is worth noting that the CLEF TEL data is actually multilingual: all collections to a greater or lesser extent contain records pointing to documents in other languages. However this is not a major problem because the majority of documents in the test collection are written in the primary language of those test collections (BL-English, BNF-French, ONB-German). Please refer to (Ferro and Peters, 2009) for a more detailed discussion about this data. These collections were chosen to test the scalability of the proposed method in different settings and over different languages.

The CLEF-2008 and CLEF-2009 query sets were also used. Both query sets consist of 50 topics in each language being tested. The CLEF-2008 queries written in English were used in training the parameters and all of the CLEF-2009 queries were used in the experiment for testing purposes. Each topic is composed of several parts, including: *Title*, *Description* and *Narrative*. *Title+Description* combinations were chosen as queries. The queries are processed similarly to the treatment of the test collections. The relevance judgments are taken from the judged pool of top retrieved documents by various participating retrieval systems from previous CLEF workshops. The initial ranker used in this study is the classic vector space model. This was selected to facilitate the LSI and ESA models used and the main purpose of the experiments is to compare different models in addition to demonstrating the effectiveness of the dual-space model.

A Wikipedia database in English, French and German was used as an explicit concept space. Only those articles that are connected via cross-language links between all three Wikipedia databases were selected. A snapshot was obtained on the 29/11/2009, which contained an aligned collection of 220,086 articles in all three languages.

The following evaluation metrics were chosen to measure the effectiveness of the various approaches: mean average precision (MAP), the precision of the top 5 documents (Prec@5), the precision of the top 10 documents (Prec@10), normalised discounted cumulative gain (NDCG) and Bpref. Statistically-significant differences in performance were determined using a paired t-test at a confidence level of 95%.

4.2 Parameter Tuning

Three primary categories of parameter combinations need to be determined in the experiments. For the latent re-ranking experiments, the parameters λ, λ' must be defined. For the explicit model the parameter μ must be chosen. For both models, the weights ς, τ have to be determined. In addition, the number of dimensions $|W_l|$ and k must be specified. Settings for these parameters were optimised with respect to MAP over the BL collection using CLEF-2008 English queries and were applied to all three collections. This optimisation was not conducted for the other metrics used.

The search ranges for these two parameters were:

$\lambda, \lambda', \mu, \varsigma, \tau$: 0.1, 0.2, …, 0.9

$|W_l|, k$: 5, 10, 15, …, 40

Note that parameters ς and τ are the weights assigned to the latent model and the explicit model in the dual-space model. The choice of one will have direct influence over another. As it turned out, for many instances, the optimal value of λ, λ' with respect to MAP was either 0.3 or 0.4, suggesting the initial retrieval scores still contain valuable information. In contrast, parameter μ shows no obvious difference in performance when the value is above 0.1. With this observation, when setting the parameters ς and τ more weight is assigned to the latent model rather than the explicit model. The optimal

[4] The test collections used in CLEF-2008 and CLEF-2009 are in fact identical.
[5] http://www.clef-campaign.org
[6] http://terrier.org
[7] ftp://ftp.cs.cornell.edu/pub/smart/
[8] http://lucene.apache.org/

	Dual space build upon LDA and ESA				**Dual space build upon LSI and ESA**			
	BL				BL			
	initial ranker	latent space	explicit space	dual space	initial ranker	latent space	explicit space	dual space
Precision@5	0.508	0.528	0.514	0.54*	0.508	0.54*	0.508	**0.556***
Precision@10	0.468	0.498*	0.47	0.508*	0.468	0.51*	0.48	**0.512***
Precision@20	0.408	0.424	0.41	**0.435***	0.408	0.408	0.407	0.409
NDCG	0.4053	0.4137*	0.4053	0.416*	0.4053	0.4145*	0.4055	**0.4213***
MAP	0.2355	0.2433*	0.2358	**0.2499***	0.2355	0.2478*	0.236	**0.2499***
R-Precision	0.316	0.3243	0.3165	**0.3248**	0.316	0.3173	0.3202*	0.3232
bpref	0.271	0.2746	0.2725	0.2812	0.271	0.2836*	0.2714	**0.2879***
	BNF				BNF			
	initial ranker	latent space	explicit space	dual space	initial ranker	latent space	explicit space	dual space
Precision@5	0.376	0.368	0.372	0.376	0.376	0.376	0.376	**0.384***
Precision@10	0.346	0.352*	0.35	0.352	0.346	0.348	0.35	**0.354***
Precision@20	0.297	0.297	0.297	0.3*	0.297	**0.303**	0.299	0.3*
NDCG	0.3162	0.3158	0.3156	0.3163	0.3162	0.317	0.3164	**0.3178**
MAP	0.1621	0.1622	0.162	**0.1634**	0.1621	0.1629	0.1622	0.1624
R-Precision	0.2274	0.2279	0.2211	**0.2285**	0.2274	0.2278	0.2264	0.2277
bpref	0.1897	0.1899	0.1887	0.19	0.1897	0.1914	0.1892	**0.1918**
	ONB				ONB			
	initial ranker	latent space	explicit space	dual space	initial ranker	latent space	explicit space	dual space
Precision@5	0.38	0.388	0.36	0.404*	0.38	0.4	0.364	**0.412***
Precision@10	0.308	0.322	0.302	**0.332***	0.308	0.324	0.302	0.324
Precision@20	0.246	0.252	0.252	**0.259***	0.246	0.247	0.251	0.252
NDCG	0.3042	0.304	0.3059	0.3101	0.3042	0.3152*	0.3062	**0.3154***
MAP	0.1482	0.1524	0.1509	0.1567*	0.1482	0.1567*	0.1494	**0.1578***
R-Precision	0.2115	0.2152	0.2137	**0.2175**	0.2115	0.212	0.2106	0.2128
bpref	0.1778	0.1871	0.1799	**0.1896**	0.1778	0.1833	0.1788	0.1832

Table 2. Experimental Results. For each evaluation setting, statistically significant differences between different methods and the initial ranker are indicated by star. Bold highlights the best results over all algorithms.

value of k was between 25 and 35 for the LDA based model and between 5 and 15 for the LSI based model. Although this demonstrates a relatively large variance, the differences in terms of MAP have remained small and statistically insignificant. $\mathbb{D}_{init}$ is set to 50 in all results reported.

4.3 Results

Primary Evaluation The main experimental results, which describe the performance of the different re-ranking algorithms on the CLEF document collection, are shown in Table 2. The first four rows in each test collection specify the most important measurements because this research is particularly interested in performance over the most highly ranked results. As illus-

trated by the data, the initial ranker was always the lowest performer in terms of nearly all measurements. This indicates the need for re-ranking. Using the method computed by the explicit space always led to an improvement in retrieval effectiveness. But this improvement is only minor in comparison to the other two models and the results are often statistically insignificant. When the re-ranking score was calculated using the latent model, retrieval effectiveness always exceeded initial ranker and the explicit model. There was a noticeable improvement in retrieval effectiveness in the English collection (BL, statistically significant results were often observed), but a modest increase for the other two collections (BNF and ONB).

The empirical results obtained using the dual space model are very promising. Pleasingly, both the LDA+ESA and LSI+ESA models outperformed the basic latent and explicit space model in the majority of retrieval runs, with the best scores relating to the LSI-based models. An important phenomenon is that statistically significant improvements are always recorded in the metrics which measure the most highly ranked results. An even more exciting observation is that in many cases, the dual-space model, even though tuned for MAP, can outperform various baselines and other models for all the evaluation metrics, with statistically significant improvements in many runs.

Another observation that can be drawn from Table 2 is that the relative performance tends to be stable across test collections written in different languages. This indicates a promising future for studying document structure with respect to latent and explicit semantic space for re-ranking purposes.

The Comparison of Latent Methods Table 2 also shows a side-by-side comparison of the various performance measurements between the latent model used in this research on the CLEF-2009 BL test collection. The LSI-based method appeared to outscore the LDA-based method in the latent model in the vast majority of cases, while the difference between the various scorings was fairly marginal as both methods deliver statistically significant results. For the dual-space model, similar results were observed. A possible reason is that the initial ranker used was based on the vector space model and LSI is also vector based. It shows that more research with respect to the latent model selection will be necessary in the future.

Effectiveness of Explicit Methods As part of experimental objectives of this research, it was also necessary to test the newly developed explicit model for re-ranking. In the parameter tuning section, the explicit model displayed no obvious difference in terms of combination effectiveness. However, some variations could be observed when applying different dimensions where statistically significant results often appear in lower dimensions. This confirms the need to find more relevant dimensions, both for performance and efficiency purposes.

5 Conclusion and Future Work

This paper proposed and evaluated a dual-space document re-ranking method for re-ordering the initial retrieval results. The key to refining the results is the global consistency over the semantic space, which leverages latent and explicit semantic information and results in state-of-art performance. This paper also proposed a novel way to apply the explicit model to the re-ranking problem, and performed a systematic comparison between different models.

Further investigation is planned in many research directions. It has been shown that the latent model-based retrieval is a promising method for ranking the whole corpus. There is a desire to call for a direct comparison between ranking and re-ranking using the proposed algorithmic variations. Future work will also include identifying improvements upon linear combination for engineering different models. At the same time, there exist a sufficient number of latent and explicit semantic techniques which will be explored to compare their performance.

Acknowledgments

The authors would like to thank the three anonymous reviewers for their many constructive comments. This research is supported by the Science Foundation Ireland (Grant 07/CE/I1142) as part of the Centre for Next Generation Localisation (www.cngl.ie) at University of Dublin, Trinity College and Dublin City University.

References

Anderka, Maik, Nedim Lipka and Benno Stein. 2009. Evaluating Cross-Language Explicit Semantic Analysis and Cross Querying at TEL@CLEF 2009. In *CLEF 2009 Workshop*, Corfu, Greece.

Baeza-Yates, Ricardo A. and Berthier Ribeiro-Neto. 1999. *Modern Information Retrieval*, Addison-Wesley Longman Publishing Co., Inc.

Blei, David M., Andrew Y. Ng and Michael I. Jordan. 2003. Latent dirichlet allocation. *J. Mach. Learn. Res.* **3**: 993-1022.

Cimiano, Philipp, Antje Schultz, Sergej Sizov, Philipp Sorg and Steffen Staab. 2009. Explicit versus latent concept models for cross-language information retrieval. In *Proceedings of the 21st international jont conference on Artifical intelligence*, Pasadena, California, USA, Morgan Kaufmann Publishers Inc. p. 1513-1518.

Deng, Hongbo, Michael R. Lyu and Irwin King. 2009. Effective latent space graph-based re-ranking model with global consistency. In *Proceedings of the Second ACM WSDM conference*, Barcelona, Spain, ACM. p. 212-221.

Diaz, Fernando. 2005. Regularizing ad hoc retrieval scores. In *Proceedings of the 14th ACM CIKM conference*, Bremen, Germany, ACM. p. 672-679.

Dumais, Susan T. 1993. Latent semantic indexing (LSI) and TREC-2. In *Proceedings of TREC*. p. 105-115.

Dumais, Susan T. 1995. Latent semantic indexing (LSI): TREC-3 report. In *Proceedings of TREC*. p. 219-230.

Ferro, Nicola and Carol Peters. 2009. CLEF 2009 Ad Hoc Track Overview: TEL & Persian Tasks. In *Working notes of CLEF2008*, Corfu, Greece.

Fodor, Imola K. 2002. A Survey of Dimension Reduction Techniques. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.8.5098. Accessed: 18th April 2010.

Furnas, G. W. , T. K. Landauer, L. M. Gomez and S. T. Dumais. 1987. The vocabulary problem in human-system communication. *Commun. ACM* **30**(11): 964-971.

Gabrilovich, Evgeniy and Shaul Markovitch. 2007. Computing semantic relatedness using Wikipedia-based explicit semantic analysis. In *Proceedings of the 20th international joint conference on Artifical intelligence*, Hyderabad, India, Morgan Kaufmann Publishers Inc. p. 1606-1611.

Hofmann, Thomas. 1999. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference*, Berkeley, California, United States, ACM. p. 50-57.

Kurland, Oren and Lillian Lee. 2005. PageRank without hyperlinks: structural re-ranking using links induced by language models. In *Proceedings of the 28th annual international ACM SIGIR conference,* Salvador, Brazil, ACM. p. 306-313.

Kurland, Oren and Lillian Lee. 2006. Respect my authority!: HITS without hyperlinks, utilizing cluster-based language models. In *Proceedings of the 29th annual international ACM SIGIR conference*, Seattle, Washington, USA, ACM. p. 83-90.

Landauer, Thomas K., Peter W. Foltz and Darrell Laham. 1998. An Introduction to Latent Semantic Analysis. *Discourse Processes* **25**: 259-284.

Manning, Christopher D., Prabhakar Raghavan and Hinrich Schtze. 2008. *Introduction to Information Retrieval*, Cambridge University Press.

Potthast, Martin, Benno Stein and Maik Anderka. 2008. A Wikipedia-Based Multilingual Retrieval Model. In *Proceedings of 30th European Conference on Information Retrieval*, Glasgow, Scotland, Springer. p. 522-530.

Serban, Radu, Annette ten Teije, Frank van Harmelen, Mar Marcos and Cristina Polo. 2005. Ontology-driven extraction of linguistic patterns for modelling clinical guidelines. *Proceedings of the 10th European Conference on Artificial Intelligence in Medicine (AIME-05)*.

Wei, Xing and W. Bruce Croft. 2006. LDA-based document models for ad-hoc retrieval. In *Proceedings of the 29th annual international ACM SIGIR conference*, Seattle, Washington, USA, ACM. p. 178-185.

Zhang, Benyu, Hua Li, Yi Liu, Lei Ji, Wensi Xi, Weiguo Fan, Zheng Chen and Wei-Ying Ma. 2005. Improving web search results using affinity graph. In *Proceedings of the 28th annual international ACM SIGIR conference*, Salvador, Brazil, ACM. p. 504-511.

Zhou, Dong and Vincent Wade. 2009a. Language Modeling and Document Re-Ranking: Trinity Experiments at TEL@CLEF-2009. In *CLEF 2009 Workshop*, Corfu, Greece.

Zhou, Dong and Vincent Wade. 2009b. Latent Document Re-Ranking. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, Singapore, ACL. p. 1571-1580.

All in Strings: a Powerful String-based Automatic MT Evaluation Metric with Multiple Granularities

Junguo Zhu[1], Muyun Yang[1], Bo Wang[2], Sheng Li[1], Tiejun Zhao[1]

[1] School of Computer Science and Technology, Harbin Institute of Technology
{jgzhu; ymy; tjzhao; lish}@mtlab.hit.edu.cn
[2] School of Computer Science and Technology, Tianjin University
bo.wang.1979@gmail.com

Abstract

String-based metrics of automatic machine translation (MT) evaluation are widely applied in MT research. Meanwhile, some linguistic motivated metrics have been suggested to improve the string-based metrics in sentence-level evaluation. In this work, we attempt to change their original calculation units (granularities) of string-based metrics to generate new features. We then propose a powerful string-based automatic MT evaluation metric, combining all the features with various granularities based on SVM rank and regression models. The experimental results show that i) the new features with various granularities can contribute to the automatic evaluation of translation quality; ii) our proposed string-based metrics with multiple granularities based on SVM regression model can achieve higher correlations with human assessments than the state-of-art automatic metrics.

1 Introduction

The automatic machine translation (MT) evaluation has aroused much attention from MT researchers in the recent years, since the automatic MT evaluation metrics can be applied to optimize MT systems in place of the expensive and time-consuming human assessments. The state-of-art strategy to automatic MT evaluation metrics estimates the system output quality according to its similarity to human references. To capture the language variability exhibited by different reference translations, a tendency is to include deeper linguistic information into machine learning based automatic MT evaluation metrics, such as syntactic and semantic information (Amigò et al., 2005; Albrecht and Hwa, 2007; Giménez and Màrquez, 2008). Generally, such efforts may achieve higher correlation with human assessments by including more linguistic features. Nevertheless, the complex and variously presented linguistic features often prevents the wide application of the linguistic motivated metrics.

Essentially, linguistic motivated metrics introduce additional restrictions for accepting the outputs of translations (Amigó et al., 2009). With more linguistic features attributed, the model is actually capturing the sentence similarity in a finer granularity. In this sense, the practical effect of employing various linguistic knowledge is changing the calculation units of the matching in the process of the automatic evaluation.

Similarly, the classical string-based metrics can be changed in their calculation units directly. For example, the calculation granularity in BLEU (Papineni et al., 2002) metric is word: n-grams are extracted on the basis of single word as well as adjacent multiple words. And the calculation granularity in PosBLEU (Popović and Ney, 2009) metric is Pos tag, which correlate well with the human assessments. Therefore, it is straight forward to apply the popular string-based automatic evaluation metrics, such as BLEU, to compute the scores of the systems outputs in the surface or linguis-

Coling 2008: Poster Volume, pages 1533–1540,
Beijing, August 2010

tic tag sequences on various granularities levels.

In this paper, we attempt to change the original calculation units (granularities) of string-based metrics to generate new features. After that, we propose a powerful string-based automatic MT evaluation metric, combining all the features with various granularities based on SVM rank (Joachims, 2002) and regression (Drucker et al., 1996) models. Our analysis indicates that: i) the new features with various granularities can contribute to the automatic evaluation of translation quality; ii) our proposed string-based metrics with multiple granularities based on SVM regression model can achieve higher correlations with human assessments than the state-of-art automatic metrics .

The remainder of this paper is organized as follows: Section 2 reviews the related researches on automatic MT evaluation. Section 3 describes some new calculation granularities of string-based metrics on sentence level. In Section 4, we propose string-based metrics with multiple granularities based on SVM rank and regression models. In Section 5, we present our experimental results on different sets of data. And conclusions are drawn in the Section 6.

2 Related Work on Automatic Machine Translation Evaluation

The research on automatic string-based machine translation (MT) evaluation is targeted at a widely applicable metric of high consistency to the human assessments. WER (Nießen et al., 2000), PER (Tillmann et al., 1997), and TER (Snover et al., 2006) focuses on word error rate of translation output. GTM (Melamed et al., 2003) and the variants of ROUGE (Lin and Och, 2004) concentrate on matched longest common substring and discontinuous substring of translation output according to the human references. BLEU (Papineni et al., 2002) and NIST (Doddington, 2002) are both based on the number of common n-grams between the translation hypothesis and human reference translations of the same sentence. BLEU and NIST are widely adopted in the open MT evaluation campaigns; however, the NIST MT evaluation in 2005 indicates that they can even error in the system level (Le and Przybocki, 2005). Callison-Burch et al. (2006) detailed the deficits of the BLEU and other similar metrics, arguing that the simple surface similarity calculation between the machines translations and the human translations suffers from morphological issues and fails to capture what are important for human assessments.

In order to attack these problems, some metrics have been proposed to include more linguistic information into the process of matching, e.g., Meteor (Banerjee and Lavie, 2005) metric and MaxSim (Chan nad Ng, 2008) metrics, which improve the lexical level by the synonym dictionary or stemming technique. There are also substantial studies focusing on including deeper linguistic information in the metrics (Liu and Gildea, 2005; Owczarzak et al., 2006; Amigó et al., 2006; Mehay and Brew, 2007; Giménez and Màrquez, 2007; Owczarzak et al., 2007; Popovic and Ney, 2007; Giménez and Màrquez, 2008b).

A notable trend improving the string-based metric is to combine various deeper linguistic information via machine learning techniques in the metrics (Amigò et al., 2005; Albrecht and Hwa, 2007; Giménez and Màrquez, 2008). Such efforts are practically amount of introducing additional linguistic restrictions into the automatic evaluation metrics (Amigó et al, 2009), achiving a higher performance at the cost of lower adaptability to other languages owing to the language dependent linguistics features.

Previous work shows that including the new features into the evaluation metrics may benefit to describe nature language accurately. In this sense, the string-based metrics will be improved, if the finer calculation granularities are introduced into the metrics.

Our study analyzes the role of the calculation granularities in the performance of metrics. We find that the new features with various granularities can contribute to the automatic evaluation of translation quality. Also we propose a powerful string based automatic MT evaluation metric with multiple granularities combined by SVM. Finally, we seek a finer feature set of metrics with multiple calculation granularities.

3 The New Calculation Granularities of String-based Metrics on Sentence Level

The string-based metrics of automatic machine translation evaluation on sentence level adopt a common strategy: taking the sentences of the documents as plain strings. Therefore, when changing the calculation granularities of the string-based metrics we can simplify the information of new granularity with plain strings. In this work, five kinds of available calculation granularities are defined: "Lexicon", "Letter", "Pos", "Constitute" and "Dependency".

Lexicon: The calculation granularity is common word in the sentences of the documents, which is popular practice at present.

Letter: Split the granularities of "Lexical" into letters. Each letter is taken as a matching unit.

Pos: The Pos tag of each "Lexicon" is taken as a matching unit in this calculation granularity.

Constitute: Syntactic Constitutes in a tree structure are available through the parser tools. We use Stanford Parser (Klein and Manning, 2003a; Klein and Manning, 2003b) in this work. The Constitute tree is changed into plain string, travelling by BFS (Breadth-first search traversal) [1].

Dependency: Dependency relations in a dependency structure are also available through the parser tools. The dependency structure can also be formed in a tree, and the same processing of being changed into plain string is adopted as "Constitute".

The following serves as an example:

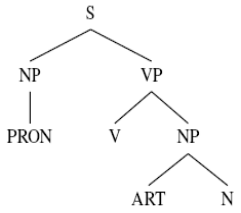
Sentence:

I have a dog

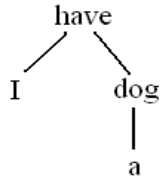
Pos tag:

I/PRON have/V a/ART dog/N

Constitute tree:



Dependency tree:



Then, we can change the sentence into the plain string in multiple calculation granularities as follows:

Lexicon string:

I have a dog

Letter string:

I h a v e a d o g

Pos string:

PRON V ART N

Constitute string:

PRON V ART N NP NP VP S

Dependency string:

a I dog have

The translation hypothesis and human reference translations are both changed into those strings of various calculation granularities. The strings are taken as inputs of the string-based automatic MT evaluation metrics. The outputs of each metric are calculated on different matching units.

4 String-based Metrics with Multiple Granularities Combined by SVM

Introducing machine learning methods to established MT evaluation metric is a popular trend. Our study chooses rank and regression support vector machine (SVM) as the learning model. Features are important for the SVM models.

Plenty of scores can be generated from the proposed metrics. In fact, not all these features are needed. Therefore, feature selection should be a necessary step to find a proper feature set and alleviate the language dependency by using fewer linguistic features.

Feature selection is an NP-Complete problem; therefore, we adopt a greedy selection algorithm called "Best One In" to find a local optimal feature set. Firstly, we select the feature among all the features which best correlates with the human assessments. Secondly, a feature among the rest features is added in to the feature set, if the correlation with the human assessments of the metric using new set is

[1] We also attempt some other traversal algorithms, including preorder, inorder and postorder traversal, the performance are proved to be similar.

the highest among all new metrics and higher than the previous metric in cross training corpus. The cross training corpus is prepared by dividing the training corpus into five parts. Each four parts of the five are for training and the rest one for testing; then, we integrate scores of the five tests as scores of cross training corpus. The five-fold cross training can help to overcome the overfitting. At the end, the feature selection stops, if adding any of the rest features cannot lead to higher correlation with human assessments than the current metric.

5 Experiments

5.1 The Impact of the Calculation Granularities on String-based Metrics

In this section, we use the data from NIST Open MT 2006 evaluation (LDC2008E43), which is described in Table 1. It consists of 249 source sentences that were translated by four human translators as well as 8 MT systems. Each machine translated sentence was evaluated by human judges for their adequacy on a 7-point scale.

	NIST 2002	**NIST 2003**	**NIST Open MT 2006**
LDC corpus	LDC2003 T17	LDC2006 T04	LDC2008 E43
Type	Newswire	Newswire	Newswire
Source	Chinese	Chinese	Arabic
Target	English	English	English
# of sentences	878	919	249
# of systems	3	7	8
# of references	4	4	4
Score	1-5, adequacy & fluency	1-5, adequacy & fluency	1-7 adequacy

Table 1: Description of LDC2006T04, LDC2003T17 and LDC2008E43

To judge the quality of a metric, we compute Spearman rank-correlation coefficient, which is a real number ranging from -1 (indicating perfect negative correlations) to +1 (indicating perfect positive correlations), between the metric's scores and the averaged human assessments on test sentences.

We select 21 features in "lexicon" calculation granularity and 11×4 in the other calculation granularities. We analyze the correlation with human assessments of the metrics in multiple calculation granularities. Table 2 lists the optimal calculation granularity of the multiple metrics on sentence level in the data (LDC2008E43).

Metric	**Granularity**
BLEU-opt	Letter
NIST-opt	Letter
GTM(e=1)	Dependency
TER	Letter
PER	Lexicon
WER	Dependency
ROUGE-opt	Letter

Table 2 The optimal calculation granularity of the multiple metrics

The most remarkable aspect is that not all the best metrics are based on the "lexicon" calculation granularities, such as the "letter" and "dependency". In other words, the granularities-shifted string-based metrics are promising to contribute to the automatic evaluation of translation quality.

5.2 Correlation with Human Assessments of String-based Metrics with Multiple Granularities Based on SVM Frame

We firstly train the SVM rank and regression models on LDC2008E43 using all the features ($21+11 \times 4$ species), without any selection. Secondly, the other two SVM rank and regression models are trained on the same data using the feature set via feature selection, which are described in Table 3. We have four string-based evaluation metrics with multiple granularities on rank and regression SVM frame "Rank_All, Regression_All, Rank_Select and Regression_Select". Then we apply the four metrics to evaluate the sentences of the test data (LDC2006T04 and LDC2003T17). The results of Spearman correlation with human assessments is summarized in Table 3. For comparison, the results from some state-of-art metrics (Papineni et al., 2002; Doddington,

2002; Melamed et al., 2003; Banerjee and Lavie, 2005; Snover et al., 2006; Liu and Gildea, 2005) and two machine learning methods (Albrecht and Hwa, 2007; Ding Liu and Gildea, 2007) are also included in Table 3. Of the two machine learning methods, both trained on the data LDC2006T04. The "Albrecht, 2007" score reported a result of Spearman correlation with human assessments on the data LDC2003T17 using 53 features, while the "Ding Liu, 2007" score reported that under five-fold cross validation on the data LDC2006T04 using 31 features.

	Feature number	**LDC 2003 T17**	**LDC 2006 T04**
Rank_All	65	0.323	0.495
Regression_All	65	**0.345**	0.507
Rank_Select	16	0.338	0.491
Regression_Select	8	0.341	**0.510**
Albrecht, 2007	53	0.309	--
Ding Liu, 2007	31	--	0.369
BLEU-opt[2]	--	0.301	0.453
NIST-opt	--	0.219	0.417
GTM(e=1)	--	0.270	0.375
METEOR[3]	--	0.277	0.463
TER	--	-0.250	-0.302
STM-opt	--	0.205	0.226
HWCM-opt	--	0.304	0.377

Table 3: Comparison of Spearman correlations with human assessments of our proposed metrics and some start-of-art metrics and two machine learning methods
"-opt" stands for the optimum values of the parameters on the metrics

Table 3 shows that the string-based meta-evaluation metrics with multiple granularities based on SVM frame gains the much higher Spearman correlation than other start-of-art metrics on the two test data and, furthermore, our proposed metrics also are higher than the machine learning metrics (Albrecht and Hwa, 2007; Ding Liu and Gildea, 2007).

The underlining is that our proposed metrics are more robust than the aforementioned two machine learning metrics. As shown in Table 1 the heterogeneity between the training and test data in our method is much more significant than that of the other two machine learning based methods.

In addition, the "Regression_Select" metric using only 8 features can achieve a high correlation rate which is close to the metric proposed in "Albrecht, 2007" using 53 features, "Ding Liu, 2007" using 31 features, "Regression_All" and "Rank_All" metrics using 65 features and "Rank_Select" metric using 16 features. What is more, "Regression_Select" metric is better than "Albrecht, 2007", and slightly lower than "Regression_All" on the data LDC2003T17; and better than both "Regression_All" and "Rank_All" metrics on the data LDC2006T04. That confirms that a small cardinal of feature set can also result in a metric having a high correlation with human assessments, since some of the features represent the redundant information in different forms. Eliminating the redundant information is benefit to reduce complexity of the parameter searching and thus improve the metrics performance based on SVM models. Meanwhile, fewer features can relieve the language dependency of the machine learning metrics. At last, our experimental results show that regression models perform better than rank models in the string-based metrics with multiple granularities based on SVM frame, since "Regression_Select" and "Regression_All" achieve higher correlations with human assessments than the others.

5.3 Reliability of Feature Selection

The motivation of feature selection is keeping the validity of the feature set and alleviating the language dependency. We also look forward to the higher Spearman correlation on the test data with a small and proper feature set.

We use SVM-Light (Joachims, 1999) to train our learning models using NIST Open MT 2006 evaluation data (LDC2008E43), and test on the two sets of data, NIST's 2002 and 2003 Chinese MT evaluations. All the data are described in Table 1. To avoid the bias in the distributions of the two judges' assessments in NIST's 2002 and 2003 Chinese MT evaluations, we normalize the scores following (Blatz et al., 2003).

[2] The result is computed by mteval11b.pl.
[3] The result is computed by meteor-v0.7.

We trace the process of the feature selection. The selected feature set of the metric based on SVM rank includes 16 features and that of the metric based on SVM regression includes 8 features. The selected features are listed in Table 4. The values in Table 4 are absolute Spearman correlations with human assessments of each single feature score. The prefixes "C_", "D_", "L_", "P_", and "W_" represent "Constitute", "Dependency", "Letter", "Pos" and "Lexicon" respectively.

Rank	spearman	Regression	spearman
C_PER	.331	C_PER	.331
C_ROUGE-W	.562	C_ROUGE-W	.562
D_NIST9	.479	D_NIST9	.479
D_ROUGE-W	.679	D_ROUGE-L	.667
L_BLEU6	.702	L_BLEU6	.702
L_NIST9	.691	L_NIST9	.691
L_ROUGE-W	.634	L_ROUGE-W	.634
P_PER	.370	P_ROUGE-W	.683
P_ROUGE-W	.616		
W_BLEU1_ind	.551		
W_BLEU2	.659		
W_GTM	.360		
W_METEOR	.693		
W_NIST5	.468		
W_ROUGE1	.642		
W_ROUGE-W	.683		

Table 4: Feature sets of SVM rank and regression

Table 4 shows that 8 features are selected from 65 features in the process of feature selection based on SVM regression while 16 features based on SVM rank. Fewer features based on SVM regression are selected than SVM rank. Only one feature in feature set based on SVM regression does not occur in that based on SVM rank. The reason is that there are more complementary advantages between the common selected features.

Next, we will verify the reliability of our feature selection algorithm. Figure 1 and Figure 2 show the Spearman correlation values between our SVM-based metrics (regression and rank) and the human assessments on both training data (LDC2008E43) and test data (LDC2006T04 and LDC2003T17).

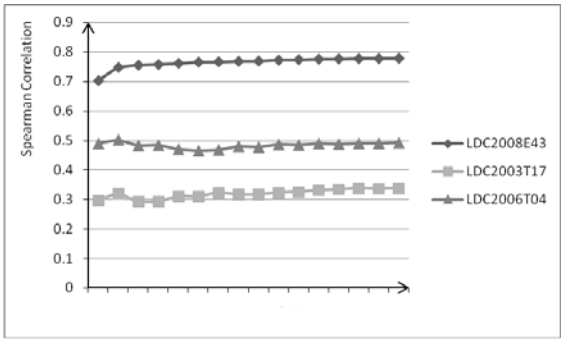


Figure 1: The Spearman correlation values between our **SVM rank** metrics and the human assessments on both training data and test data with the extension of the feature sets

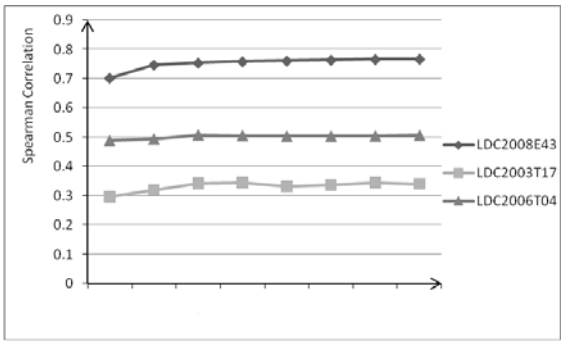


Figure 2: The Spearman correlation values between our **SVM regression** metrics and the human assessments on both training data and test data with the extension of the feature sets

From Figure 1 and Figure 2, with the extension of the feature sets, we can find that the tendency of correlation obtained by each metric based on SVM rank or regression roughly the same on both the training data and test data. Therefore, the two feature sets of SVM rank and regression models are reliable.

6 Conclusion

In this paper we propose an integrated platform for automatic MT evaluation by improving the string based metrics with multiple granularities. Our proposed metrics construct a novel integrated platform for automatic MT evaluation based on multiple features. Our key contribution consists of two parts: i) we suggest a strategy of changing the various complex features into plain string form. According to the strategy, the automatic MT evaluation frame are

much more clarified, and the computation of the similarity is much more simple, since the various linguistic features may express in the uniform strings with multiple calculation granularities. The new features have the same form and are dimensionally homogeneous; therefore, the consistency of the features is enhanced strongly. ii) We integrate the features with machine learning and proposed an effective approach of feature selection. As a result, we can use fewer features but obtain the better performance.

In this framework, on the one hand, string-based metrics with multiple granularities may introduce more potential features into automatic evaluation, with no necessarily of new similarity measuring method, compared with the other metrics. On the other hand, we succeed in finding a finer and small feature set among the combinations of plentiful features, keeping or improving the performance. Finally, we proposed a simple, effective and robust string-based automatic MT evaluation metric with multiple granularities.

Our proposed metrics improve the flexibility and performance of the metrics based on the multiple features; however, it still has some drawbacks: i) some potential features are not yet considered, e.g. the semantic roles; and ii) the loss of information exists in the process of changing linguistic information into plain strings. For example, the dependency label in the calculation granularity "Dependency" is lost when changing information into string form. Though the final results obtain the better performance than the other linguistic metrics, the performance is promising to be further improved if the loss of information can be properly dealt with.

Acknowledgement

This work is supported by Natural Science foundation China (Grant No.60773066 & 60736014) and National Hi-tech Program (Project No.2006AA010108), and the Natural Scientific Reserach Innovation Foundation in Harbin Institute of Technology (Grant No. HIT.NSFIR.20009070).

References

Albrecht S. Joshua and Rebecca Hwa. 2007. *A Reexamination of Machine Learning Approaches for Sentence-Level MT Evaluation*. In Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, pages 880-887.

Amigó Enrique, Julio Gonzalo, Anselmo Pènas, and Felisa Verdejo. 2005. *QARLA: a Framework for the Evaluation of Automatic Summarization.* In Proceedings of the 43th Annual Meeting of the Association for Computational Linguistics.

Amigó Enrique, Jesús Giménez, Julio Gonzalo, Felisa Verdejo. 2009. *The Contribution of Linguistic Features to Automatic Machine Translation Evaluation.* In proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP.

Amigó Enrique, Jesús Giménez, Julio Gonzalo, and Lluís Màrquez. 2006. *MT Evaluation: Human-Like vs. Human Acceptable.* In Proceedings of the Joint 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistic, pages 17–24.

Banerjee Satanjeev and Alon Lavie. 2005. *METEOR: An automatic metric for MT evaluation with improved correlation with human judgments.* In Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures.

Blatz John, Erin Fitzgerald, George Foster, Simona Gandrabur, Cyril Goutte, Alex Kulesza, Alberto Sanchis, and Nicola Ueffing. 2003. *Confidence estimation for machine translation.* In Technical Report Natural Language Engineering Workshop Final Report, pages 97-100.

Callison-Burch Chris, Miles Osborne, and Philipp Koehn. 2006. *Re-evaluating the Role of BLEU in Machine Translation Research.* In Proceedings of 11th Conference of the European Chapter of the Association for Computational Linguistics

Chan S. Yee and Hwee T. Ng. 2008. *MAXSIM: A maximum similarity metric for machine translation evaluation.* In Proceedings of ACL-08: HLT, pages 55–62.

Doddington George. 2002. *Automatic Evaluation of Machine Translation Quality Using N-gram Co-Occurrence Statistics.* In Proceedings of the 2nd International Conference on Human Language Technology, pages 138–145.

Drucker Harris, Chris J. C. Burges, Linda Kaufman, Alex Smola, Vladimir Vapnik. 1996. *Support vector regression machines.* In NIPS.

Giménez Jesús and Lluís Màrquez. 2007. *Linguistic Features for Automatic Evaluation of Heterogeneous MT Systems.* In Proceedings of the ACL Workshop on Statistical Machine Translation.

Giménez Jesús and Lluís Màrquez. 2008a. *Heterogeneous Automatic MT Evaluation Through Non-Parametric Metric Combinations.* In Proceedings of IJCNLP, pages 319–326.

Giménez Jesús and Lluís Màrquez. 2008b. *On the Robustness of Linguistic Features for Automatic MT Evaluation.*

Joachims Thorsten. 2002. *Optimizing search engines using clickthrough data.* In KDD.

Klein Dan and Christopher D. Manning. 2003a. *Fast Exact Inference with a Factored Model for Natural Language Parsing.* In Advances in Neural Information Processing Systems 15, pp. 3-10.

Klein Dan and Christopher D. Manning. 2003b. *Accurate Unlexicalized Parsing.* Proceedings of the 41st Meeting of the Association for Computational Linguistics, pp. 423-430.

Le Audrey and Mark Przybocki. 2005. *NIST 2005 machine translation evaluation official results.* In Official release of automatic evaluation scores for all submission.

Lin Chin-Yew and Franz Josef Och. 2004. *Automatic Evaluation of Machine Translation Quality Using Longest Common Subsequence and Skip-Bigram Statistics.* Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics, pp. 605-612.

Liu Ding and Daniel Gildea. 2005. *Syntactic Features for Evaluation of Machine Translation.* In Proceedings of ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for MT and/or Summarization, pages 25–32.

Liu Ding and Daniel Gildea. 2007. *Source Language Features and Maximum Correlation Training for Machine Translation Evaluation.* In proceedings of NAACL HLT 2007, pages 41–48

Mehay Dennis and Chris Brew. 2007. *BLEUATRE: Flattening Syntactic Dependencies for MT Evaluation.* In Proceedings of the 11th Conference on Theoretical and Methodological Issues in Machine Translation.

Melamed Dan I., Ryan Green, and Joseph P. Turian. 2003. *Precision and Recall of Machine Translation.* In Proceedings of the Joint Conference on Human Language Technology and the North American Chapter of the Association for Computational Linguistics.

Nießen Sonja, Franz Josef Och, Gregor Leusch, and Hermann Ney. 2000. *An Evaluation Tool for Machine Translation: Fast Evaluation for MT Research.* In Proceedings of the 2nd International Conference on Language Resources and Evaluation .

Owczarzak Karolina, Declan Groves, Josef Van Genabith, and Andy Way. 2006. *Contextual Bitext- Derived Paraphrases in Automatic MT Evaluation.* In Proceedings of the 7th Conference of the Association for Machine Translation in the Americas, pages 148–155.

Owczarzak Karolina, Josef van Genabith, and Andy Way. 2007. *Labelled Dependencies in Machine Translation Evaluation.* In Proceedings of the ACL Workshop on Statistical Machine Translation, pages 104–111.

Papineni Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. *BLEU: a Method for Automatic Evaluation of Machine Translation.* In Proceedings of 40th Annual Meeting of the Association for Computational Linguistics.

Popović Maja and Hermann Ney. 2007. *Word Error Rates: Decomposition over POS classes and Applications for Error Analysis.* In Proceedings of the Second Workshop on Statistical Machine Translation, pages 48–55.

Popović Maja and Hermann Ney. 2009. *Syntax-oriented evaluation measures for machine translation output.* In Proceedings of the 4th EACL Workshop on Statistical Machine Translation, pages 29–32.

Snover Matthew, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. *A study of translation edit rate with targeted human annotation.* In Proceedings of AMTA, pages 223–231.

Tillmann Christoph, Stefan Vogel, Hermann Ney, A. Zubiaga, and H. Sawaf. 1997. *Accelerated DP based Search for Statistical Translation.* In Proceedings of European Conference on Speech Communication and Technology.

Automatic Treebank Conversion via Informed Decoding

Muhua Zhu
Natural Language Processing Lab.
Northeastern University
zhumuhua@gmail.com

Jingbo Zhu
Natural Language Processing Lab.
Northeastern University
zhujingbo@mail.neu.edu.cn

Abstract

In this paper, we focus on the challenge of automatically converting a constituency treebank (source treebank) to fit the standard of another constituency treebank (target treebank). We formalize the conversion problem as an *informed decoding* procedure: information from original annotations in a source treebank is incorporated into the decoding phase of a parser trained on a target treebank during the parser assigning parse trees to sentences in the source treebank. Experiments on two Chinese treebanks show significant improvements in conversion accuracy over baseline systems, especially when training data used for building the parser is small in size.

1 Introduction

Recent years have seen extensive applications of machine learning methods to natural language processing problems. Typically, increase in the scale of training data boosts the performance of machine learning methods, which in turn enhances the quality of learning-based NLP systems (Banko and Brill, 2001). However, annotating data by human is time consuming and labor intensive. For this reason, human-annotated corpora are considered as the most valuable resource for NLP.

In practice, there often exist more than one corpus for the same NLP tasks. For example, for constituent syntactic parsing (Collins, 1999; Charniak, 2000; Petrov et al., 2006) for Chinese, in addition to the most popular treebank Chinese Treebank (CTB) (Xue et al., 2002), there are also other treebanks such as Tsinghua Chinese Treebank (TCT) (Zhou, 1996). For the purpose of full use of readily available human annotations for the same tasks, it is significant if such corpora can be used jointly. Such attempt is especially significant for some languages that have limited size of labeled data. At first sight, a direct combination of multiple corpora is a way to this end. However, corpora created for the same NLP tasks are generally built by different organizations. Thus such corpora often follow different annotation standards and/or even different linguistic theories. We take CTB and TCT as a case study. Although both CTB and TCT are Chomskian-style treebanks, they have annotation divergences in at least two dimensions: a) CTB and TCT have dramatically different tag sets, including parts-of-speech and grammar labels, and the tags cannot be mapped one to one; b) CTB and TCT have distinct hierarchical structures. For example, the Chinese words "中国 (Chinese) 传统 (traditional) 文化 (culture)" are grouped as a flat noun phrase according to the CTB standard (right side in Fig. 1), but in TCT, the last two words are instead grouped together beforehand (left side in Fig. 1). The differences cause such treebanks of different annotation standard to be generally used independently.

In this paper, we focus on unifying multiple constituency treebanks of distinct annotation standards through treebank conversion. The task of treebank conversion is defined to be conversion of annotations in one treebank (source treebank) to

Coling 2008: Poster Volume, pages 1541–1549,
Beijing, August 2010

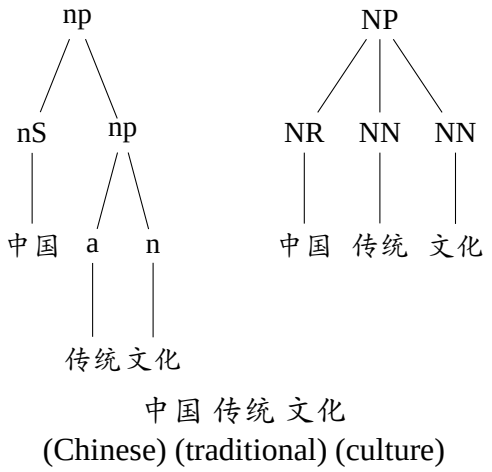


Figure 1: Example tree fragments with TCT (left) and CTB (right) annotations

fit the standard of another treebank (target treebank). To this end, we propose a language independent approach called *informed decoding* [1], in which a parser trained on a target treebank automatically assigns new parse trees to sentences in a source treebank with the aid of information derived from annotations in the source treebank. We conduct experiments on two open Chinese treebanks [2]: CTB and TCT. Experimental results show that our approach achieves significant improvements over baseline systems, especially when training data used for building the parser is small in size.

The rest of the paper is structured as follows. In Section 2 we describe previous work on treebank conversion. In Section 3, we describe in detail the informed decoding approach. Section 4 presents experimental results which demonstrate the effectiveness of our approach. Finally, Section 5 concludes our work.

2 Related Work

Previous work on treebank conversion can be grouped into two categories according to whether grammar formalisms of treebanks are identical. One type focuses on converting treebanks of different grammar formalisms. Collins et al. (1999) addressed constituent syntactic parsing on Czech using a treebank converted from a Prague dependency treebank, where conversion rules derived from head-dependent pairs and heuristic rules are applied. Xia and Palmer (2001) compared three algorithms for conversion from dependency structures to phrase structures. The algorithms expanded each node in input dependency structures into a projection chain, and labeled the newly inserted node with syntactic categories. The three algorithms differ only in heuristics adopted to build projection chains. Xia et al. (2008) automatically extracted conversion rules from a target treebank and proposed strategies to handle the case when more than one conversion rule are applicable. Instead of using conversion rules, Niu et al. (2009) proposed to convert a dependency treebank to a constituency one by using a parser trained on a constituency treebank to generate k-best lists for sentences in the dependency treebank. Optimal conversion results are selected from the k-best lists. There also exists work in the reverse direction: from a constituency treebank to a dependency treebank (Nivre, 2006; Johansson and Nugues, 2007).

Relatively few efforts have been put on conversion between treebanks that have the same grammar formalisms but follow different annotation standards. Wang et al. (1994) applied a similar framework as in (Niu et al., 2009) to convert from a simple constituency treebank to a more informative one. The basic idea is to apply a parser built on a target treebank to generate k-best lists for sentences in the source treebank. Then, a matching metric is defined on the number of identical bracketing spans between two trees. Such a function computes a score for each parse tree in a k-best list and its corresponding parse tree in the source treebank. Finally, the parse tree with the highest score in a k-best list is selected to be the conversion result. The difference between our work and (Wang et al., 1994) is that, instead of using trees from the source treebank to select parse trees from k-best lists, we propose to use such trees to guide the decoding phase of the parser built on the target treebank. Making use of the source treebank in such a novel way is believed to be the major contribution of our work.

[1] The terminology *decoding* is referred to the parsing phase of a parser.

[2] Note that although we use Chinese treebanks, our approach is language independent.

3 Treebank Conversion via Informed Decoding

The task of treebank conversion is defined to convert parse trees in a source treebank to fit the standard of a target treebank. In the informed decoding approach, treebank conversion proceeds in two steps: 1) build a parser on a target treebank; 2) apply the parser to decode sentences in a source treebank with the aid of information derived from the source treebank. For convenience, parse trees in a source treebank are referred to as *source trees* and corresponding, trees from a target treebank are referred to as *target trees*. Moreover, a parser built on a target treebank is referred to as *target parser*. In the following sections, we first describe motivation of our work and then present details of the informed decoding approach.

3.1 Motivation

We use the example in Fig. 2 to illustrate why original annotations in a source treebank can help in treebank conversion. The figure depicts three tree fragments for the Chinese words 发 *(pay)* 了 *(already)* 一 *(one)* 天 *(day)* 的 *(of)* 工资*(salary)*, among which Fig. 2(a) and Fig. 2(b) are tree fragments of the CTB standard and Fig. 2(c) is a tree fragment of the TCT standard. From the figure, we can see that these Chinese words actually have (at least) two plausible interpretations of the meaning. In Fig. 2(a), the words mean *pay salary for one-day work* while in Fig. 2(b), the words mean *spend one day on paying salary*. If Fig. 2(c) is a source tree to be converted into the CTB standard, then Fig. 2(b) will be rejected since it conflicts with Fig. 2(c) with respect to tree structures. Note that structures reflect underlying sentence meaning. On the other hand, although Fig. 2(a) also has (minor) differences in tree structures from Fig. 2(c), it is preferred as the conversion result[3]. From the example we can get inspired by the observation that original annotations in a source treebank are informative and necessary to converting parse trees in the source treebank.

In general, conversion like that from Fig. 2(c) to Fig. 2(a) requires sentence-specific conversion rules which are difficult to obtain in practice. In order to make use of information provided by original annotations in a source treebank, Wang et al. (1994) proposed a selecting-from-k-best approach where source trees are used to select one "optimal" parse tree from each k-best list generated by a target parser. In this paper, we instead incorporate information of original annotations into the parsing phase. The underlying motivation is two-fold:

- The decoding phase of a parser is essentially a search process. Due to the extreme magnitude of searching space, pruning of search paths is practically necessary. If reliable information is provided to guide the pruning of search paths, more efficient parsing and better results are expected.

- Selecting-from-k-best works on the basis of k-best lists. Unfortunately, we often see very few variations in k-best lists. For example, 50-best trees present only 5 to 6 variations (Huang, 2008). The lack of diversities in k-best lists makes information from the source treebank less effective in selecting parse trees. By contrast, incorporating such information into decoding makes the information affect the whole parse forest.

3.2 Formalization of Information from Source Treebank

In this paper, information from a source treebank translates into two strategies which help a target parser to prune illegal partial parse trees and to rank legal partial parse trees higher. Following are the two strategies:

- Pruning strategy: despite distinctions existing between annotation standards of a source treebank and a target treebank, a source treebank indeed provides treebank conversion with indicative information on bracketing structures and grammar labels. So when a partial parse tree is generated, it should be examined against the corresponding source tree. Unless the partial parse tree does *not conflict* with any constituent in the source tree, it should be pruned out.

[3]Note that we don't deny existence of annotation distinctions between the treebanks, but we aim to make use of what they both agree on. We assume that consensus is the majority.

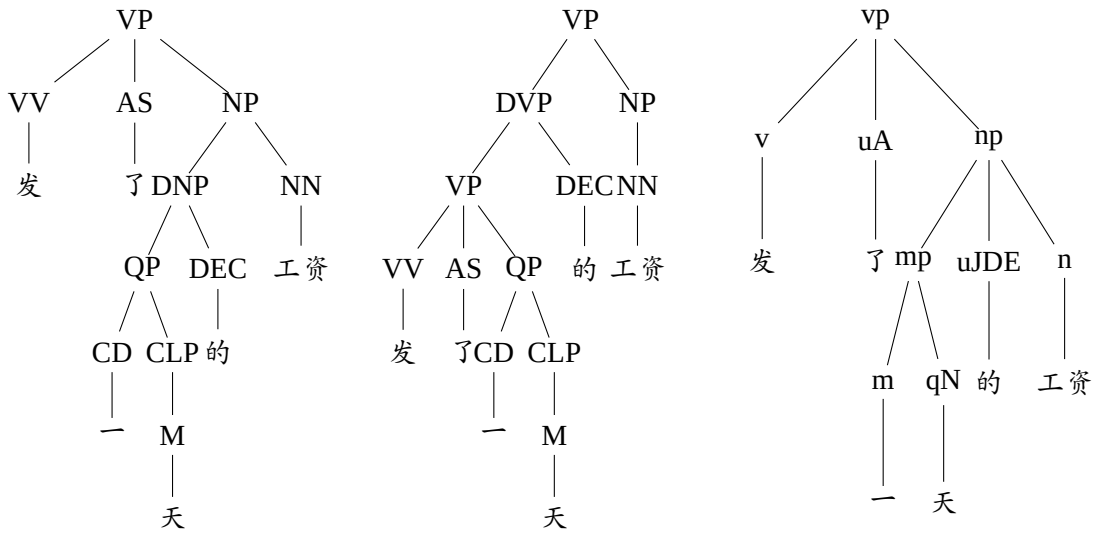


Figure 2: tree fragments of words 发 了 一 天 的 工资: (a) and (b) show two plausible tree fragments of the words using the CTB standard; (c) shows a tree fragment of the TCT standard which has the same interpretation as (a).

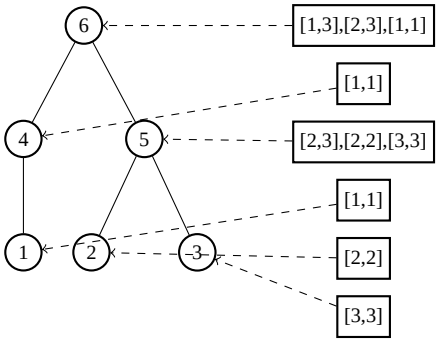


Figure 3: Constituent set of a synthetic parse tree

- Rescoring strategy: in practice, decoding is often a local optimal search process. In some cases even if a correct parse tree exits in the parse forest, parsers may fail to rank it to the top position. Rescoring strategy is used to increase scores for partial parse trees which are confidently thought to be valid.

3.2.1 Pruning Strategy

The pruning strategy used in this paper is based on the concept of *conflict* which is defined in two dimensions: structures and grammar labels. Since a tree structure can be equivalently represented as its span (interval of word indices) set, we can check whether two trees conflict by checking their spans. See Fig. 3 for an illustration of spans of a tree. Following are criteria determining whether two trees conflict in their structures.

- If one node in tree A is raised to be a child of the node's grandfather in tree B, and the grandfather has more than two children, then tree A and tree B conflict in structures.
- If tree A has a span $[a, b]$ and tree B has a span $[m, k]$ and these two spans satisfy the condition of either $a < m \leq b < k$ or $m < a \leq k < b$, then tree A and B conflict in structures.

Fig. 4 illustrates criteria mentioned above, where Fig. 4(a) is compatible (not conflict) with Fig. 4(b) although they have different structures. But Fig. 4(a) conflicts with Fig. 4(c) (according to criterion 1; node 3 is raised) and (d) (according to criterion 2).

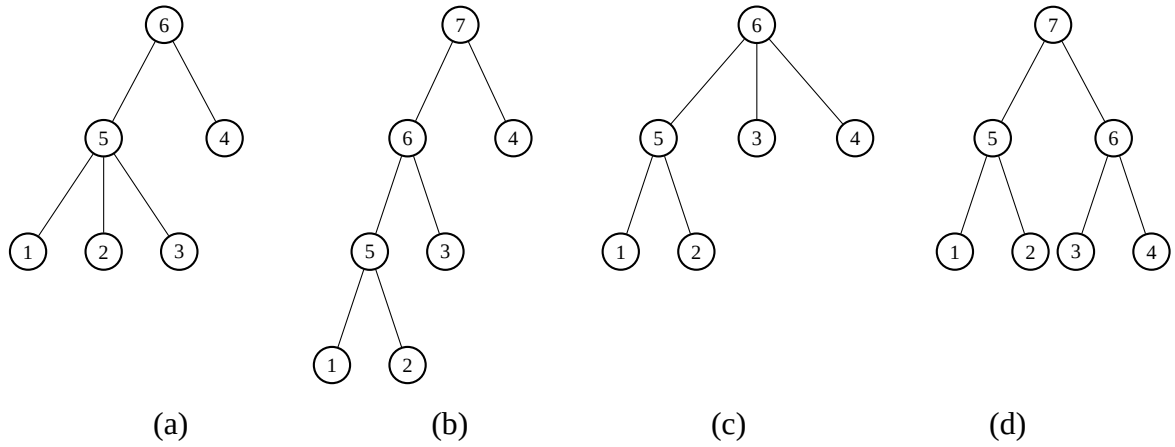


Figure 4: Illustrating example of the concept of *conflict*: (a) and (b) are compatible (not conflict); (a) conflicts with (c) (condition 1) and (d) (condition 2)

For the dimension of grammar labels, we manually construct a mapping between label sets (POS tags excluded) of source and target treebanks. Such a mapping is frequently a many-to-many mapping. Two labels are said to be conflicting if they are from different label sets and they cannot be mapped.

By combining these two strategies, two parse trees (of different standards) which yield the same sentence are said to be conflicting if they conflict in both structures and labels. Note that we describe pruning strategy for the case of two parse trees. In informed decoding process, this strategy is actually applied to every partial parse tree generated during decoding.

3.2.2 Rescoring Strategy

As mentioned above, despite that the pruning strategy helps in improving conversion accuracy, we are faced with the problem of how to rank valid parse trees higher in a parse forest. To solve the problem, we adjust the scores of those partial parse trees that are considered to be confidently "good". The criteria which is used to judge "goodness" of a partial parse are listed as follows:

- The partial parse tree can find in the source tree a constituent that has the same structure as it.
- When the first criterion is satisfied, grammar categories of this partial parse should not conflict with the grammar categories of its counterpart.

In practice, we use a parameter λ to adjust the score.

$$P_{new}(e) = \lambda * P(e) \quad (1)$$

Here e represents any partial tree that is rescored, and $P(e)$ and $P_{new}(e)$ refer to original and new scores, respectively.

3.3 Parsing Model

Theoretically all parsing models are applicable in informed decoding, but we prefer to adopt a CKY-style parser for two reasons: CKY style parsers are dynamically bottom-up and always have edges (or parsing items) belonging to the same span stacked together in the same chart[4] cell. The property of CKY-style parsers being dynamically bottom-up can make the pruning strategy efficient by avoiding rechecking subtrees that have already been checked. The property of stacking edges in the same chart cell makes CKY-style parsers easily portable to the situaiton of informed decoding. In this paper, Collins parser (Collins, 1999) is used. Algorithm 1 presents the extended version of the decoding algorithm used in Collins parser. What the algorithm needs to do is to generate edges for each span. And before edges are allowed to enter the chart, pruning conditions

[4]Data structure used to store paring items that are not pruned

Algorithm 1 CKY-style decoding
Argument: a parsing decoder
a sentence to be parsed and corresponding source tree

Begin
Steps:
1. initialization steps
2. for span from 2 to sentence_length **do**
for start from 1 to (sentence_length-span+1) **do**
end := (start + span - 1)
for each edge e for span [start, end] **do**
generate(e**, start, end)**
prune(e**, start, end)**
rescore(e**, start, end)**
add_edge(e**, start, end)**
End

Subroutine:
generate: generates an edge which belongs to the span [start, end].
prune: apply *pruning strategy* to check whether the edge should be pruned.
rescore: apply *rescoring strategy* to weight the edge.
add_edge: add the edge into *chart*.

should be checked in *prune* subroutine and rescoring should be conducted in *rescore* subroutine with respect to the corresponding source tree.

4 Experiments

4.1 Experimental Setup

In this paper, we conduct two groups of experiments in order to evaluate 1) treebank conversion accuracy and 2) how much newly generated data can boost syntactic parsing accuracy. For the experiments of treebank conversion, Penn Chinese Treebank (CTB) 5.1 is used as the target treebank. That is, the CTB standard is the one we are interested in. Following the conventional data splitting of CTB5.1, articles 001-270 and 400-1151 (18,100 sentences, 493,869 words) are used for training, articles 271-300 (348 sentences, 8,008 words) are used as test data, and articles 301-325 (352 sentences, 6,821 words) are used as development data [5]. Moreover, in order to directly evaluate conversion accuracy, we randomly sampled 150 sentences from the CTB test set and have three annotators manually label sentences of these parse trees according to the standard of Tsinghua Chinese Treebank (TCT). Thus each of the 150 sentences has two parse trees, following the CTB and TCT standard, respectively. For convenience of reference, the set of 150 parse trees of the CTB standard is referred to as *Sample-CTB* and its counterpart which follows the TCT standard is referred to as *Sample-TCT*. In such setting, the experiments of treebank conversion is designed to use the informed decoding approach to convert Sample-TCT to the standard of CTB and conversion results are evaluated with respect to Sample-CTB. The CTB training data (or portion of it) is used as target training data on which parsers are trained for conversion.

For the experiments of syntactic parsing, the TCT corpus is used as the source treebank. The TCT corpus contains 27,268 sentences and 587,298 words, which are collected from the literature and newswire domains. In this group of experiments, the CTB training data is again used as target training data and the whole TCT corpus is converted using the informed decoding approach. The newly-gained parse trees are used as additional training data for syntactic parsing on the CTB test data. One thing worth noting in the experiments is that, using Collins parser to convert the TCT corpus requires Part-of-Speech tags of the CTB standard be assigned to sentences in TCT ahead of conversion being conducted. To this end, instead of using POS taggers, we use the *label correspondence learning* method described in (Zhu and Zhu, 2009) in order to get high POS tagging accuracy.

For all the experiments in this paper, *bracketing F1* is used as the performance metric, provided by the EVALB program [6]. λ in Eq.1 is set to 3.0 since it provides best conversion results in our experiments.

4.2 Experiments on Conversion

The setup of conversion experiments is described above. In the experiments, we use two representative baseline systems. One, named *directly parsing (DP)* converts Sample-TCT by directly parsing using Collins parser which is trained on target training data, and the other is the method proposed in (Wang et al., 1994) (hereafter referred to as *Wang94*). For the latter baseline, we use Berkeley parser (Petrov et al., 2006) instead of

[5] Development set is not used in this paper.

[6] http://nlp.cs.nyu.edu/evalb

Ratio	20%	40%	60%	80%	100%
DP	73.19	75.21	79.43	80.64	81.40
Wang94	75.00	76.82	78.08	81.50	82.47
This paper	**82.71**	**83.00**	**83.37**	**84.80**	**84.34**

Table 1: Conversion accuracy with varying size of target training data

Span Length	2	4	6	8	10
Wang94	82.45	**83.97**	**80.72**	**77.83**	**71.72**
This paper	**83.72**	82.95	79.84	77.27	70.67
Span Length	12	14	16	18	20
Wang94	**75.29**	68.00	77.27	70.83	76.66
This paper	71.79	**75.00**	**86.27**	**80.00**	**80.00**

Table 2: Conversion accuracy on different span lengths

Category	ADJP	VCD	CP	DNP	ADVP
Wang94	79.62	57.14	65.43	84.76	91.73
This paper	**88.00**	**66.67**	**71.60**	**88.31**	**93.44**

Table 3: Conversion results with respect to different grammar categories

Collins parser. The reason is that we want to build a strong baseline since Berkeley parser is able to generate better k-best lists than Collins parser does (Zhang et al., 2009). In detail, Wang94 proceeds in two steps: 1) use Berkeley parser to generate k-best lists for sentences in Sample-TCT; 2) select a parse tree from each k-best list with respect to original annotations in Sample-TCT. Here we set k to 50. Table 1 reports F1 scores of the baseline systems and our informed decoding approach with varying size of target training data. The first row of the table represents fractions of the CTB training data which are used as target training data. For example, 40% means 7,240 parse trees (of 18,100) in the CTB training data are used. To relieve the effect of ordering, we randomly shuffled parse trees in the CTB training data.

From the table, we can see that our approach performs significantly better than DP and Wang94. In detail, when 100% CTB training data is used as target training data, 2.95% absolute improvement is achieved. When the size of target training data decreases, absolute improvements of our approach over baseline systems are further enlarged. More interestingly, decreasing in target training data only results in marginal decrement in conversion accuracy of our approach. This is of significant importance in the situation where target treebank is small in size.

In order to evaluate the accuracy of conversion methods on different span lengths, we compare the results of Wang94 and informed decoding produced by using 100% CTB training data. Table 2 shows the statistics.

From the results we can see that our approach performs significantly better on long spans and achieves marginally lower accuracy on small ones. But notice that the informed decoding approach is implemented on the base of Collins parser and that Wang94 works on the basis of Berkeley parser. Taking the performance gap of Collins parser and Berkeley parser, we actually can conclude that on small spans, our approach is able to achieve results comparable with or even better than Wang94. We can also infer from the observation that our approach can outperform Wang94 when converting parse trees which yield long sentences.

Another line of analysis is to compare the results of Wang94 and our approach, with respect to different grammar categories. Table 3 lists five grammar categories in which our approach achieves most improvements. For categories *NP* and *VP*, absolute improvements are 1.1% and 1.4% respectively. Take into account large amounts of instances of *NP* and *VP*, the improvements are also quite significant.

4.3 Experiments on Parsing

Before doing the experiments of parsing, we first converted the whole TCT corpus using 100% CTB training data as target training data. Using the newly-gained data only as training data for Collins parser, we can get F1 score 75.4% on the CTB test data. We can see that the score is much lower than the accuracy achieved by using the CTB training data (75.4% vs. 82.04%). Possible reasons that result in lower accuracy includes: 1) divergences in word segmentation standards between TCT and CTB; 2) divergences of domains of TCT and CTB; 3) conversions errors in newly-gained data. Although the newly-gained

data cannot replace the CTB training data thoroughly, we would like to use it as additional training data besides the CTB training data. Following experiments aim to examine effectiveness of the newly-gained data when used as additional training data.

In the first parsing experiment, the TCT corpus is converted using portions of the CTB training data. As in the conversion experiments, parse trees in the CTB training data are randomly ordered before splitting of the training set. For each portion, newly-gained data together with the portion of the CTB training data are used to train a new parser. Evaluation results on the CTB test data are presented in Table 4.

Ratio	20%	40%	60%	80%	100%
Collins	75.74	77.65	79.43	81.22	82.04
Collins+	**78.86**	**79.52**	**80.06**	**81.77**	**82.38**

Table 4: Parsing accuracy with new data added in

Here in Table 4, the first row represents ratios of parse trees from the CTB training data. For example, 40% means the first 40% parse trees in the CTB training data are used. The *Collins* row represents the results of only using portions of the CTB training data, and the *Collins+* row contains the results achieved with enlarged training data. From the results, we find that new data indeed provides complementary information to the CTB training data, especially when the training data is small in size. But benefits of Collins parser gained from additional training data level out with the increment of the training data size. Actually if techniques like corpus weighting (Niu et al., 2009) are applied to weight differently training data and the additional data, higher parsing accuracy is reasonably expected.

Another obversion from Table 4 is that the parser trained on 40% CTB training data plus additional training data achieves higher accuracy than using 60% CTB training data. We incrementally add labeled training data and automatic training data respectively to 40% CTB training data. The purpose of this experiment is to see the magnitude of automatic training data which can achieve the same effect as labeled training data does. The results are depicted in Table 5.

# of Added Data	2k	4k	6k	8k
Labeled Data	78.51	79.52	80.01	81.37
Auto Data	78.23	79.11	79.85	79.67

Table 5: Parsing accuracy with new data added in

From the results we see that accuracy gaps between using labeled data and using automatic data get large with the increment of added data. One possible reason is that more noise is taken when more data is added. This observation further verifies that refining techniques like corpus weighting are necessary for using automatically-gained data.

5 Conclusions

In this paper we proposed an approach called informed decoding for the task of conversion between treebanks which have different annotation standards. Experiments which evaluate conversion accuracy directly showed that our approach significantly outperform baseline systems. More interestingly we found that the size of target training data have limited effect on the conversion accuracy of our approach. This is extremely important for languages which lack enough treebanks in whose standards we are interested.

We also added newly-gained data to target training data to check whether new data can boost parsing results. Experiments showed additional training data provided by treebank conversion could boost parsing accuracy.

References

Banko, Michele and Eric Brill. 2001. *Scaling to very very large corpora for natural language disambiguation*. In Proc. of ACL 2001, pages 26-33.

Charniak, Eugene. 2000. *A Maximum-Entropy-Inspired Parser*. In Proc. of NAACL 2000, pages 132-139.

Collins, Michael. 1999. *Head-driven statistical models for natural language parsing*. Ph.D. thesis, University of Pennsylvania.

Collins, Michael, Lance Ramshaw, Jan Hajic, and Christoph Tillmann. 1999. *A Statistical Parser for Czech*. In Proc. of ACL 1999, pages 505-512.

Charniak, Eugene. 2000. *A maximum-entropy-inspired parser*. In Proc. of NAACL 2000, pages 132-139.

Huang, Liang. 2008. *Forest reranking: Discriminative parsing with non-local features.* In Proc. of ACL 2008, pages 586-594.

Johansson, Richard and Pierre Nugues. 2007. *Extended constituent-to-dependency conversion for English.* In Proc. of NODALIDA 2007, pages 105-112.

Nivre, Joakim. 2006. *Inductive Dependency Parsing.* In Springer, Volume 34.

Niu, Zheng-Yu, Haifeng Wang, Hua Wu. 2009. *Exploiting heterogeneous treebanks for parsing*. In Proc. of ACL 2009, pages 46-54.

Petrov, Slav, Leon Barrett, Romain Thibaux, and Dan Klein. 2006. *Learning accurate, compact, and interpretable tree annotation*. In Proc. of COLING-ACL 2006, pages 433-440.

Xue, Nianwen, Fu dong Chiou, and Martha Palmer. 2002. *Building a large-scale Annotated Chinese corpus*. In Proc. of COLING 2002, pages 1-8.

Wang, Jong-Nae, Jing-Shin Chang, and Keh-Yih Su. 1994. *An automatic treebank conversion algorithm for corpus sharing*. In Proc. of ACL 1994, pages 248-254.

Xia, Fei, Rajesh Bhatt, Owen Rambow, Martha Palmer, and Dipti M. Sharma. 2008. *Towards a Multi-Representational Treebank.* In Proc. of the 7th International Workshop on Treebanks and Linguistic THeories, pages 159-170.

Zhang, Hui, Min Zhang, Chew Lim Tan, and Haizhou Li. 2009. *K-best combination of syntactic parsers*. In Proc. of EMNLP 2009, pages 1552-1560.

Zhou, Qiang. 1996. *Phrase bracketing and annotating on Chinese language corpus. (in Chinese)* Ph.D. thesis, Beijing University.

Zhu, Muhua and Jingbo Zhu. 2009. *Label Correspondence Learning for Part-of-Speech Annotation Transformation*. In Proc. of CIKM 2009, pages 1461-1464.

Imposing Hierarchical Browsing Structures onto Spoken Documents

Xiaodan Zhu & Colin Cherry
Institute for Information Technology
National Research Council Canada
{Xiaodan.Zhu,Colin.Cherry}@nrc-cnrc.gc.ca

Gerald Penn
Department of Computer Science
University of Toronto
gpenn@cs.toronto.edu

Abstract

This paper studies the problem of imposing a known hierarchical structure onto an unstructured spoken document, aiming to help browse such archives. We formulate our solutions within a dynamic-programming-based alignment framework and use minimum error-rate training to combine a number of global and hierarchical constraints. This pragmatic approach is computationally efficient. Results show that it outperforms a baseline that ignores the hierarchical and global features and the improvement is consistent on transcripts with different WERs. Directly imposing such hierarchical structures onto raw speech without using transcripts yields competitive results.

1 Introduction

Though speech has long served as a basic method of human communication, revisiting and browsing speech content had never been a possibility before human can record their own voice. Recent technological advances in recording, compressing, and distributing such archives have led to the consistently increasing availability of spoken content.

Along with this availability comes a demand for better ways to browse such archives, which is inherently more difficult than browsing text. In relying on human beings' ability to browse text, a solution is therefore to reduce the speech browsing problem to a text browsing task through technologies that can automatically convert speech to text, i.e., the automatic speech recognition (ASR). Research along this line has implicitly changed the traditional speaking-for-hearing and writing-for-reading construals: now speech can be *read* through its transcripts, though it was not originally intended for this purpose, which in turn raises a new set of problems.

The efficiency and convenience of reading spoken documents are affected by at least two facts. First, the quality of transcripts can impair browsing efficiency, e.g., as shown in (Stark et al., 2000; Munteanu et al., 2006), though if the goal is only to browse salient excerpts, recognition errors on the extracts can be reduced by considering the confidence scores assigned by ASR (Zechner and Waibel, 2000; Hori and Furui, 2003).

Even if transcription quality is not a problem, browsing transcripts is not straightforward. When intended to be read, written documents are almost always presented as more than uninterrupted strings of text. Consider that for many written documents, e.g., books, indicative structures such as section/subsection headings and tables-of-contents are standard constituents created manually to help readers. Structures of this kind, however, are rarely aligned with spoken documents.

In this paper, we are interested in addressing the second issue: adding hierarchical browsable structures to speech transcripts. We define a hierarchical browsable structure as a set of nested labelled bracketing which, when placed in text, partition the document into labeled segments. Examples include the sequence of numbered section headings in this paper, or the hierarchical slide/bullet structure in the slides of a presentation.

Coling 2008: Poster Volume, pages 1550–1557,
Beijing, August 2010

An ideal solution to this task would directly infer both the hierarchical structure and the labels from unstructured spoken documents. However, this is a very complex task, involving the analysis of not only local but also high-level discourse over large spans of transcribed speech. Specifically for spoken documents, spoken-language characteristics as well as the lack of formality and thematic boundaries in transcripts violate many conditions that a reliable algorithm (Marcu, 2000) relies on and therefore make the task even harder.

In this paper, we aim at a less ambitious but naturally occurring problem: imposing a known hierarchical structure, e.g., presentation slides, onto the corresponding document, e.g., presentation transcripts. Given an ordered, nested set of topic labels, we must place the labels so as to correctly segment the document into appropriate units. Such an alignment would provide a useful tool for presentation browsing, where a user could easily navigate through a presentation by clicking on bullets in the presentation slides. The solution to this task should also provide insights and techniques that will be useful in the harder structure-inference task, where hierarchies and labels are not given.

We present a dynamic-programming-based alignment framework that considers global document features and local hierarchical features. This pragmatic approach is computationally efficient and outperforms a baseline alignment that ignores the hierarchical structure of bullets within slides. We also explore the impact of speech recognition errors on this task. Furthermore, we study the feasibility of directly aligning a structure to raw speech, as opposed to a transcript.

2 Related work

Topic/slide boundary detection The previous work most directly related to ours is research that attempts to find *flat* structures of spoken documents, such as topic and slide boundaries. For example, the work of (Chen and Heng, 2003; Ruddarraju, 2006; Zhu et al., 2008) aims to find slide boundaries in the corresponding lecture transcripts. Malioutov et al. (2007) developed an approach to detecting topic boundaries of lecture recordings by finding repeated acoustic patterns. None of this work, however, has involved hierarchical structures that exist at different levels of a document.

In addition, researchers have also analyzed other multimedia channels, e.g., video (Liu et al., 2002; Wang et al., 2003; Fan et al., 2006), to detect slide transitions. Such approaches, however, are unlikely to find semantic structures that are more detailed than slide transitions, e.g., the bullet hierarchical structures that we are interested in.

Building tables-of-contents on written text A notable effort going further than topic segmentation is the work by Branavan et al. (2007), which aims at the ultimate goal of building tables-of-contents for written texts. However, the authors assumed the availability of the hierarchical structures and the corresponding text spans. Therefore, their problem was restricted to generating titles for each span. Our work here can be thought of as the inverse problem, in which the title of each section is known, but the corresponding segments in the spoken documents are unknown. Once the correspondence is found, an existing hierarchical structure along with its indicative titles is automatically imposed on the speech recordings. Moreover, this paper studies spoken documents instead of written text. We believe it is more attractive not only because of the necessity of browsing spoken content in a more efficient way but also the general absence of helpful browsing structures that are often available in written text, as we have already discussed above.

Rhetoric analysis In general, analyzing discourse structures can provide thematic skeletons (often represented as trees) of a document as well as relationship between the nodes in the trees. Examples include the widely known discourse parsing work by Marcu (2000). However, when the task involves the understanding of high-level discourse, it becomes more challenging than just finding local discourse conveyed on small spans of text; e.g., the latter is more likely to benefit from the presence of discourse markers. Specifically for spoken documents, spoken-language characteristics as well as the absence of formality and thematic boundaries in transcripts pose additional

difficulty. For example, the boundaries of sentences, paragraphs, and larger text blocks like sections are often missing. Together with speech recognition errors as well as other speech characteristics such as speech disfluences, they will impair the conditions on which an effective and reliable algorithm of discourse analysis is often built.

3 Problem formulation

We are given a speech sequence $U = u_1, u_2, ..., u_m$, where u_i is an utterance. Depending on the application, u_i can either stand for the audio or transcript of the utterance. We are also given a corresponding hierarchical structure. In our work, this is a sequence of lecture slides containing a set of slide titles and bullets, $B = \{b_1, b_2, ..., b_n\}$, organized in a tree structure $T(\Re, \aleph, \Psi)$, where $\Re$ is the root of the tree that concatenates all slides of a lecture; i.e., each slide is a child of the root $\Re$ and each slide's bullets form a subtree. In the rest of this paper, the word *bullet* means both the title of a slide (if any) and any bullet in it. $\aleph$ is the set of nodes of the tree (both terminal and non-terminals, excluding the root $\Re$), each corresponding to a bullet b_i in the slides. Ψ is the edge set. With the definitions, our task is herein to find the triple (b_i, u_k, u_l), denoting that a bullet b_i starts from the kth utterance u_k and ends at the lth. Constrained by the tree structure, the text span corresponding to an ancestor bullet contains those corresponding to its descendants; i.e., if a bullet b_i is the ancestor of another bullet b_j in the tree, the acquired boundary triples (b_i, u_{k1}, u_{l1}) and (b_j, u_{k2}, u_{l2}) should satisfy $u_{k1} \leq u_{k2}$ and $u_{l1} \geq u_{l2}$. In implementation, we only need to find the starting point of a bullet, i.e., a pair (b_i, u_k), since we know the tree structure in advance and therefore we know that the starting position of the next sibling bullet is the ending boundary for the current bullet.

4 Our approaches

Our task is to find the correspondence between slide bullets and a speech sequence or its transcripts. Research on finding correspondences between parallel texts pervades natural language processing. For example, aligning bilingual sentence pairs is an essential step in training machine translation models. In text summarization, the correspondence between human-written summaries and their original texts has been identified (Jing, 2002), too. In speech recognition, forced alignment is applied to align speech and transcripts. In this paper, we keep the general framework of alignment in solving our problem.

Our solution, however, should be flexible to consider multiple constraints such as those conveyed in hierarchical bullet structures and global word distribution. Accordingly, the model proposed in this paper depends on two orthogonal strategies to ensure efficiency and richness of the model. First of all, we formulate all our solutions within a classic dynamic programming framework to enforce computational efficiency (section 4.1). On the other hand, we explore the approach to incorporating hierarchical and global features into the alignment framework (Section 4.2). The associated parameters are then optimized with Powell's algorithm (Section 4.3).

4.1 A pre-order walk of bullet trees

We formulate our solutions within the classic dynamic-programming-based alignment framework, dynamic time warping (DTW). To this end, we need to sequentialize the given hierarchies, i.e., bullet trees. We propose to do so through a pre-order walk of a bullet tree; i.e., at any step of a recursive traversal of the tree, the alignment model always visits the root first, followed by its children in a left-to-right order. This sequentialization actually corresponds to a reasonable assumption: words appearing earlier on a given slide are spoken earlier by the speaker. The pre-order walk is also used by (Branavan et al., 2007) to reduce the search space of their discriminative table-of-contents generation. Our sequentialization strategy can be intuitively thought of as removing indentations that lead each bullet. As shown in Figure 1, the right panel is a bullet array resulting from a pre-walk of the slide in the left panel. In our baseline model, the resulted bullet array is directly aligned with lecture utterances.

Other orders of bullet traversal could also be considered, e.g., when speech does not strictly follow bullet orders. In general, one can regard our

task here as a tagging problem to allow further flexibility on bullet-utterance correspondence, in which bullets are thought of as tags. However, considering the fact that bullets are created to organize speech and in most cases they correspond to the development of speech content monotonically, this paper focuses on addressing the problem in the alignment framework.

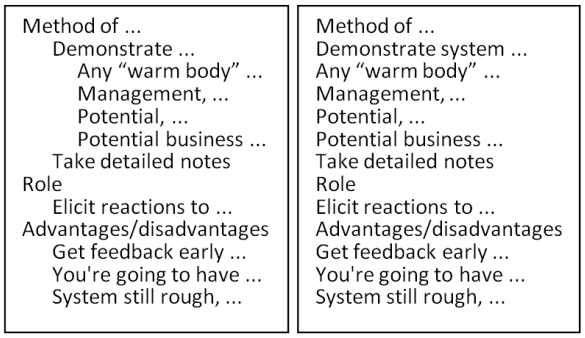


Figure 1: A pre-order walk of a bullet tree.

4.2 Incorporating hierarchical and global features

Our models should be flexible enough to consider constraints that could be helpful, e.g., the hierarchical bullet structures and global word distribution. We propose to consider all these constraints in the phase of estimating similarity matrices. To this end, we use two levels of similarity matrices to capture local tree constraints and global word distributions, respectively.

First of all, information conveyed in the hierarchies of bullet trees should be considered, such as the potentially discriminative nature between two sibling bullets (Branavan et al., 2007) and the relationships between ancestor and descendant bullets. We incorporate them in the bullet-utterance similarity matrices. Specifically, when estimating the similarity between a bullet b_i and an utterance u_j, we consider local tree constraints based on where the node b_i is located on the slide. We do so by accounting for first and second-order tree features. Given a bullet, b_i, we first represent it as multiple vectors, one for each of the following: its own words, the words appearing in its parent bullet, grandparent, children, grandchildren, and the bullets immediately adjacent to b_i. That is, b_i is now represented as 6 vectors of words (we do not discriminate between its left and right siblings and put these words in the same vector). Similarity between the bullet b_i and an utterance u_j is calculated by taking a weighted average over the similarities between each of the 6 vectors and the utterance u_j. A linear combination is used and the weights are optimized on a development set.

Global property of word distributions could be helpful, too. A general term often has less discriminative power in the alignment framework than a word that is localized to a subsection of the document and is related to specific subtopics. For example, in a lecture that teaches introductory computer science topics, aligning a general term "computer" should receive a smaller weight than aligning some topic-specific terms such as "automaton." The latter word is more likely to appear in a more narrow text span. It is not straightforward to directly calculate *idf* scores unless a lecture is split into smaller segments in some way. Instead, in our models, the distribution property of a word is considered in word-level similarity matrices with the following formula.

$$sim(w_i, w_j) = \begin{cases} 0 & : i \neq j \\ 1 - \lambda \frac{var(w_i)}{\max_k(var(w_k))} & : i = j \end{cases}$$

Aligning different words receives no bonus, while matching the same word between bullets and utterances receives a score of 1 minus a distribution penalty, as shown in the formula above. The function $var(w_i)$ calculates the standard variance of the positions where the word w_i appears. Divided by the maximal standard variance of word positions in the same lecture, the score is normalized to [0,1]. This distribution penalty is weighted by λ, which is tuned in a development set. Again, a general term is expected to have a larger positional variance.

Once a word-level matrix is acquired, it is combined with the bullet-utterance level matrix discussed above. Specifically, when measuring the similarity between a word vector (one of the 6 vectors) and the transcripts of an utterance, we sum up the word-level similarity scores of all matching words between them, normalize the resulted score by the length of the vector and utterance, and then renormalize it to the range

[0, 1] within the same spoken document. The final bullet-utterance similarity matrix is incorporated into the pre-order-walk suquentialization discussed above, when alignment is conducted.

4.3 Parameter optimization

Powell's algorithm (Press et al., 2007) is used to find the optimal weights for the constraints we incorporated above, to directly minimize the objective function, i.e., the P_k and WindowDiff scores that we will discuss later. As a summary, we have 7 weights to tune: a weight for each of the following: parent bullet, grandparent, adjacent siblings, children, grandchildren, and the current bullet, plus the word distribution penalty λ. The values of these weights are determined on a development set.

Note that the model we propose here does not exclude the use of further features; instead, many other features, such as smoothed word similarity scores, can be easily added to this model. We are conservative on our model complexity here, in terms of number of weights need to be tuned, for the consideration of the size of data that we can used to estimate these weights. Finally, with all the 7 weights being determined, we apply the standard dynamic time warping (DTW).

5 Experimental set-up

5.1 Data

We use a corpus of lectures recorded at a large research university. The correspondence between bullets and speech utterances are manually annotated in a subset of this lecture corpus, which contains approximately 30,000 word tokens in its manual transcripts. Intuitively, this roughly equals a 120-page double-spaced essay in length. The lecturer's voice was recorded with a head-mounted microphone with a 16kHz sampling rate and 16-bit samples. Students' comments and questions were not recorded. The speech is split into utterances by pauses longer than 200ms, resulting in around 4000 utterances. There are 119 slides that are composed of 921 bullets. A subset containing around 25% consecutive slides and their corresponding speech/transcripts are used as our development set to tune the parameters discussed earlier; the rest data are used as our test set.

5.2 Evaluation metric

We evaluate our systems according to how well the segmentation implied by the inferred bullet alignment matches that of the manually annotated gold-standard bullet alignment. Though one may consider that different bullets may be of different importance, in this paper we do not use any heuristics to judge this and we treat all bullets equally in our evaluation. We evaluate our systems with the P_k and WindowDiff metrics (Malioutov et al., 2007; Beeferman et al., 1999; Pevsner and Hearst, 2002). Note that for both metrics, the lower a score is, the better the performance of a system is. The P_k score computes the probability of a randomly chosen pair of words being inconsistently separated. The WindowDiff is a variant of P_k; it penalizes false positives and near misses equally.

6 Experimental results

6.1 Alignment performance

Table 1 presents the results on automatic transcripts with a 39% WER, a typical WER in realistic and uncontrolled lecture conditions (Leeuwis et al., 2003; Hsu and Glass, 2006). The transcripts were generated with the SONIC toolkit (Pellom, 2001). The acoustic model was trained on the Wall Street Journal dictation corpus. The language model was trained on corpora obtained from the Web through searching the words appearing on slides as suggested by (Munteanu et al., 2007).

	Pk	WindowDiff
UNI	0.481	0.545
TT	0.469	0.534
B-ALN	0.283	0.376
HG-ALN	0.266	0.359

Table 1: The P_k and WindowDiff scores of uniform segmentation (UNI), TextTiling (TT), baseline alignment (B-ALN), and alignment with hierarchical and global information (HG-ALN).

From Table 1, we can see that the model that

utilizes the hierarchical structures of slides and global distribution of words, i.e., the HG-ALN model, reduces both P_k and WindowDiff scores over the baseline model, B-ALN. As discussed earlier, the baseline is a re-implementation of standard dynamic time warping based only on a pre-order walk of the slides, while the HG-ALN model incorporates also hierarchical bullet constraints and global word distribution.

Table 1 also presents the performance of a typical topic segmentation algorithm, TextTiling (Hearst, 1997). Note that similar to (Malioutov et al., 2007), we force the number of predicted topic segments to be the target number, i.e., in our task, the number of bullets. The results show that both the P_k and WindowDiff scores of TextTiling are significantly higher than those of the alignment algorithms. Our manual analysis suggests that many segments are as short as several utterances and the difference between two consecutive segments is too subtle to be captured by a lexical cohesion-based method such as TextTiling. For comparison, We also present the results of uniform segmentation (UNI), which simply splits the transcript of each lecture evenly into segments with same numbers of words.

6.2 Performance under different WERs

Speech recognition errors within reasonable ranges often have very small impact on many spoken language processing tasks such as spoken language retrieval (Garofolo et al., 2000) and speech summarization (Christensen et al., 2004; Maskey, 2008; Murray, 2008; Zhu, 2010). To study the impact of speech recognition errors on our task here, we experimented with the alignment models on manual transcripts as well as on automatic transcripts with different WERs, including a 39% and a 46% WER produced by two real recognition systems. To increase the spectrum of our observation, we also overfit our ASR models to obtain smaller WERs at the levels of 11%, 19%, and 30%.

From Figure 2, we can see that at all levels of these different WERs, the HG-ALN model consistently outperforms the B-ALN system (the AUDIO model will be discussed below). The P_k and WindowDiff curves also show that the alignment performance is sensitive to recognition errors, particularly when the WER is in the range of 30%–45%, suggesting that the problem we study here can benefit from the improvement of current ASR systems in this range, e.g., the recent advance achieved in (Glass et al., 2007).

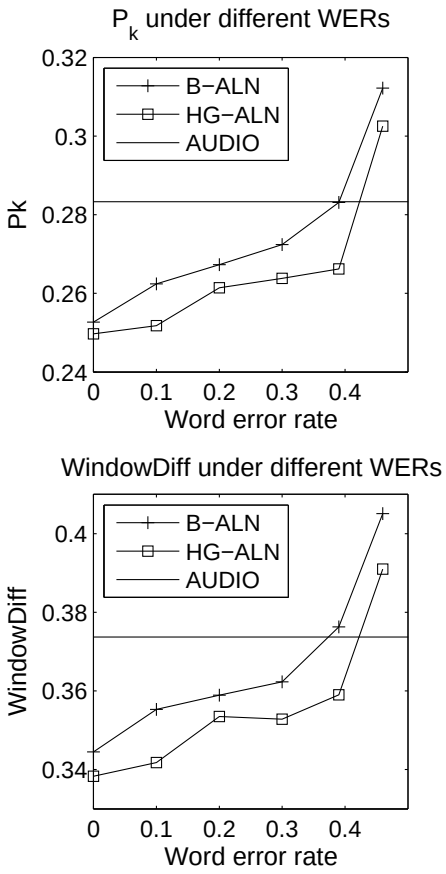


Figure 2: The impact of different WERs on the alignment models. The performance of an audio-based model (AUDIO) is also presented.

6.3 Imposing hierarchical structures onto raw speech

We can actually impose hierarchical structures directly onto raw speech, through estimating the similarity between bullets and speech. This enables navigation through the raw speech by using slides; e.g., one can hear different parts of speech by clicking a bullet. We apply keyword spotting to solve this problem, which detects the occurrences of each bullet word in the corresponding lecture audio.

In this paper, we use a token-passing based algorithm provided in the ASR toolkit SONIC (Pellom, 2001). Since the slides are given in advance, we manually add into the pronunciation dictionary the words that appear in slides but not in the pronunciation dictionary. To estimate similarity between a word vector (discussed earlier in Section 4.2) and an utterance, we sum up all keyword-spotting confidence scores assigned between them, normalize the resulted score by the length of the vector and the duration of the utterance, and then renormalize it to the range [0, 1] within the same spoken lecture.

We present the performance of our bullet-audio alignment model (AUDIO) in Figure 2 so that one can compare its effectiveness with the transcription based methods. The figure shows that the performance of the AUDIO model is comparable to the baseline transcription-based model, i.e., B-ALN, when the WERs of the transcripts are in the range of 37%–39%. The performance is comparable to the HG-ALN model when WERs are in the range of 42%–44%. Also, this suggests that incorporating hierarchical and global features compensates for the performance degradation of speech recognition in this range when the WER is 4%-6% higher.

Note that we did not observe that the performance is different when incorporating hierarchical information and global word distributions into the AUDIO model, so the AUDIO results in Figure 2 are the performance of both types of methods. The current keyword spotting component yields a high false-positive rate; e.g., it incorrectly reports many words that are acoustically similar to parts of other words that really appear in an utterance. This happened even when a high threshold is set. The noise impairs the benefit of hierarchical and distribution features.

7 Conclusions and discussions

This paper investigates the problem of imposing a known hierarchical structure onto an unstructured spoken document. Results show that incorporating local hierarchical constraints and global word distributions in the efficient dynamic programming framework yields a better performance over the baseline. Further experiments on a wide range of WERs confirm that the improvement is consistent, and show that both types of models are sensitive to speech recognition errors, particularly when WER increases to 30% and above. Moreover, directly imposing hierarchical structures onto raw speech through keyword spotting achieves competitive performance.

References

Beeferman, D., A. Berger, and J. Lafferty. 1999. Statistical models for text segmentation. *Machine Learning*, 34(1-3):177–210.

Branavan, S., Deshpande P., and Barzilay R. 2007. Generating a table-of-contents: A hierarchical discriminative approach. In *Proc. of Annual Meeting of the Association for Computational Linguistics.*

Chen, Y. and W. J. Heng. 2003. Automatic synchronization of speech transcript and slides in presentation. In *Proc. International Symposium on Circuits and Systems*.

Christensen, H., B. Kolluru, Y. Gotoh, and S. Renals. 2004. From text summarisation to style-specific summarisation for broadcast news. In *Proc. of the 26th European Conference on Information Retrieval*, pages 223–237.

Fan, Q., K. Barnard, A. Amir, A. Efrat, and M. Lin. 2006. Matching slides to presentation videos using sift and scene background. In *Proc. of ACM International Workshop on Multimedia Information Retrieval*, pages 239–248.

Garofolo, J., G. Auzanne, and E. Voorhees. 2000. The trec spoken document retrieval track: A success story. In *Proc. of Text Retrieval Conference*, pages 16–19.

Glass, J., T. Hazen, S. Cyphers, I. Malioutov, D. Huynh, and R. Barzilay. 2007. Recent progress in the mit spoken lecture processing project. *Proc. of Annual Conference of the International Speech Communication Association*, pages 2553–2556.

Hearst, M. 1997. Texttiling: Segmenting text into multi-paragraph subtopic passages. *Computational Linguistics*, 23(1):33–64.

Hori, C. and S. Furui. 2003. A new approach to automatic speech summarization. *IEEE Transactions on Multimedia*, 5(3):368–378.

Hsu, B. and J. Glass. 2006. Style and topic language model adaptation using hmm-lda. In *Proc. of Conference on Empirical Methods in Natural Language Processing*.

Jing, H. 2002. Using hidden markov modeling to decompose human-written summaries. *Computational Linguistics*, 28(4):527–543.

Leeuwis, E., M. Federico, and M. Cettolo. 2003. Language modeling and transcription of the ted corpus lectures. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing*.

Liu, T., R. Hjelsvold, and J. R. Kender. 2002. Analysis and enhancement of videos of electronic slide presentations. In *Proc. IEEE International Conference on Multimedia and Expo*.

Malioutov, I., A. Park, B. Barzilay, and J. Glass. 2007. Making sense of sound: Unsupervised topic segmentation over acoustic input. In *Proc. of Annual Meeting of the Association for Computational Linguistics*, pages 504–511.

Marcu, D. 2000. The theory and practice of discourse parsing and summarization. The MIT Press.

Maskey, S. 2008. *Automatic Broadcast News Speech Summarization*. Ph.D. thesis, Columbia University.

Munteanu, C., R. Baecker, G. Penn, E. Toms, and E. James. 2006. Effect of speech recognition accuracy rates on the usefulness and usability of webcast archives. In *Proc. of ACM Conference on Human Factors in Computing Systems*, pages 493–502.

Munteanu, C., G. Penn, and R. Baecker. 2007. Web-based language modelling for automatic lecture transcription. In *Proc. of Annual Conference of the International Speech Communication Association*.

Murray, G. 2008. *Using Speech-Specific Characteristics for Automatic Speech Summarization*. Ph.D. thesis, University of Edinburgh.

Pellom, B. L. 2001. Sonic: The university of colorado continuous speech recognizer. *Tech. Rep. TR-CSLR-2001-01, University of Colorado*.

Pevsner, L. and M. Hearst. 2002. A critique and improvement of an evaluation metric for text segmentation. *Computational Linguistics*, 28:19–36.

Press, W.H., S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. 2007. Numerical recipes: The art of science computing. Cambridge University Press.

Ruddarraju, R. 2006. *Indexing Presentations Using Multiple Media Streams*. Ph.D. thesis, Georgia Institute of Technology. M.S. Thesis.

Stark, L., S. Whittaker, and J. Hirschberg. 2000. Finding information in audio: A new paradigm for audio browsing and retrieval. In *Proc. of International Conference on Spoken Language Processing*.

Wang, F., C. W. Ngo, and T. C. Pong. 2003. Synchronization of lecture videos and electronic slides by video text analysis. In *Proc. of ACM International Conference on Multimedia*.

Zechner, K. and A. Waibel. 2000. Minimizing word error rate in textual summaries of spoken language. In *Proc. of Applied Natural Language Processing Conference and Meeting of the North American Chapter of the Association for Computational Linguistics*, pages 186–193.

Zhu, X., X. He, C. Munteanu, and G. Penn. 2008. Using latent dirichlet allocation to incorporate domain knowledge for topic transition detection. In *Proc. of Annual Conference of the International Speech Communication Association*.

Zhu, X. 2010. *Summarizing Spoken Documents Through Utterance Selection*. Ph.D. thesis, University of Toronto.

Interpreting Pointing Gestures and Spoken Requests – A Probabilistic, Salience-based Approach

Ingrid Zukerman and **Gideon Kowadlo** and **Patrick Ye**
Faculty of Information Technology
Monash University
Ingrid.Zukerman@monash.edu, gkowadlo@gmail.com, ye.patrick@gmail.com

Abstract

We present a probabilistic, salience-based approach to the interpretation of pointing gestures together with spoken utterances. Our mechanism models dependencies between spatial and temporal aspects of gestures and features of utterances. For our evaluation, we collected a corpus of requests which optionally included pointing. Our results show that pointing information improves interpretation accuracy.

1 Introduction

DORIS (Dialogue Oriented Roaming Interactive System) is a spoken dialogue system designed for a household robot. In (Zukerman et al., 2008), we described *Scusi?* — a spoken language interpretation module which considers multiple sub-interpretations at different levels of the interpretation process, and estimates the probability of each sub-interpretation at each level (Section 2). This formalism is required for requests such as "Get me the blue cup" in the context of the scene depicted in Figure 1, where possible candidates are the three white cups, and the blue and purple tumblers, but it is unclear which is the intended object, as none of the alternatives match the request perfectly.

In this paper, we integrate pointing gestures into *Scusi?*'s probabilistic formalism. We adopt a salience-based approach, where we take into account spatial and temporal information to estimate the probability that a pointing gesture refers to an object. To evaluate our formalism, we collected a corpus of requests where people were allowed to point (Section 4). Our results show that when people point, our mechanism yields significant improvements in interpretation accuracy; and when pointing was artificially added to utterances where the people did not point, its effect on interpretation accuracy was reduced.

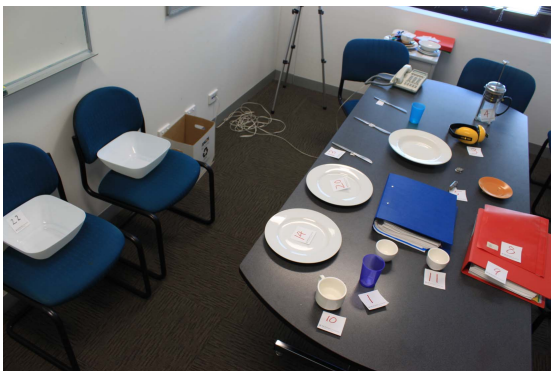


Figure 1: Experimental Setup

This paper is organized as follows. Section 2 outlines the interpretation of a spoken request and the estimation of the probability of an interpretation. Section 3 describes how pointing affects this probability. Our evaluation is detailed in Section 4. Related research is discussed in Section 5, followed by concluding remarks.

2 Interpreting Spoken Requests

Here we summarize our previous work on the interpretation of single-sentence requests (Makalic et al., 2008; Zukerman et al., 2008).

Coling 2008: Poster Volume, pages 1558–1566,
Beijing, August 2010

Scusi? processes spoken input in three stages: speech recognition, parsing and semantic interpretation. First, Automatic Speech Recognition (ASR) software (Microsoft Speech SDK 5.3) generates candidate hypotheses (texts) from a speech signal. The ASR produces up to 50 texts for a spoken utterance, where each text is associated with a probability. In the parsing stage, the texts are considered in descending order of probability. Charniak's probabilistic parser (`ftp://ftp.cs.brown.edu/pub/nlparser/`) is applied to each text, yielding up to 50 parse trees — each associated with a probability.

During semantic interpretation, parse trees are successively mapped into two representations based on Concept Graphs (Sowa, 1984). First *Uninstantiated Concept Graphs (UCGs)*, and then *Instantiated Concept Graphs (ICGs)*. UCGs are obtained from parse trees deterministically — one parse tree generates one UCG. A UCG represents syntactic information, where the concepts correspond to the words in the parent parse tree, and the relations are derived from syntactic information in the parse tree and prepositions. Each UCG can generate many ICGs. This is done by nominating different instantiated concepts and relations from the system's knowledge base as potential realizations for each concept and relation in a UCG. Instantiated concepts are objects and actions in the domain (e.g., `mug01`, `mug02` and `cup01` are possible instantiations of the uninstantiated concept "mug"), and instantiated relations are similar to semantic role labels (Gildea and Jurafsky, 2002). The interpretation process continues until a preset number of sub-interpretations (including texts, parse trees, UCGs and ICGs) has been generated or all options have been exhausted.

Figure 2 illustrates a UCG and an ICG for the request "get the large red folder on the table". The *intrinsic* features of an object (lexical item, colour and size) are stored in the UCG node for this object. *Structural* features, which involve two objects (e.g., "*folder*-on-*table*"), are represented as sub-graphs of the UCG (and the ICG).

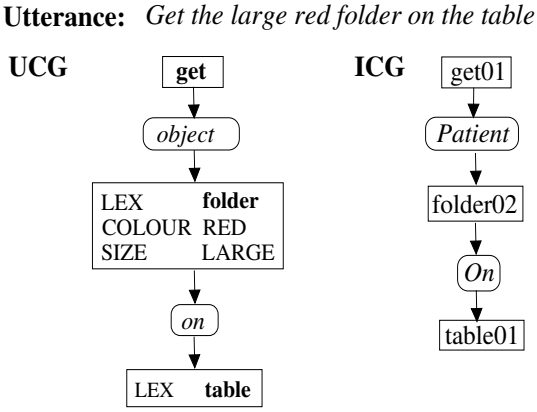


Figure 2: UCG and ICG for a sample utterance

2.1 Estimating the probability of an ICG

Scusi? ranks candidate ICGs according to their probability of being the intended meaning of a spoken utterance. Given a speech signal W and a context $\mathcal{C}$, the probability of an ICG I, $\Pr(I|W,\mathcal{C})$, is proportional to

$$\sum_{\Lambda} \Pr(T|W)\cdot\Pr(P|T)\cdot\Pr(U|P)\cdot\Pr(I|U,\mathcal{C}) \quad (1)$$

where T, P and U denote text, parse tree and UCG respectively. The summation is taken over all possible paths $\Lambda = \{P, U\}$ from the speech wave to the ICG, because a UCG and an ICG can have more than one ancestor. As mentioned above, the ASR and the parser return an estimate of $\Pr(T|W)$ and $\Pr(P|T)$ respectively; and $\Pr(U|P) = 1$, since the process of generating a UCG from a parse tree is deterministic. The estimation of $\Pr(I|U,\mathcal{C})$ is described in (Zukerman et al., 2008). Here we present the final equation obtained for $\Pr(I|U,\mathcal{C})$, and outline the ideas involved in its calculation.

$$\Pr(I|U,\mathcal{C}) \approx \prod_{k\in I} \Pr(u|k,\mathcal{C})\,\Pr(k|k_p,k_{gp})\,\Pr(k|\mathcal{C}) \quad (2)$$

where u is a node in UCG U, k is the corresponding instantiated node in ICG I, k_p is k's parent node, and k_{gp} is k's grandparent node. For example, *On* is the parent of `table01`, and `folder02` the grandparent in the ICG in Figure 2.

- $\Pr(u|k)$ is the "match probability" between the specifications for node u in UCG U and the intrinsic features of the corresponding node k in ICG I, i.e., the probability that a speaker who intended a particular object k gave the specifications in u.

- $\Pr(k|k_p, k_{gp})$ represents the structural probability of ICG I, where structural information is simplified to node trigrams, e.g., whether `folder02` is *On* `table01`.

- $\Pr(k|\mathcal{C})$ is the probability of a concept in light of the context, which includes information about domain objects, actions and relations.

Scusi? handles three intrinsic features: lexical item, colour and size; and two structural features: ownership and several locative relations (e.g., on, under, near). The match probability $\Pr(u|k)$ and the structural probability $\Pr(k|k_p, k_{gp})$ are estimated using a distance function between the requirements specified by the user and what is found in reality — the closer the distance between the specifications and reality, the higher the probability (for details see (Makalic et al., 2008)).

3 Incorporating Pointing Gestures

Pointing affects the salience of objects and the language used to refer to objects: objects in the temporal and spatial vicinity of a pointing gesture are more salient than objects that are farther away, and pointing is often associated with demonstrative determiners. Thus, the incorporation of pointing into *Scusi?* affects the following elements of Equation 2 (Section 2.1).

- $\Pr(k|\mathcal{C})$ – the context-based probability of an object (i.e., its salience) is affected by the time of a pointing gesture and the space it encompasses. For instance, if the user says "Get the cup" in the context of the scene in Figure 1, pointing around the time s/he said "cup", the gesture most likely refers to an object that may be called "cup". Further, among the candidate cups in Figure 1, those closer to the "pointing vector" have a higher probability.[1]

- $\Pr(u|k, \mathcal{C})$ – when pointing, people often use demonstrative determiners, e.g., "get me *that* cup". Also, people often use generic identifiers in conjunction with demonstrative determiners to refer to unfamiliar objects, e.g., "that thing" to refer to a vacuum tube (Figure 1).

These probabilities are estimated in Sections 3.1 and 3.2. Our calculations are based on information returned by the gesture recognition system described in (Li and Jarvis, 2009): gesture type, time, probability and relevant parameters (e.g., a vector for a pointing gesture). Since we focus on pointing gestures, we convert the probabilities expected from Li and Jarvis's system into the probability of Pointing and that of Not Pointing, which comprises all other gestures and no gesture (these hypotheses are returned at the same time).[2]

3.1 Calculating salience from pointing

When pointing is taken into account, the probability of object k is expressed as follows.

$$\begin{aligned}\Pr(k|\mathcal{C}) &= \Pr(k|\mathcal{P}, \mathcal{C}) \cdot \Pr(\mathcal{P}|\mathcal{C}) + \\ &\quad \Pr(k|\neg\mathcal{P}, \mathcal{C}) \cdot \Pr(\neg\mathcal{P}|\mathcal{C})\end{aligned} \tag{3}$$

where $\mathcal{P}$ designates Pointing, $\Pr(\mathcal{P}|\mathcal{C})$ and its complement are returned by the gesture recognition system, and $\Pr(k|\neg\mathcal{P}, \mathcal{C}) = \frac{1}{N}$ (N is the number of objects in the room, i.e., in the absence of pointing, we assume that all the objects in the room are equiprobable[3]).

As indicated above, we posit that pointing is spatially correlated with an intended object, and temporally correlated with a word referring to the intended object. Hence, we separate Pointing into two components: spatial (s) and temporal (t), obtaining $\langle \mathcal{P}_s, \mathcal{P}_t \rangle$. Thus

$$\begin{aligned}\Pr(k|\mathcal{P}, \mathcal{C}) &= \frac{\Pr(k, \mathcal{P}_t, \mathcal{P}_s, \mathcal{C})}{\Pr(\mathcal{P}, \mathcal{C})} \\ &= \frac{\Pr(\mathcal{P}_t|k, \mathcal{P}_s, \mathcal{C}) \cdot \Pr(k|\mathcal{P}_s, \mathcal{C}) \cdot \Pr(\mathcal{P}_s|\mathcal{C})}{\Pr(\mathcal{P}|\mathcal{C})}\end{aligned} \tag{4}$$

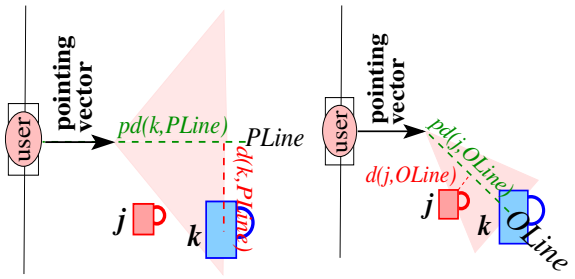
We assume that given k, $\mathcal{P}_t$ is conditionally independent from $\mathcal{P}_s$; and that $\Pr(\mathcal{P}_s|\mathcal{C}) = \Pr(\mathcal{P}|\mathcal{C})$, i.e., the spatial probability of a pointing gesture is the probability returned by the gesture system for the entire pointing hypothesis (time and space). This yields

$$\Pr(k|\mathcal{P}, \mathcal{C}) = \Pr(\mathcal{P}_t|k, \mathcal{C}) \cdot \Pr(k|\mathcal{P}_s, \mathcal{C}) \tag{5}$$

[1] At present, we assume that an utterance is associated with at most one pointing gesture, and that pointing pertains to objects. This assumption is supported by our user study (Section 4.1).

[2] Owing to timing limitations of the gesture recognition system (Section 4), we simulate its output.

[3] At present, we do not consider dialogue salience.



(a) Pointing (b) Occlusion

Figure 3: Spatial pointing and occlusion

where $\Pr(k|\mathcal{P}_s, \mathcal{C})$ and $\Pr(\mathcal{P}_t|k, \mathcal{C})$ are estimated as described in Section 3.1.1 and 3.1.2 respectively. This equation is smoothed as follows (and incorporated into Equation 3) to take into account objects that are (spatially or temporally) excluded from the pointing gesture.

$$\Pr'(k|\mathcal{P}, \mathcal{C}) = \frac{\Pr(k|\mathcal{P}, \mathcal{C}) + \frac{1}{N}}{1 + \sum_{j=1}^{N} \Pr(k_j|\mathcal{P}, \mathcal{C})} \quad (6)$$

3.1.1 Estimating $\Pr(k|\mathcal{P}_s, \mathcal{C})$

$\Pr(k|\mathcal{P}_s, \mathcal{C})$, the probability that the user intended object k when pointing to a location, is estimated using a conic Gaussian density function around *PLine*, the *Pointing Line* created by extending the pointing vector returned by the gesture identification system (Figure 3(a)).[4]

$$\Pr(k|\mathcal{P}_s, \mathcal{C}) = \frac{\alpha\theta_k}{\sqrt{2\pi}\sigma_{P_s}(pd)} e^{-\frac{d(k,PLine)^2}{2\sigma^2_{P_s}(pd)}} \quad (7)$$

where α is a normalizing constant; $\sigma_{P_s}(pd)$ is the standard deviation of the Gaussian cone as a function of $pd(k, PLine)$, the *projected distance* between the user's pointing hand and the projection of object k on *PLine*; $d(k, PLine)$ is the shortest distance between the center of object k and *PLine*; and θ_k is a factor that reduces the probability of object k if it is (partially) *occluded* (Figure 3(b)).

The ***projected distance*** pd takes into account the imprecision of pointing actions — a problem that is exacerbated by the uncertainty associated with sensing a pointing vector. A small angular error in the detected pointing vector yields a discrepancy in the distance between the pointing line and candidate objects. This discrepancy increases as $pd(k, PLine)$ increases. To compensate for this situation, we increase the variance of the Gaussian distribution linearly with the projected distance from the user's hand (we start with a small standard deviation of $\sigma_0 = 5$ mm at the user's fingers, attributed to sensor error). This allows farther objects with a relatively high displacement from the pointing vector to be encompassed in a pointing gesture (e.g., the larger mug in Figure 3(a)), while closer objects with the same displacement are excluded (e.g., the smaller mug). This yields the following equation for the variance.

$$\sigma^2_{P_s}(pd) = \sigma_0^2 + K \cdot pd(k, PLine)$$

where $K = 2.5$ mm is an empirically determined increase rate.

The ***occlusion factor*** θ_k reduces the probability of objects as they become more occluded. We approximate θ_k by considering the objects that are closer to the user than k, and estimating the extent to which these objects occlude k (Figure 3(b)). This estimate is a function of the position of these objects and their size — the larger an intervening object, the lower the probability that the user is pointing at k. These factors are taken into account as follows.

$$\Pr(j \text{ occl } k) = \frac{\gamma}{\sqrt{2\pi}\sigma_\theta(pd)} e^{-\frac{(d(j,OLine) - \frac{1}{2}\dim_{\min}(j))^2}{2\sigma^2_\theta(pd)}} \quad (8)$$

where γ is a normalizing constant; the numerator of the exponent represents the maximum distance from the edge of object j to the line between the user's hand and object k, denoted *Object Line* (*OLine*); and

$$\sigma^2_\theta(pd) = \frac{1}{2}\left(\sigma_0^2 + K \cdot pd(j, OLine)\right)$$

represents the variance of a cone from the user's hand to object k as a function of distance. In order to represent the idea that object j must be close to the Object Line to occlude object k, we use half the variance of that used for the "pointing cone", which yields a thinner "occlusion cone" (Figure 3(b)). θ_k is then estimated as 1 minus the

[4] Since this is a continuous density function, it does not directly yield a point probability. Hence, it is normalized on the basis of the largest possible returned value.

maximum occlusion caused by the objects that are closer to the user than k.

$$\theta_k = 1 - \max_{\forall j \ d(j,\text{hand}) < d(k,\text{hand})} \{\Pr(j \text{ occl } k)\} \quad (9)$$

3.1.2 Estimating $\Pr(\mathcal{P}_t|k,\mathcal{C})$

$\Pr(\mathcal{P}_t|k,\mathcal{C})$ is obtained as follows.

$$\Pr(\mathcal{P}_t|k,\mathcal{C}) = \sum_{i=1}^{n} \frac{\Pr(\mathcal{P}_t, k, W_i, \mathcal{C})}{\Pr(k,\mathcal{C})} \quad (10)$$

$$= \sum_{i=1}^{n} \frac{\Pr(k|\mathcal{P}_t, w_i, \mathcal{C}) \cdot \Pr(T(w_i)|\mathcal{P}_t, \mathcal{C}) \cdot \Pr(\mathcal{P}_t|\mathcal{C})}{\Pr(k|\mathcal{C})}$$

where n is the number of nouns in the user's utterance, and $W_i = \langle w_i, T(w_i)\rangle$ is a tuple comprising the ith noun and the mid point of the time when it was uttered.

We make the following assumptions.

- $\Pr(\mathcal{P}_t|\mathcal{C}) = 1$, as all the gesture hypotheses are returned at the same time;
- given $\mathcal{P}_t$, the timing of a word $T(w_i)$ is conditionally independent of $\mathcal{C}$; and
- given w_i, k is conditionally independent of the timing of the pointing gesture $\mathcal{P}_t$, i.e., $\Pr(k|\mathcal{P}_t, w_i, \mathcal{C}) = \Pr(k|w_i, \mathcal{C})$.

This probability is represented as

$$\Pr(k|w_i,\mathcal{C}) = \frac{\Pr(w_i|k) \cdot \Pr(k|\mathcal{C})}{\sum_{j=1}^{N} \{\Pr(w_i|k_j) \cdot \Pr(k_j|\mathcal{C})\}}$$

where N is the number of objects.

These assumptions yield

$$\Pr(\mathcal{P}_t|k,\mathcal{C}) = \sum_{i=1}^{n} \frac{\Pr(w_i|k) \cdot \Pr(T(w_i)|\mathcal{P}_t)}{\sum_{j=1}^{N} \{\Pr(w_i|k_j) \cdot \Pr(k_j|\mathcal{C})\}} \quad (11)$$

where $\Pr(T(w_i)|\mathcal{P}_t)$, the probability of the time of word w_i given the time of the pointing gesture, is obtained from the following Gaussian time distribution for pointing.

$$\Pr(T(w_i)|\mathcal{P}_t) = \frac{\beta}{\sqrt{2\pi}\sigma_{P_t}} e^{-\frac{(T(w_i)-PTime)^2}{2\sigma_{P_t}^2}} \quad (12)$$

where β is a normalizing constant, *PTime* is the time of the gesture, and σ_{P_t} is the standard deviation of the Gaussian density function, which is currently set to 650 msec (based on our corpus).

As in our previous work (Makalic et al., 2008), we estimate $\Pr(w_i|k)$ using the Leacock and Chodorow (1998) WordNet similarity metric. This metric also yields a match probability between most objects and generic words like "object, thing, here, there", enabling us to handle requests such as "Get that *thing* over *there*".

3.2 Calculating the probability of a referring expression

As mentioned in Section 2, the intrinsic features previously considered in *Scusi?* are lexical item, colour and size (Makalic et al., 2008). Pointing affects referring expressions in that people may point instead of generating complex descriptions, they may employ demonstrative determiners together with generic terms such as "thing" (especially when they are unfamiliar with the name of an object), and they may use demonstrative pronouns. The first two behaviours were exhibited in our user study (Section 4), but none of our trial participants used demonstrative pronouns.

To incorporate pointing into the calculation of $\Pr(u|k,\mathcal{C})$, we add determiners to *Scusi?*'s formalism for intrinsic features, which yields

$$\Pr(u|k,\mathcal{C}) = \Pr(u_{\text{lex}}, u_{\text{det}}, u_{\text{colr}}, u_{\text{size}}|k,\mathcal{C})$$

After adding weights for the intrinsic features (inspired by (Dale and Reiter, 1995)), and making some simplifying assumptions, we obtain

$$\Pr(u|k,\mathcal{C}) = \quad (13)$$
$$\Pr(u_{\text{lex}}|k,\mathcal{C})^{w_{lex}} \cdot \Pr(u_{\text{det}}|k,\mathcal{C})^{w_{det}} \cdot$$
$$\Pr(u_{\text{colr}}|k)^{w_{colr}} \cdot \Pr(u_{\text{size}}|u_{\text{lex}},k)^{w_{size}}$$

The estimation of $\Pr(u_{\text{lex}}|k,\mathcal{C})$, $\Pr(u_{\text{colr}}|k)$ and $\Pr(u_{\text{size}}|u_{\text{lex}},k)$ is described in (Makalic et al., 2008). Here we focus on $\Pr(u_{\text{det}}|k,\mathcal{C})$.

3.2.1 Estimating $\Pr(u_{\text{det}}|k,\mathcal{C})$

$\Pr(u_{\text{det}}|k,\mathcal{C})$ is estimated as follows.

$$\Pr(u_{\text{det}}|k,\mathcal{C}) = \frac{\Pr(k|u_{\text{det}},\mathcal{C}) \cdot \Pr(u_{\text{det}}|\mathcal{C})}{\Pr(k|\mathcal{C})} \quad (14)$$

$$= \frac{\Pr(k|u_{\text{det}},\mathcal{C})}{\Pr(k|\mathcal{C})} \left[\begin{array}{l} \Pr(u_{\text{det}}|\mathcal{P},\mathcal{C}) \cdot \Pr(\mathcal{P}|\mathcal{C}) + \\ \Pr(u_{\text{det}}|\neg\mathcal{P},\mathcal{C}) \cdot \Pr(\neg\mathcal{P}|\mathcal{C}) \end{array} \right]$$

where det = {def_article, indef_article, demonstr_this, demonstr_that}; $\Pr(\mathcal{P}|\mathcal{C})$ and $\Pr(\neg\mathcal{P}|\mathcal{C})$ are returned by the gesture system; $\Pr(u_{\text{det}}|\mathcal{P},\mathcal{C})$ and $\Pr(u_{\text{det}}|\neg\mathcal{P},\mathcal{C})$ are obtained from our corpus; and for now we assume that $\Pr(k|u_{\text{det}},\mathcal{C}) = \Pr(k|\mathcal{C})$.[5] This yields

$$\begin{aligned}\Pr(u_{\text{det}}|k,\mathcal{C}) = \Pr(u_{\text{det}}|\mathcal{P},\mathcal{C}) \cdot \Pr(\mathcal{P}|\mathcal{C}) + \\ \Pr(u_{\text{det}}|\neg\mathcal{P},\mathcal{C}) \cdot \Pr(\neg\mathcal{P}|\mathcal{C})\end{aligned} \quad (15)$$

4 Evaluation

To obtain a corpus, we conducted a user study whereby we set up a room with labeled objects (Figure 1), and asked trial participants to request 12 selected items from *DORIS* (the room included 33 items in total, including distractors, and one of the authors pretended to be *DORIS*). The objects were selected and laid out in the room to reflect a variety of conditions, e.g., common and rare objects (e.g., vacuum tube); unique, non-unique and similar objects (e.g., white cups); and objects placed near each other and far from each other.

We divided our corpus of requests into two parts: with and without pointing. *Scusi?*'s performance was tested on input obtained from the ASR and on textual input (perfect ASR). We considered two scenarios for each sub-corpus: Pointing, where our pointing mechanism was activated on the basis of a simulated pointing gesture,[6] and No-Pointing, where no pointing gesture was detected. This was done in order to test two hypotheses: (1) when people point, pointing information improves interpretation performance; and (2) when they do not point, even perfect pointing has little effect on interpretation performance.

Scusi? was set to generate at most 300 sub-interpretations in total (including texts, parse trees, UCGs and ICGs) for each spoken request, and at most 200 sub-interpretations for each textual request. On average, *Scusi?* takes 10 seconds to go from texts to ICGs. An interpretation was deemed successful if it correctly represented the speaker's intention, which was encoded in one or more *Gold ICG*s. These ICGs were manually constructed on the basis of the requested objects and the participants' utterances. Multiple Gold ICGs were allowed if there were several suitable actions and objects.

4.1 The Corpus

19 people participated in the trial, generating a total of 276 requests, of which 136 involved pointing gestures (3 participants were asked to repeat the experiment after it became clear that they were refraining from pointing, as they erroneously assumed they were not allowed to gesture). We filtered out 64 requests, which included concepts our system cannot yet handle, specifically "the end of the table", projective modifiers (e.g., "behind/left"), ordinals ("first/second"), references to groups of things (e.g., "six blue pens"), and zero- and one-anaphora. This yielded 212 requests, of which 105 involved pointing gestures.

In addition, the software we used has the following limitations: the gesture recognition system (Li and Jarvis, 2009) requires users to hold a gesture for 2 seconds, and the ASR system is speaker dependent and cannot recognize certain words (e.g., "mug", "bowl" and "pen"). To circumvent these problems, each pointing gesture was manually encoded into a time-stamped vector; and one of the authors read slightly sanitized versions of participants' utterances into the ASR: "can you", "please" and "DORIS" were omitted; long prepositional phrases were shortened (e.g., "the thing with wires *sticking out of it*"); and words that were problematic for the ASR were replaced (e.g., "pencil" was used instead of "pen").

There was some difference in the length of requests with and without pointing, but it wasn't as pronounced as reported in (Johnston et al., 2002): requests with/without pointing had 5.84/6.27 words on average. ASR performance was worse for the requests that had pointing, with the top ASR interpretation being correct for only 46% of these requests, compared to 57.5% for the requests without pointing. This difference may be attributed to the ASR having trouble with sentence constructs associated with pointing. Overall

[5] In the future, we will incorporate distance from the user to refine the probabilities of determiners.

[6] At present, we assume accurate pointing and gesture detection, and precise information regarding the position of objects. In the near future, we will study the sensitivity of our mechanism to pointing inaccuracies, and to errors in gesture detection and scene analysis.

	% Gold ICGs in top 1	% Gold ICGs in top 3	Avg adj rank (rank)	% Not found	Avg adj rank (rank) 20	% Not found 20
Sub-corpus without pointing						
Text, *Scusi?*-NoPointing	89.7	93.5	4.39 (0.78)	0.9	1.18 (0.13)	4.7
Text, *Scusi?*-Pointing	86.9	87.9	3.28 (1.89)	0.9	**0.39** (0.35)	4.7
ASR, *Scusi?*-NoPointing	81.3	85.0	4.67 (0.83)	7.5	1.24 (0.17)	12.1
ASR, *Scusi?*-Pointing	79.4	81.3	5.00 (2.62)	5.6	**0.46** (0.40)	12.1
Sub-corpus with pointing						
Text, *Scusi?*-NoPointing	84.8	89.5	3.54 (0.59)	4.8	1.48 (0.20)	9.5
Text, *Scusi?*-Pointing	82.9	86.7	4.19 (1.63)	1.9	**0.41** (0.29)	7.6
ASR, *Scusi?*-NoPointing	76.2	82.9	7.93 (0.95)	10.5	1.79 (0.27)	15.2
ASR, *Scusi?*-Pointing	73.3	81.0	8.65 (2.76)	8.6	**0.68** (0.40)	14.3

Table 1: *Scusi?*'s interpretation performance

the ASR returned the correct interpretation, at any rank, for 88% of the requests.

4.2 Results

Table 1 summarizes our results. Column 1 displays the test condition (sub-corpus with/without pointing, text/ASR, and with/without *Scusi?*'s pointing mechanism). Columns 2-3 show the percentage of utterances that had Gold ICGs whose probability was among the top 1 and top 3, e.g., in the sub-corpus with pointing, when *Scusi?*-Pointing was run on text, 82.9% of the utterances had Gold ICGs with the highest probability (top 1). The average *adjusted rank (AR)* and average *rank* of the Gold ICG appear in Column 4. The rank of an ICG I is its position in a list sorted in descending order of probability (starting from position 0), such that all equiprobable ICGs are deemed to have the same position. The adjusted rank of an ICG I is the mean of the positions of all ICGs that have the same probability as I. For example, if we have 4 equiprobable ICGs in positions 0-3, each has a rank of 0, but an adjusted rank of $\frac{r_{\text{best}}+r_{\text{worst}}}{2} = 1.5$. Column 5 shows the percentage of utterances that didn't yield a Gold ICG. Column 6 shows the average AR for interpretations with AR < 20 (and their average rank), and Column 7 shows the percentage of utterances that had AR ≥ 20 or were not found. We distinguish between Gold ICGs with ARs 0 to 19 and total Gold ICGs that were found, because a dialogue manager is likely to inspect the promising options, i.e., those with AR $< K$ (we assume $K = 20$). In addition, there is normally a trade-off between the number of Not Found Gold ICGs and average AR. ICGs that are not found by one approach but are found by another approach typically have a high (bad) rank when they are eventually found (Zukerman et al., 2008). Thus, an approach that fails to find such "difficult" ICGs usually yields a lower average AR than an approach that finds these ICGs. Capping the ARs of the found Gold ICGs at 20 clarifies the trade-off between average AR and Not Found.

Our results show that, as expected, the main role of pointing is in referent disambiguation. This is evident from the significant reduction in average AR-20 (Column 6) for the pointing and no-pointing sub-corpora, under the text/ASR input conditions. All the differences are statistically significant with $p < 0.01$.[7] Nonetheless, the improvements in average AR-20 obtained by artificially introduced pointing in the no-pointing sub-corpus are smaller for both text and ASR than the improvements obtained with actual pointing. We posit that this smaller impact is due to the fact that utterances without pointing are more descriptive than those with pointing, hence benefitting less from the disambiguating effect of pointing.

The Pointing condition has a seemingly adverse effect on the number of interpretations with top ranks (Columns 2-3). This is explained by the fact

[7]The differences were calculated using a paired t-test for all the Gold ICGs that were found in both configurations.

that all equiprobable interpretations have the same rank, which happens more often under the No-Pointing condition than under the Pointing condition (as pointing has a disambiguating effect).

Finally, under all conditions, the rank of the request at the 75%-ile is 0, which indicates creditable performance. The larger number of Not Found Gold ICGs for the ASR condition is expected, as the ASR failed to find 12% of the correct texts on average, performing worse for the pointing sub-corpus. The other Not Found Gold ICGs were mainly due to parsing preferences, and multiple parses for some utterances that had the word "thing" (which matched all objects).

5 Related Research

Gesture recognition systems endeavour to detect the gesture being made. Common approaches include Hidden Markov Models, e.g., (Nickel and Stiefelhagen, 2003), and Finite State Machines, e.g., (Li and Jarvis, 2009). Systems that focus on pointing also identify the target object, without recognizing the type of this object (Nickel and Stiefelhagen, 2003; Li and Jarvis, 2009).

Most of the research in gesture and speech integration focuses on pointing gestures, employing speech as the main input modality, and using semantic fusion to combine spoken input with gesture. Different approaches are used for gesture detection, e.g., vision (Stiefelhagen et al., 2004; Brooks and Breazeal, 2006) and sensor glove (Corradini et al., 2002); and for language interpretation, e.g., dedicated grammars (Stiefelhagen et al., 2004; Brooks and Breazeal, 2006) and keywords (Einstein and Christoudias, 2004). Fusion is variously implemented using heuristics based on temporal overlap (Bolt, 1980; Johnston et al., 2002), querying a gesture-sensing module when ambiguous referents are identified (Fransen et al., 2007), or unification to determine which elements can be merged (Corradini et al., 2002; Stiefelhagen et al., 2004). These are sometimes combined with search techniques coupled with penalties (Einstein and Christoudias, 2004; Brooks and Breazeal, 2006). With the exception of Bolt's system, these systems were tested on utterances that were quite short and constrained.

Our approach integrates spatial and temporal aspects of gesture into our probabilistic formalism (Zukerman et al., 2008), focusing on the effect of pointing on object salience. Other salience-based approaches are described in (Einstein and Christoudias, 2004; Huls et al., 1995). However, they are not directly comparable with our approach, as they use salience to weigh the importance of factors pertaining to gesture-speech alignment, but there is no uncertainty associated with the visual salience resulting from pointing.

Our use of a probabilistic parser enables us to handle more complex utterances than those considered by most speech-gesture systems (Section 2). At the same time, we do not yet handle speech disfluencies, which are currently handled by (Einstein and Christoudias, 2004; Stiefelhagen et al., 2004). Also, at present we do not consider the challenges pertaining to the real-time synchronization of the output of a gesture-sensing and a speech-recognition system (Stiefelhagen et al., 2004; Brooks and Breazeal, 2006).

6 Conclusion and Future Work

We have extended *Scusi?*, our spoken language interpretation system, to incorporate pointing gestures. Specifically, we have offered a formalism that takes into account relationships between aspects of gesture and spoken language to integrate information about pointing gestures into the estimation of the probability of candidate interpretations of an utterance. Our empirical evaluation shows that our formalism significantly improves interpretation accuracy.

In the future, we propose to refine our model of demonstrative determiners. We also intend to perform sensitivity analysis regarding the accuracy of the vision system, and that of the gesture recognition system. In addition, we will conduct user studies to gain insights with respect to conditions that influence the probability of pointing, e.g., type of object and its position relative to the speaker.

Acknowledgments

This research was supported in part by ARC grant DP0878195. The authors thank R. Jarvis and D. Li for their help with the gesture system.

References

Bolt, R.A. 1980. "Put-that-there": voice and gesture at the graphics interface. In *Proceedings of the 7th Annual Conference on Computer Graphics and Interactive Techniques*, pages 262–270, Seattle, Washington.

Brooks, A.G. and C. Breazeal. 2006. Working with robots and objects: Revisiting deictic reference for achieving spatial common ground. In *Proceedings of the 1st ACM SIGCHI/SIGART Conference on Human-robot Interaction*, pages 297–304, Salt Lake City, Utah.

Corradini, A., R.M. Wesson, and P.R. Cohen. 2002. A Map-Based system using speech and 3D gestures for pervasive computing. In *ICMI'02 – Proceedings of the 4th International Conference on Multimodal Interfaces*, pages 191–196, Pittsburgh, Pennsylvania.

Dale, R. and E. Reiter. 1995. Computational interpretations of the Gricean maxims in the generation of referring expressions. *Cognitive Science*, 18(2):233–263.

Einstein, J. and C.M. Christoudias. 2004. A salience-based approach to gesture-speech alignment. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 25–32, Boston, Massachusetts.

Fransen, B., V. Morariu, E. Martinson, S. Blisard, M. Marge, S. Thomas, A. Schultz, and D. Perzanowski. 2007. Using vision, acoustics, and natural language for disambiguation. In *Proceedings of the ACM/IEEE International Conference on Human-robot Interaction*, pages 73–80, Washington, DC.

Gildea, D. and D. Jurafsky. 2002. Automatic labeling of semantic roles. *Computational Linguistics*, 28(3):245–288.

Huls, C., W. Claassen, and E. Bos. 1995. Automatic referent resolution of deictic and anaphoric expressions. *Computational Linguistics*, 21(1):59–79.

Johnston, M., S. Bangalore, G. Vasireddy, A. Stent, P. Ehlen, M. Walker, S. Whittaker, and P. Maloor. 2002. MATCH: an architecture for multimodal dialogue systems. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 376–383, Philadelphia, Pennsylvania.

Leacock, C. and M. Chodorow. 1998. Combining local context and WordNet similarity forword sense identification. In Fellbaum, C., editor, *WordNet: An Electronic Lexical Database*, pages 265–285. MIT Press.

Li, Z. and R. Jarvis. 2009. Real time hand gesture recognition using a range camera. In *Proceedings of the Australasian Conference on Robotics and Automation*, Sydney, Australia.

Makalic, E., I. Zukerman, M. Niemann, and D. Schmidt. 2008. A probabilistic model for understanding composite spoken descriptions. In *PRICAI 2008 – Proceedings of the 10th Pacific Rim International Conference on Artificial Intelligence*, pages 750–759, Hanoi, Vietnam.

Nickel, K. and R. Stiefelhagen. 2003. Pointing gesture recognition based on 3D-tracking of face, hands and head orientation. In *ICMI'03 – Proceedings of the 5th International Conference on Multimodal Interfaces*, pages 140–146, Vancouver, British Columbia.

Sowa, J.F. 1984. *Conceptual Structures: Information Processing in Mind and Machine*. Addison-Wesley, Reading, MA.

Stiefelhagen, R., C. Fugen, R. Gieselmann, H. Holzapfel, K. Nickel, and A. Waibel. 2004. Natural human-robot interaction using speech, head pose and gestures. In *IROS 2004 – Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 3, pages 2422–2427, Sendai, Japan.

Zukerman, I., E. Makalic, M. Niemann, and S. George. 2008. A probabilistic approach to the interpretation of spoken utterances. In *PRICAI 2008 – Proceedings of the 10th Pacific Rim International Conference on Artificial Intelligence*, pages 581–592, Hanoi, Vietnam.