Corpus-based Semantic Class Mining: Distributional vs. Pattern-Based Approaches

Shuming Shi[1] **Huibin Zhang**[2*] **Xiaojie Yuan**[2] **Ji-Rong Wen**[1]
[1] Microsoft Research Asia
[2] Nankai University
{shumings, jrwen}@microsoft.com
zhanghuibin@126.com; yuanxj@nankai.edu.cn

Abstract

Main approaches to corpus-based semantic class mining include *distributional similarity* (DS) and *pattern-based* (PB). In this paper, we perform an empirical comparison of them, based on a publicly available dataset containing 500 million web pages, using various categories of queries. We further propose a frequency-based rule to select appropriate approaches for different types of terms.

1 Introduction

Computing the semantic relationship between terms, which has wide applications in natural language processing and web search, has been a hot topic nowadays. This paper focuses on corpus-based semantic class mining (Lin 1998; Pantel and Lin 2002; Pasca 2004; Shinzato and Torisawa, 2005; Ohshima, et al., 2006; Zhang et al., 2009), where *peer terms* (or *coordinate terms*) are discovered from a corpus.

Existing approaches to semantic class mining could roughly be divided into two categories: *distributional similarity* (*DS*), and *pattern-based* (*PB*). The first type of work (Hindle, 1990; Lin 1998; Pantel and Lin 2002) is based on the distributional hypothesis (Harris, 1985), saying that terms occurring in analogous (lexical or syntactic) *contexts* tend to be similar. DS approaches basically exploit *second-order* co-occurrences to discover strongly associated concepts. In pattern-based approaches (Hearst 1992; Pasca 2004; Shinzato and Torisawa, 2005; Ohshima, et al., 2006; Zhang et al., 2009), patterns are applied to discover specific relationships between terms, from the general *first-order* co-occurrences. For example, "*NP such as NP, NP…, and NP*" is a popular and high-quality pattern for extracting peer terms (and also hyponyms). Besides the natural language patterns, some HTML tag tree patterns (e.g., the drop down list) are also effective in semantic class mining.

It is worth-noting that the word "*pattern*" also appears in some DS approaches (Pasca et al., 2006; Tanev and Magnini, 2006; Pennacchiotti and Pantel, 2009), to represent the *context* of a term or a term-pair, e.g., "(invent, subject-of)" for the term "Edison", and "- starring -" for the term-pair "(The Terminal, Tom Hanks)". Although "*patterns*" are utilized, we categorize them as DS approaches rather than PB, because they match the DS framework well. In this paper, *PB* only refers to the approaches that utilize patterns to exploit *first-order* co-occurrences. And the patterns in DS approaches are called *contexts* in the following part of this paper.

Progress has been made and promising results have been reported in the past years for both DS and PB approaches. However, most previous research work (some exceptions are discussed in related work) involves solely one category of approach. And there is little work studying the comparison of their performance for different types of terms (we use "*term*" to represent a single word or a phrase).

In this paper, we make an empirical study of this problem, based on a large-scale, publicly available dataset containing 500 million web pages. For each approach *P*, we build a *term-similarity graph G*(*P*), with vertices representing terms, and edges being the confidence that the two terms are peers. Approaches are compared by the quality of their corresponding term graphs.

[*] Work done during an internship at Microsoft

Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010), pages 993–1001,
Beijing, August 2010

We measure the quality of a term graph by set expansion. Two query sets are adopted: One contains 49 semantic classes of named entities and 20220 trials (queries), collected by Pantel et al. (2009) from Wikipedia[2]; and the other contains 100 queries of five lexical categories (proper nouns, common nouns, verbs, adjectives, and adverbs), built in this paper for studying the performance comparison on different term types. With the dataset and the query sets, we study the comparison of DS and PB. Key observations and preliminary conclusions are,

- ***DS vs. PB***: DS approaches perform much better on common nouns, verbs, adjectives, and adverbs; while PB generates higher-quality semantic classes for proper nouns.
- ***Lexical vs. Html-tag patterns***: If only lexical patterns are adopted in PB, the performance drops significantly; while the performance only becomes slightly worse with only Html-tag patterns being included.
- ***Corpus-size***: For proper nouns, PB beats DS even based on a much smaller corpus; similarly, for other term types, DS performs better even with a smaller corpus.

Given these observations, we further study the feasibility of selecting appropriate approaches for different term types to obtain better results. A simple and effective frequency-based rule is proposed for approach-selection. Our online semantic mining system (*NeedleSeek*)[3] adopts both PB and DS to build semantic classes.

2 Related Work

Existing efforts for semantic class mining has been done upon various types of data sources, including text-corpora, search-results, and query logs. In corpus-based approaches (Lin 1998; Lin and Pantel 2001; Pantel and Lin 2002; Pasca 2004; Zhang et al., 2009), semantic classes are obtained by the offline processing of a corpus which can be unstructured (e.g., plain text) or semi-structured (e.g., web pages). Search-results-based approaches (Etzioni et al., 2004; Kozareva et al., 2008; Wang and Cohen, 2008) assume that multiple terms (or, less often, one term) in a semantic class have been provided as *seeds*. Other terms in the class are retrieved by sending queries (constructed according to the seeds) to a web search engine and mining the search results. Query logs are exploited in (Pasca 2007; Komachi and Suzuki, 2008; Yamaguchi 2008) for semantic class mining. This paper focuses on corpus-based approaches.

As has been mentioned in the introduction part, primarily two types of methodologies are adopted: DS and PB. Syntactic context information is used in (Hindle, 1990; Ruge, 1992; Lin 1998; Lin and Pantel, 2001; Pantel and Lin, 2002) to compute term similarities. The construction of syntactic contexts requires sentences to be parsed by a dependency parser, which may be extremely time-consuming on large corpora. As an alternative, lexical context (such as text window) has been studied (Pantel et al., 2004; Agirre et al., 2009; Pantel et al., 2009). In the pattern-based category, a lot of work has been done to discover term relations by sentence lexical patterns (Hearst 1992; Pasca 2004), HTML tag patterns (Shinzato and Torisawa, 2005), or both (Shi et al., 2008; Zhang et al., 2009). In this paper, our focus is not one specific methodology, but the comparison and combination of them.

A small amount of existing work is related to the comparison or combination of multiple methods. Pennacchiotti and Pantel (2009) proposed a feature combination framework (named ensemble semantic) to combine features generated by different extractors (distributional and "pattern-based") from various data sources. As has been discussed in the introduction, in our terminology, their "pattern-based" approaches are actually DS for term-pairs. In addition, their study is based on three semantic classes (actors, athletes, and musicians), all of which are proper nouns. Differently, we perform the comparison by classifying terms according to their lexical categories, based on which additional insights are obtained about the pros and cons of each methodology. Pantel et al., (2004) proposed, in the scenario of extracting *is-a* relations, one pattern-based approach and compared it with a baseline syntactic distributional similarity method (called *syntactic co-occurrence* in their paper). Differently, we study the comparison in a different scenario (semantic class mining). In addition, they did not differentiate the lexical types of terms in the study. The third difference is that we proposed a rule for method-selection while they did not. In (Pasca and Durme,

[2] http://www.wikipedia.org/
[3] http://needleseek.msra.cn/

2008), clusters of distributional similar terms were adopted to expand the labeled semantic classes acquired from the "such as | including" pattern. Although both patterns and distributional similarity were used in their paper, they did not do any comparison about their performance. Agirre et al. (2009) compared DS approaches with WordNet-based methods in computing word similarity and relatedness; and they also studied the combination of them. Differently, the methods for comparison in our paper are DS and PB.

3 Similarity Graph Construction

A key operation in corpus-based semantic class mining is to build a term similarity graph, with vertices representing terms, and edges being the similarity (or distance) between terms. Given the graph, a clustering algorithm can be adopted to generate the final semantic classes. Now we describe the state-of-the-art DS and PB approaches for computing term similarities.

3.1 Distributional Similarity

DS approaches are based on the distributional hypothesis (Harris, 1985), which says that terms appearing in analogous contexts tend to be similar. In a DS approach, a term is represented by a feature vector, with each feature corresponding to a context in which the term appears. The similarity between two terms is computed as the similarity between their corresponding feature vectors. Different approaches may have different ways of 1) defining a context, 2) assigning feature values, or 3) measuring the similarity between two feature vectors.

Contexts	Text window (window size: 2, 4)
	Syntactic
Feature value	PMI
Similarity measure	Cosine, Jaccard

Table 1. DS approaches implemented in this paper

Mainly two kinds of contexts have been extensively studied: *syntactic context* and *lexical context*. The construction of *syntactic contexts* relies on the syntactic parsing trees of sentences, which are typically the output of a syntactic parser. Given a syntactic tree, a syntactic context of a term w can be defined as the parent (or one child) of w in the tree together with their relationship (Lin, 1998; Pantel and Lin, 2002; Pantel et al., 2009). For instance, in the syntactic tree of sentence "*this is an interesting read for anyone studying logic*", one context of the word "logic" can be defined as "*study V:obj:N*". In this paper, we adopt Minipar (Lin, 1994) to parse sentences and to construct syntactic trees.

One popular *lexical context* is *text window*, where a context c for a term w in a sentence S is defined as a substring of the sentence containing but removing w. For example, for sentence "$\ldots w_1w_2w_3ww_4w_5w_6\ldots$", a text window context (with size 4) of w can be "$w_2w_3w_4w_5$". It is typically time-consuming to construct the syntactic trees for a large-scale dataset, even with a lightweight syntactic parser like Minipar. The construction of lexical contexts is much more efficient because it does not require the syntactic dependency between terms. Both contexts are studied in this paper.

After defining contexts for a term w, the next step is to construct a feature vector for the term: $F(w)=(f_{w1}, f_{w2}\ldots, f_{w,m})$, where m is the number of distinct contexts, and $f_{w,c}$ is the feature value of context c with respect to term w. Among all the existing approaches, the dominant way of assigning feature values (or context values) is computing the pointwise mutual information (PMI) between the feature and the term,

$$f_{w,c} = \mathrm{PMI}_{w,c} = \log\frac{F(w,c)\cdot F(*,*)}{F(w,*)\cdot F(*,c)} \qquad (3.1)$$

where $F(w,c)$ is the frequency of context c occurring for term w, $F(w,*)$ is the total frequency of all contexts for term w, $F(*,c)$ is the frequency of context c for all terms, and $F(*,*)$ is the total frequency of all context for all terms. They are calculated as follows respectively,

$$\begin{aligned} F(w,*) &= \textstyle\sum_{j=1}^{m} F(w,j) \\ F(*,c) &= \textstyle\sum_{i=1}^{n} F(i,c) \\ F(*,*) &= \textstyle\sum_{i=1}^{n}\sum_{j=1}^{m} F(i,j) \end{aligned} \qquad (3.2)$$

where m and n are respectively the distinct numbers of contexts and terms.

Following state-of-the-art, we adopt PMI in this paper for context weighting.

Given the feature vectors of terms, the similarity of any two terms is naturally computed as the similarity of their corresponding feature vectors. Cosine similarity and Jaccard similarity (weighted) are implemented in our experiments,

$$Cosine(\vec{x},\vec{y}) = \frac{\sum_i x_i y_i}{\sqrt{\sum_i x_i^2}\cdot\sqrt{\sum_i y_i^2}} \qquad (3.3)$$

$$Jaccard(\vec{x},\vec{y}) = \frac{\sum_i \min(x_i,y_i)}{\sum_i x_i + \sum_i y_i - \sum_i \min(x_i,y_i)} \quad (3.4)$$

Jaccard similarity is finally used in presenting our experimental results (in Section 6), because it achieves higher performance.

3.2 Pattern-based Approaches

In PB approaches, a list of carefully-designed (or automatically learned) patterns is exploited and applied to a text collection, with the hypothesis that the terms extracted by applying each of the patterns to a specific piece of text tend to be similar. Two categories of patterns have been studied in the literature: sentence lexical patterns, and HTML tag patterns. Table-2 lists some popular patterns utilized in existing semantic class mining work (Heast 1992; Pasca 2004; Kozareva et al., 2008; Zhang et al., 2009). In the table, "T" means a term (a word or a phrase). Exactly the same set of patterns is employed in implementing our pattern-based approaches in this paper.

Type	Pattern
Lexical	T {, T}* {,} (and|or) {other} T
	(such as | including) T (and|,|.)
	T, T, T {,T}*
Tag	- T … - T
	- T … - T
	<select> <option> T …<option> T </select>
	<table> <tr> <td> T </td> … <td> T </td> </tr> ... </table>
	Other Html-tag repeat patterns

Table 2. Patterns employed in this paper (Lexical: sentence lexical patterns; Tag: HTML tag patterns)

We call the set of terms extracted by applying a pattern one time as a *raw semantic class* (RASC). The term similarity graph needs to be built by aggregating the information of the extracted RASCs.

One basic idea of estimating term similarity is to count the number of RASCs containing both of them. This idea is extended in the state-of-the-art approaches (Zhang et al., 2009) to distinguish the reliability of different patterns and to punish term similarity contributions from the same domain (or site), as follows,

$$Sim(a,b) = \sum_{i=1}^{m} \log(1 + \sum_{j=1}^{k_i} w(P(C_{i,j}))) \quad (3.5)$$

where $C_{i,j}$ is a RASC containing both term a and term b, $P(C_{i,j})$ is the pattern via which the RASC is extracted, and $w(P)$ is the weight of pattern P. The above formula assumes all these RASCs belong to m sites (or domains) with $C_{i,j}$ extracted from a page in site i, and k_i being the number of RASCs corresponding to site i.

In this paper, we adopt an extension of the above formula which considers the frequency of a single term, as follows,

$$Sim^*(a,b) = Sim(a,b) \cdot \sqrt{IDF(a) \cdot IDF(b)} \quad (3.6)$$

where $IDF(a)=\log(1+N/N(a))$, N is the total number of RASCs, and $N(a)$ is the number of RASCs containing a. In the experiments, we simply set the weight of every pattern type to be the same value (1.0).

4 Compare PB and DS

We compare PB and DS by the quality of the term similarity graphs they generated. The quality of a term graph is measured by set expansion: Given a list of *seed* terms (e.g., S={*lent*, *epiphany*}) belonging to a semantic class, our task is to find other members of this class, such as *advent*, *easter*, and *christmas*.

In this section, we first describe our set expansion algorithm adopted in our study. Then DS and PB are compared in terms of their set-expansion performance. Finally we discuss ways of selecting appropriate approaches for different types of seeds to get better expansion results.

4.1 Set Expansion Algorithm

Having at hand the similarity graph, set expansion can be implemented by selecting the terms most similar to the seeds. So given a query Q={s_1, s_2, …, s_k}, the key is to compute $f(t,Q)$, the similarity between a term t and the seed-set Q. Naturally, we define it as the weighted average similarity between t and every seed in Q,

$$f(t,Q) = \sum_{i=1}^{k} w_i \cdot Sim(t,s_i) \quad (4.1)$$

where w_i is the weight of seed s_i, which can be a constant value, or a function of the frequency of term s_i in the corpus. Although Formula 3.6 can be adopted directly for calculating $Sim(t,s_i)$, we use the following rank-based formula because it generate better expansion results.

$$Sim(t,s_i) = \frac{1}{\log(\lambda + r(t,s_i))} \quad (4.2)$$

where $r(t,s_i)$ is the rank of term t among the neighbors of s_i.

In our experiments, we fix w_i=1 and λ=10.

4.2 Compare DS with PB

In order to have a comprehensive comparison of the two approaches, we intentionally choose terms of diverse types and do experiments based on various data scales. We classify terms into 5 types by their lexical categories: proper nouns, common nouns, verbs, adjectives, and adverbs. The data scales for experiments are from one million to 500 million web pages. Please refer to sections 5.1 and 5.2 for more details about the corpora and seeds used for experiments.

Experimental results (refer to Section 6) will show that, for proper nouns, the ranking of approaches (in terms of performance) is:

$$PB > PB\text{-}HtmlTag > DS \approx PB\text{-}Lexical$$

While for common nouns, verbs, adjectives, and adverbs, we have:

$$DS > PB$$

Here "PB-lexical" means only the lexical patterns of Table 2 are adopted. Similarly, "PB-HtmlTag" represents the PB approach with only Html-tag patterns being utilized.

Please pay attention that this paper by no means covers all PB or DS approaches (although we have tried our best to include the most popular ones). For PB, there are of course other kinds of patterns (e.g., patterns based on deeper linguistic analysis). For DS, other types of contexts may exist in addition to those listed in Table 1. So in interpreting experimental results, making observations, and drawing preliminary conclusions, we only means the patterns in Table 2 for PB and Table 1 for DS. It will be an interesting future work to include more DS and PB approaches in the study.

In order to understand why PB performs so well in dealing with proper nouns while so badly for other term categories, we calculated the frequency of each seed term in the extracted *RASCs*, the output of the pattern-matching algorithm. We define the *normalized frequency* of a term to be its frequency in the RASCs divided by the frequency in the sentences of the original documents (with duplicate sentences merged). Then we define the *mean normalized frequency* (MNF) of a seed set *S*, as follows,

$$MNF(S) = \frac{\sum_{t \in S} F_{norm}(t)}{|S|} \qquad (4.3)$$

where $F_{norm}(t)$ is the normalized frequency of t.

The MNF values for the five seed sets are listed in Table 3, where we can see that proper nouns have the largest MNF values, followed by common nouns. In other words, the patterns in Table 2 capture the relations of more proper nouns than other term categories.

Seed Categories	Terms	MNF
Proper nouns	40	0.2333
Common nouns	40	0.0716
Verbs	40	0.0099
Adjectives	40	0.0126
Adverbs	40	0.0053

Table 3. MNF values of different seed categories

As mentioned in the introduction, the PB and DS approaches we studied capture first-order and second-order term co-occurrences respectively. Some existing work (e.g., Edmonds, 1997) showed that second-order co-occurrence leads to better results for detecting synonymy. Considering that a high proportion of coordinate terms of verbs, adjectives, and adverbs are their synonyms and antonyms, it is reasonable that DS behaves better for these term types because it exploits second-order co-occurrence. For PB, different from the standard way of dealing with first-order co-occurrences where statistics are performed on all pairs of near terms, a *subset* of co-occurred terms are *selected* in PB by specific patterns. The patterns in Table-2 help detecting coordinate proper nouns, because they are frequently occurred together obeying the patterns in sentences or web pages. But it is not the case for other term types. It will be interesting to study the performance of PB when more pattern types are added.

4.3 Approach Selection

Having observed that the two approaches perform quite differently on every type of queries we investigated, we hope we can improve the expansion performance by smartly selecting an approach for each query. In this section, we propose and study several approach-selection methods, by which we hope to gain some insights about the possibility and effectiveness of combining DS and PB for better set expansion.

Oracle selection: In order to get an insight about the upper bound that we could obtain when combing the two methods, we implement an oracle that chooses, for each query, the approach that generates better expansion results.

Frequency-based selection: It is shown in Table 3 that the *mean normalized frequency* of proper nouns is much larger than other terms. Motivated by this observation, we select a set expansion methodology for each query as follows: Select PB if the normalized frequency values of all terms in the query are larger than 0.1; otherwise choose DS.

We demonstrate, in Section 6.3, the effectiveness of the above selection methods.

5 Experimental Setup

5.1 Dataset and Exp. Environment

We adopt a public-available dataset in our experiments: ClueWeb09[4]. This is a very large dataset collected by Carnegie Mellon University in early 2009 and has been used by several tracks of the Text Retrieval Conference (TREC)[5]. The whole dataset consists of 1.04 billion web pages in ten languages while only those in English, about 500 million pages, are used in our experiments. The reason for selecting such a dataset is twofold: First, it is a corpus large enough for conducting web-scale experiments and getting meaningful results. Second, since it is publicly available, it is possible for other researchers to reproduce the experiments in the paper.

Corpora	Docs (millions)	Sentences (millions)	Description
Clue500	500	13,000	All En pages in ClueWeb09
Clue050	50	1,600	ClueWeb09 category B
Clue010	10	330	Sampling from Clue050
Clue001	1	42	Sampling from Clue050

Table 4. Corpora used in experiments

To test the impact of corpus size on set expansion performance, four corpora are derived from the dataset, as outlined in Table 4. The Clue500 corpus contains all the 500 million English web pages in the dataset; while Clue050 is a subset of ClueWeb09 (named category B) containing 50 million English web pages. The remaining two corpora are respectively the 1/5 and 1/50 random sampling of web pages from Clue050.

Documents in the corpora are stored and processed in a cluster of 40 four-core machines.

[4] http://boston.lti.cs.cmu.edu/Data/clueweb09/
[5] http://trec.nist.gov/

5.2 Query Sets

We perform our study using two query sets.

WikiGold: It was collected by Pantel et al. (2009) from the "List of" pages in Wikipedia and used as the gold standard in their paper. This gold standard consists of 49 entity sets, and 20220 trials (used as queries) of various numbers of seeds. Most seeds in the query set are named entities. Please refer to Pantel et al. (2009) for details of the gold standard.

Mix100: This query set consists of 100 queries in five categories: verbs, adjectives, adverbs, common nouns, and proper nouns. There are 20 queries in every category and two seeds in every query. The query set was built by the following steps: First, 20 terms of each category were randomly selected from a term list (which is constructed by part-of-speech tagging the Clue050 corpus and removing low-frequency terms), and were treated as the first seed of the each query. Then, we manually added one additional seed for each query. The reason for utilizing two seeds instead of one is the observation that a large portion of the terms selected in the previous step belong to multiple categories. For example, "*colorful*" is both an adjective and a proper noun (a Japanese manga).

5.3 Results Labeling

No human labeling efforts are needed for the expansion results of the WikiGold query set. Every returned term is automatically judged to be "Good" (otherwise "Bad") if it appears in the corresponding gold standard entity set.

For Mix100, the search results of various approaches are merged and labeled by three human labelers. Each labeler assigns each term in the search results a label of "Good", "Fair" or "Bad". The labeling agreement values (measured by percentage agreement) between labelers I and II, I and III, II and III are respectively 0.82, 0.81, and 0.81. The ultimate judgment of each result term is obtained from the three labelers by majority voting. In the case of three labelers giving mutually different results (i.e., one "Good", one "Fair" and one "Bad"), the ultimate judgment is set to "Fair" (the average).

5.4 Evaluation Metrics

After removing seeds from the expansion results, we adopt the following metrics to evaluate the

results of each query. The evaluation score on a query set is the average over all the queries.

Precision@*k*: The percentage of relevant (good or fair) terms in the top-*k* expansion results (terms labeled as "Fair" are counted as 0.5)

Recall@*k*: The ratio of relevant terms in the top-*k* results to the total number of relevant terms

R-Precision: Precision@R where *R* is the total number of terms labeled as "Good"

Mean average precision (MAP): The average of precision values at the positions of all good or fair results

6 Experimental Results

6.1 Overall Performance Comparison

Table 5 lists the performance (measured by MAP, R-precision, and the precisions at ranks 25, 50, and 100) of some key approaches on corpus Clue050 and query set WikiGold. The results of query set Mix100 are shown in Table 6. In the results, *TWn* represents the DS approach with text-window of size *n* as contexts, *Syntactic* is the DS approach with syntactic contexts, *PB-Lexical* means only the lexical patterns of Table 2 are adopted, and *PB-HtmlTag* represents the PB approach with only Html-tag patterns utilized.

Approach	MAP	R-Prec	P@25	P@50	P@100
TW2	0.218	0.287	0.359	0.278	0.204
TW4	0.152	0.210	0.325	0.244	0.173
Syntactic	0.170	0.247	0.314	0.242	0.178
PB-Lexical	0.227	0.276	0.352	0.272	0.190
PB-HtmlTag	0.354	0.417	0.513	0.413	0.311
PB	**0.362**	**0.424**	**0.520**	**0.418**	**0.314**
Pantel-24M	N/A	0.264	0.353	0.298	0.239
Pantel-120M	N/A	0.356	0.377	0.319	0.250
Pantel-600M	N/A	0.404	0.407	0.347	0.278

Table 5. Performance comparison on the Clue050 corpus (query set: WikiGold)

It is shown that PB gets much higher evaluation scores than other approaches on the *WikiGold* query set and the proper-nouns category of *Mix100*. While for other seed categories in *Mix100*, TW2 return significantly better results. We noticed that most seeds in *WikiGold* are proper nouns. So the experimental results tend to indicate that the performance comparison between state-of-the-art DS and PB approaches depends on the types of terms to be mined, specifically, DS approaches perform better in mining semantic classes of common nouns, verbs, adjectives, and adverbs; while state-of-the-art PB approaches are more suitable for mining semantic classes of proper nouns. The performance of PB is low in dealing with other types of terms (especially adverbs). The performance of PB drops significantly if only lexical patterns are used; and the HtmlTag-only version of PB performs only slightly worse than PB.

The observations are verified by the precision-recall graph in Figure 1 on Clue500. The results of the *syntactic* approach on Clue500 are not included, because it is too time-consuming to parse all the 500 million web pages by a dependency parser (even using a high-performance parser like Minipar). It took overall about 12,000 CPU-hours to parse all the sentences in Clue050 by Minipar.

Query types & Approaches		MAP	P@5	P@10	P@20
Proper Nouns	TW2	0.302	0.835	0.810	0.758
	PB	**0.336**	**0.920**	**0.838**	**0.813**
Common Nouns	TW2	**0.384**	**0.735**	**0.668**	**0.595**
	PB	0.212	0.640	0.548	0.485
Verbs	TW2	**0.273**	**0.655**	**0.543**	**0.465**
	PB	0.176	0.415	0.373	0.305
Adjectives	TW2	**0.350**	**0.655**	**0.563**	**0.473**
	PB	0.120	0.335	0.285	0.234
Adverbs	TW2	**0.432**	**0.605**	**0.505**	**0.454**
	PB	0.043	0.100	0.095	0.089

Table 6. Performance comparison on different query types (Corpus: Clue050; query set: Mix100)

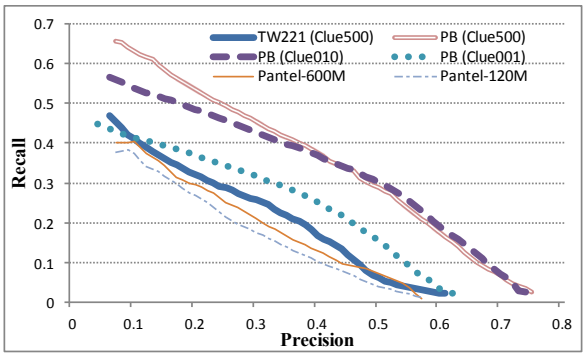

Figure 1. Precision and recall of various approaches (query set: WikiGold)

The methods labeled Pantel-24M etc. (in Table 5 and Figure 1) are the approaches presented in (Pantel et al., 2009) on their corpus (called Web04, Web20, and Web100 in the paper) containing respectively 24 million, 120 million, and 600 million web pages. Please pay attention that their results and ours may not be directly comparable, because different corpora and set-

expansion algorithms were used. Their results are listed here for reference purpose only.

6.2 Corpus Size Effect

Table 7 shows the performance (measured by MAP) of two approaches on query set *Mix100*, by varying corpus size. We observed that the performance of TW2 improves rapidly along with the growth of corpus size from one million to 50 million documents. From Clue050 to Clue500, the performance is slightly improved.

Query types & Approaches		Clue001	Clue010	Clue050	Clue500
Proper Nouns	TW2	0.209	0.265	0.302	**0.311**
	PB	**0.355**	0.351	0.336	0.327
Common Nouns	TW2	0.259	0.348	0.384	**0.393**
	PB	0.200	**0.234**	0.212	0.205
Verbs	TW2	0.224	0.268	0.273	**0.278**
	PB	0.101	0.134	**0.176**	0.148
Adjectives	TW2	0.309	0.326	0.350	**0.353**
	PB	0.077	**0.158**	0.120	0.129
Adverbs	TW2	0.413	0.423	0.432	**0.437**
	PB	0.028	0.058	0.043	**0.059**

Table 7. Effect of different corpus size (query set: Mix100; metric: MAP)

For PB, however, the performance change is not that simple. For proper nouns, the best performance (in terms of MAP) is got on the two small corpora Clue001 and Clue010; and the score does not increase when corpus size grows. Different observations are made on WikiGold (see Figure 1), where the performance improves a lot with the data growth from Clue001 to Clue010, and then stabilizes (from Clue010 to Clue500). For other term types, the MAP scores do not grow much after Clue010. To our current understanding, the reason may be due to the two-fold effect of incorporating more data in mining: *bringing useful information as well as noise*. Clue001 contains enough information, which is fully exploited by the PB approach, for expanding the proper-nouns in Mix100. So the performance of PB on Clue001 is excellent. The named entities in WikiGold are relatively rare, which requires a larger corpus (Clue010) for extracting peer terms from. But when the corpus gets larger, we may not be able to get more useful information to further improve results quality.

Another interesting observation is that, for proper nouns, the performance of PB on Clue001 is even much better than that of TW2 on corpus Clue500. Similarly, for other query types (common nous, verbs, adjectives, and adverbs), TW2 easily beats PB even with a much smaller corpus.

6.3 Approach Selection

Here we demonstrate the experimental results of combining DS and PB with the methods we proposed in Section 4.3. Table 8 shows the combination of PB and TW2 on corpus Clue050 and query set Mix100. The overall performance relies on the number (or percentage) of queries in each category. Two ways of mixing the queries are tested: avg(4:1:1:1:1) and avg(1:1:1:1:1), where the numbers are the proportion of proper nouns, common nouns, verbs, adjectives, and adverbs.

Approach	Avg (1:1:1:1:1)			Avg (4:1:1:1:1)		
	P@5	P@10	P@20	P@5	P@10	P@20
TW2	0.697	0.618	0.548	0.749	0.690	0.627
PB	0.482	0.428	0.385	0.646	0.581	0.545
Oracle	0.759	0.663	0.591	0.836	0.759	0.695
Freq-based	0.721	0.633	0.570	0.799	0.723	0.671

Table 8. Experiments of combining both approaches (Corpus: Clue050; query set: Mix100)

The expansion performance is improved a lot with our frequency-based combination method. As expected, oracle selection achieves great performance improvement, which shows the large potential of combining DS and PB. Similar results (omitted due to space limitations) are observed on the other corpora.

Our online semantic mining system (*NeedleSeek*, http://needleseek.msra.cn) adopts both PB and DS for semantic class construction.

7 Conclusion

We compared two mainstream methods (DS and PB) for semantic class mining, based on a dataset of 500 million pages and using five term types. We showed that PB is clearly adept at extracting semantic classes of proper nouns; while DS is relatively good at dealing with other types of terms. In addition, a small corpus is sufficient for each approach to generate better semantic classes of its "favorite" term types than those obtained by its counterpart on a much larger corpus. Finally, we tried a frequency-based method of combining them and saw apparent performance improvement.

References

Eneko Agirre, Enrique Alfonseca, Keith Hall, Jana Kravalova, Marius Pasca, Aitor Soroa. A Study on Similarity and Relatedness Using Distributional and WordNet-based Approaches. NAACL-HLT 2009.

Philip Edmonds. 1997. Choosing the Word most Typical in Context Using a Lexical Co-Occurrence Network. *ACL'97*, pages 507-509.

Oren Etzioni, Michael Cafarella, Doug Downey, Stanley Kok, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel Weld, and Alexander Yates. 2004. Web-Scale Information Extraction in KnowItAll. *WWW'04*, pages 100–110, New York.

Zelig S. Harris. 1985. Distributional Structure. *The Philosophy of Linguistics*. New York: Oxford University Press.

Marti A. Hearst. 1992. Automatic Acquisition of Hyponyms from Large Text Corpora. *COLING'92*, Nantes, France.

Donald Hindle. 1990. Noun Classification from Predicate-Argument Structures. In ACL'90, pages 268–275, Pittsburg, Pennsylvania, June.

Mamoru Komachi and Hisami Suzuki. Minimally Supervised Learning of Semantic Knowledge from Query Logs. *IJCNLP 2008*, pages 358–365, 2008.

Zornitsa Kozareva, Ellen Riloff, Eduard Hovy. 2008. Semantic Class Learning from the Web with Hyponym Pattern Linkage Graphs. ACL'08: HLT.

Dekang Lin. 1994. Principar - an Efficient, Broad-Coverage, Principle-based Parser. *COLING'94*, pp. 482-488.

Dekang Lin. 1998. Automatic Retrieval and Clustering of Similar Words. *COLING-ACL'98*, pages 768-774.

Dekang Lin and Patrick Pantel. 2001. Induction of Semantic Classes from Natural Language Text. *SIGKDD'01*, pages 317-322.

Hiroaki Ohshima, Satoshi Oyama and Katsumi Tanaka. 2006. Searching Coordinate Terms with their Context from the Web. *WISE'06*.

Patrick Pantel, Eric Crestan, Arkady Borkovsky, Ana-Maria Popescu and Vishnu Vyas. 2009. Web-Scale Distributional Similarity and Entity Set Expansion. *EMNLP'09*. Singapore.

Patrick Pantel and Dekang Lin. 2002. Discovering Word Senses from Text. *SIGKDD'02*.

Patric Pantel, Deepak Ravichandran, and Eduard Hovy. 2004. Towards Terascale Knowledge Acquisition. *COLING'04*, Geneva, Switzerland.

Marius Pasca. 2004. Acquisition of Categorized Named Entities for Web Search. *CIKM'04*.

Marius Pasca. 2007. Weakly-Supervised Discovery of Named Entities Using Web Search Queries. *CIKM'07*. pp. 683-690.

Marius Pasca and Benjamin Van Durme. 2008. Weakly-supervised Acquisition of Open-Domain Classes and Class Attributes from Web Documents and Query Logs. *ACL'08*.

Marius Pasca, Dekang Lin, Jeffrey Bigham, Andrei Lifchits, and Alpa Jain. 2006. Organizing and Searching the World Wide Web of Facts - Step One: The One-Million Fact Extraction Challenge. In AAAI'06.

Marco Pennacchiotti and Patrick Pantel. 2009. Entity Extraction via Ensemble Semantics. EMNLP'09.

Gerda Ruge. 1992. Experiments on Linguistically-Based Term Associations. *Information Processing & Management*, 28(3): 317-32.

Keiji Shinzato and Kentaro Torisawa. 2005. A Simple WWW-based Method for Semantic Word Class Acquisition. *Recent Advances in Natural Language Processing (RANLP'05)*, Borovets, Bulgaria.

Shuming Shi, Xiaokang Liu, Ji-Rong Wen. 2008. Pattern-based Semantic Class Discovery with Multi-Membership Support. *CIKM'08*, Napa Valley, California, USA.

Hristo Tanev and Bernardo Magnini. 2006. Weakly Supervised Approaches for Ontology Population. EACL'2006, Trento, Italy.

Richard C. Wang and William W. Cohen. 2008. Iterative Set Expansion of Named Entities Using the Web. *ICDM'08*, pages 1091–1096.

Masashi Yamaguchi, Hiroaki Ohshima, Satoshi Oyama, and Katsumi Tanaka. Unsupervised Discovery of Coordinate Terms for Multiple Aspects from Search Engine Query Logs. *The 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*.

Huibin Zhang, Mingjie Zhu, Shuming Shi, and Ji-Rong Wen. 2009. Employing Topic Models for Pattern-based Semantic Class Discovery. *ACL'09*, Singapore.