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

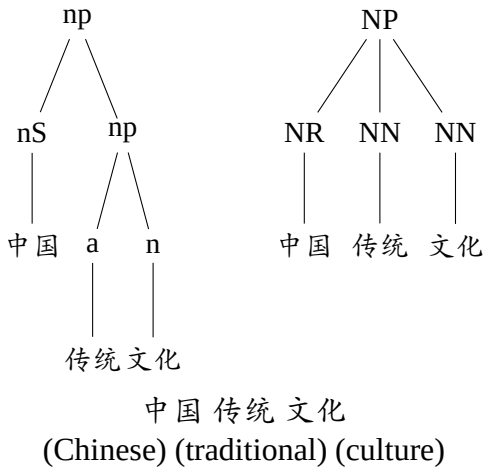

Figure 1: Example tree fragments with TCT (left) and CTB (right) annotations

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.

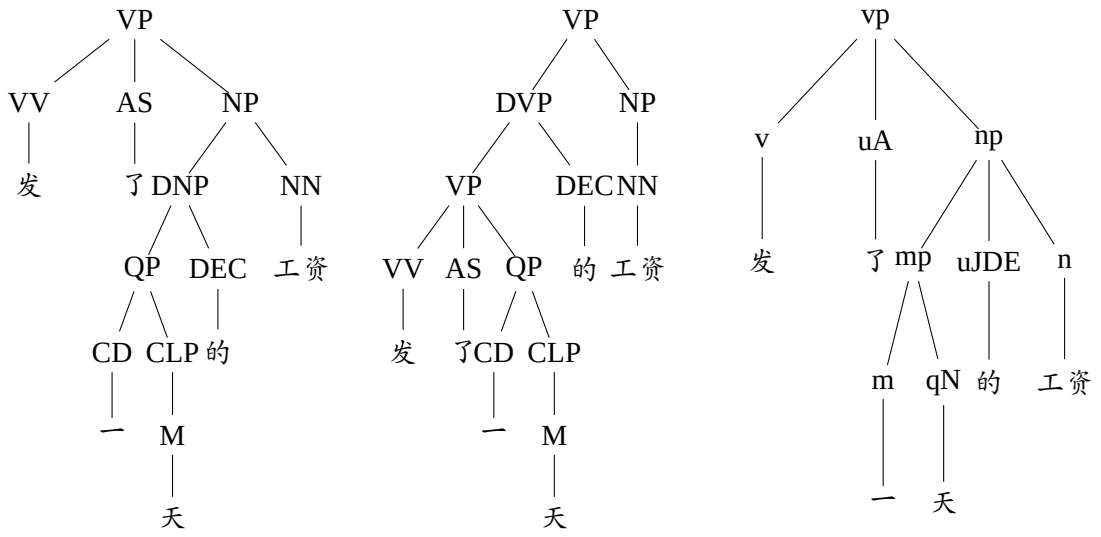

Figure 2: tree fragments of words 发 了 一 天 的 工资: (a) and (b) show two plausible tree fragments of the words using the CTB standard; (c) shows a tree fragment of the TCT standard which has the same interpretation as (a).

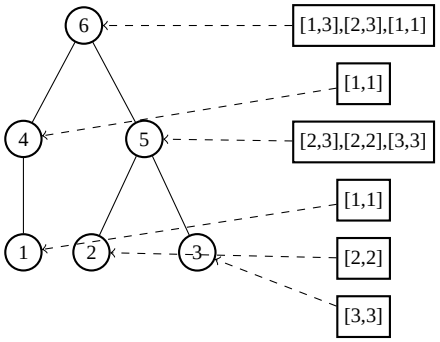

Figure 3: Constituent set of a synthetic parse tree

- 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).

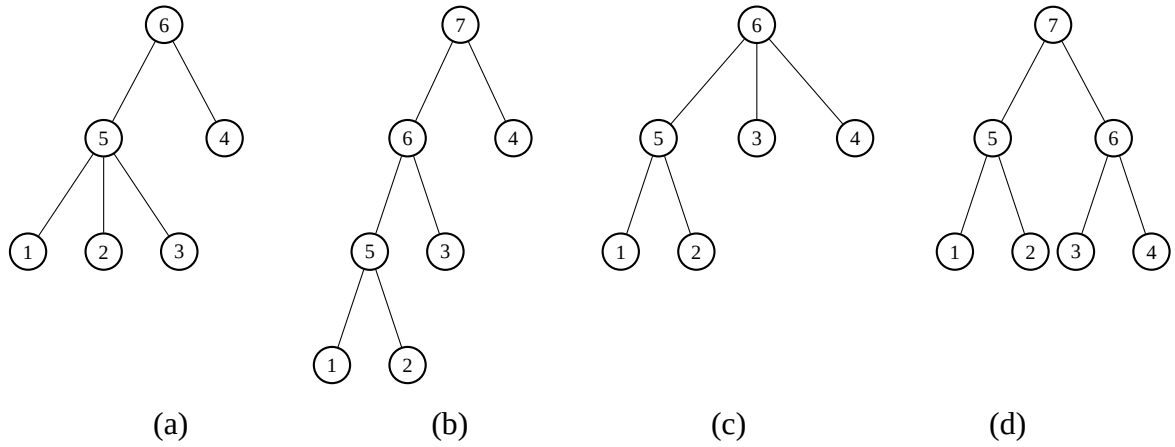

Figure 4: Illustrating example of the concept of *conflict*: (a) and (b) are compatible (not conflict); (a) conflicts with (c) (condition 1) and (d) (condition 2)

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.