Iterative Transformation of Annotation Guidelines for Constituency Parsing

Xiang Li [1, 2] **Wenbin Jiang** [1] **Yajuan Lü** [1] **Qun Liu** [1, 3]

[1]Key Laboratory of Intelligent Information Processing
Institute of Computing Technology, Chinese Academy of Sciences
{lixiang, jiangwenbin, lvyajuan}@ict.ac.cn
[2]University of Chinese Academy of Sciences
[3]Centre for Next Generation Localisation
Faculty of Engineering and Computing, Dublin City University
qliu@computing.dcu.ie

Abstract

This paper presents an effective algorithm of annotation adaptation for constituency treebanks, which transforms a treebank from one annotation guideline to another with an iterative optimization procedure, thus to build a much larger treebank to train an enhanced parser without increasing model complexity. Experiments show that the transformed Tsinghua Chinese Treebank as additional training data brings significant improvement over the baseline trained on Penn Chinese Treebank only.

1 Introduction

Annotated data have become an indispensable resource for many natural language processing (NLP) applications. On one hand, the amount of existing labeled data is not sufficient; on the other hand, however there exists multiple annotated data with incompatible annotation guidelines for the same NLP task. For example, the People's Daily corpus (Yu et al., 2001) and Chinese Penn Treebank (CTB) (Xue et al., 2005) are publicly available for Chinese segmentation.

An available treebank is a major resource for syntactic parsing. However, it is often a key bottleneck to acquire credible treebanks. Various treebanks have been constructed based on different annotation guidelines. In addition to the most popular CTB, Tsinghua Chinese Treebank (TCT) (Zhou, 2004) is another real large-scale treebank for Chinese constituent parsing. Figure 1 illustrates some differences between CTB and TCT in grammar category and syntactic structure. Unfortunately, these heterogeneous treebanks can not be directly merged together for training a parsing model. Such divergences cause a great waste of human effort. Therefore, it is highly desirable to transform a treebank into another compatible with another annotation guideline.

In this paper, we focus on harmonizing heterogeneous treebanks to improve parsing performance. We first propose an effective approach to automatic treebank transformation from one annotation guideline to another. For convenience of reference, a treebank with our desired annotation guideline is named as target treebank, and a treebank with a differtn annotation guideline is named as source treebank. Our approach proceeds in three steps. A parser is firstly trained on source treebank. It is used to relabel the raw sentences of target treebank, to acquire parallel training data with two heterogeneous annotation guidelines. Then, an annotation transformer is trained on the parallel training data to model the annotation inconsistencies. In the last step, a parser trained on target treebank is used to generate k-best parse trees with target annotation for source sentences. Then the optimal parse trees are selected by the annotation transformer. In this way, the source treebank is transformed to another with our desired annotation guideline. Then we propose an optimization strategy of iterative training to further improve the transformation performance. At each iteration, the annotation transformation of source-to-target and target-to-source are both performed. The transformed treebank is used to provide better annotation guideline for the parallel training data of next iteration. As a result, the better parallel training data will bring an improved annotation transformer at next iteration.

We perform treebank transformation from TC-

Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, pages 591–596,
Sofia, Bulgaria, August 4-9 2013.

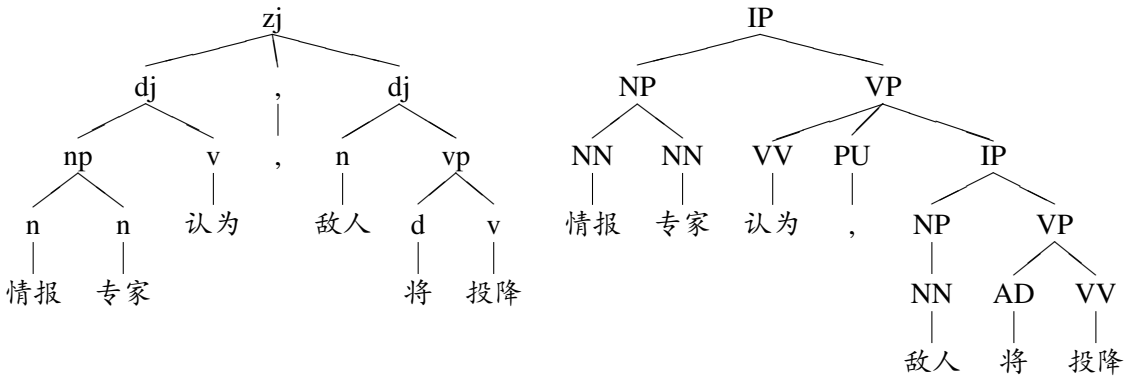

Figure 1: Example heterogeneous trees with TCT (left) and CTB (rigth) annotation guidelines.

T to CTB, in order to obtain additional treebank to improve a parser. Experiments on Chinese constituent parsing show that, the iterative training strategy outperforms the basic annotation transformation baseline. With addidional transformed treebank, the improved parser achieves an F-measure of 0.95% absolute improvement over the baseline parser trained on CTB only.

2 Automatic Annotation Transformation

In this section, we present an effective approach that transforms the source treebank to another compatible with the target annotation guideline, then describe an optimization strategy of iterative training that conducts several rounds of bidirectional annotation transformation and improves the transformation performance gradually from a global view.

2.1 Principle for Annotation Transformation

In training procedure, the source parser is used to parse the sentences in the target treebank so that there are k-best parse trees with the source annotation guideline and one gold tree with the target annotation guideline for each sentence in the target treebank. This parallel data is used to train a source-to-target tree transformer. In transformation procedure, the source k-best parse trees are first generated by a parser trained on the target treebank. Then the optimal source parse trees with target annotation are selected by the annotation transformer with the help of gold source parse trees. By combining the target treebank with the transformed source treebank, it can improve parsing accuracy using a parser trained on the enlarged treebank.

Algorithm 1 shows the training procedure of treebank annotation transformation. $treebank_s$ and $treebank_t$ denote the source and target treebank respectively. $parser_s$ denotes the source parser. $transformer_{s \to t}$ denotes the annotation transformer. $treebank_m^n$ denotes m treebank re-labeled with n annotation guideline. Function TRAIN invokes the Berkeley parser (Petrov et al., 2006; Petrov and Klein, 2007) to train the constituent parsing models. Function PARSE generates k-best parse trees. Function TRANSFORMTRAIN invokes the perceptron algorithm (Collins, 2002) to train a discriminative annotation transformer. Function TRANSFORM selects the optimal transformed parse trees with the target annotation.

2.2 Learning the Annotation Transformer

To capture the transformation information from the source treebank to the target treebank, we use the discriminative reranking technique (Charniak and Johnson, 2005; Collins and Koo, 2005) to train the annotation transformer and to score k-best parse trees with some heterogeneous features.

In this paper, the averaged perceptron algorithm is used to train the treebank transformation model. It is an online training algorithm and has been successfully used in many NLP tasks, such as parsing (Collins and Roark, 2004) and word segmentation (Zhang and Clark, 2007; Zhang and Clark, 2010).

In addition to the target features which closely follow Sun et al. (2010). We design the following quasi-synchronous features to model the annotation inconsistencies.

- **Bigram constituent relation** For two consecutive fundamental constituents s_i and s_j in the target parse tree, we find the minimum categories N_i and N_j of the spans of s_i and s_j in the source parse tree respectively. Here

Algorithm 1 Basic treebank annotation transformation.

function TRANSFORM-TRAIN($treebank_s$, $treebank_t$)
  $parser_s \leftarrow$ TRAIN($treebank_s$)
  $treebank_t^s \leftarrow$ PARSE($parser_s, treebank_t$)
  $transformer_{s \to t} \leftarrow$ TRANSFORMTRAIN($treebank_t, treebank_t^s$)
  $treebank_s^t \leftarrow$ TRANSFORM($transformer_{s \to t}, treebank_s$)
  return $treebank_s^t \cup treebank_t$

Algorithm 2 Iterative treebank annotation transformation.

function TRANSFORM-ITERTRAIN($treebank_s, treebank_t$)
  $parser_s \leftarrow$ TRAIN($treebank_s$)
  $parser_t \leftarrow$ TRAIN($treebank_t$)
  $treebank_t^s \leftarrow$ PARSE($parser_s,\ treebank_t$)
  $treebank_s^t \leftarrow$ PARSE($parser_t, treebank_s$)
  repeat
    $transformer_{s \to t} \leftarrow$ TRANSFORMTRAIN($treebank_t$,$treebank_t^s$)
    $transformer_{t \to s} \leftarrow$ TRANSFORMTRAIN($treebank_s$,$treebank_s^t$)
    $treebank_s^t \leftarrow$ TRANSFORM($transformer_{s \to t}, treebank_s$)
    $treebank_t^s \leftarrow$ TRANSFORM($transformer_{t \to s}, treebank_t$)
    $parser_t \leftarrow$ TRAIN($treebank_s^t \cup treebank_t$)
  until EVAL($parser_t$) converges
  return $treebank_s^t \cup treebank_t$

a fundamental constituent is defined to be a pair of word and its POS tag. If N_i is a sibling of N_j or each other is identical, we regard the relation between s_i and s_j as a positive feature.

- **Consistent relation** If the span of a target constituent can be also parsed as a constituent by the source parser, the combination of target rule and source category is used.
- **Inconsistent relation** If the span of a target constituent cannot be analysed as a constituent by the source parser, the combination of target rule and corresponding treelet in the source parse tree is used.
- **POS tag** The combination of POS tags of same words in the parallel data is used.

2.3 Iterative Training for Annotation Transformation

Treebank annotation transformation relies on the parallel training data. Consequently, the accuracy of source parser decides the accuracy of annotation transformer. We propose an iterative training method to improve the transformation accuracy by iteratively optimizing the parallel parse trees. At each iteration of training, the treebank transformation of source-to-target and target-to-source are both performed, and the transformed treebank provides more appropriate annotation for subsequent iteration. In turn, the annotation transformer can be improved gradually along with optimization of the parallel parse trees until convergence.

Algorithm 2 shows the overall procedure of iterative training, which terminates when the performance of a parser trained on the target treebank and the transformed treebank converges.

3 Experiments

3.1 Experimental Setup

We conduct the experiments of treebank transformation from TCT to CTB. CTB 5.1 is used as the target treebank. We follow the conventional corpus splitting of CTB 5.1: articles 001-270 and 400-1151 are used for training, articles 271-300 are used as test data and articles 301-325 are used as developing data. We use slightly modified version of CTB 5.1 by deleting all the function tags and empty categories, e.g., *OP*, using Tsurgeon (Levy and Andrew, 2006). The whole TCT 1.0 is taken as the source treebank for training the annotation transformer.

The Berkeley parsing model is trained with 5 split-merge iterations. And we run the Berkeley parser in 100-best mode and construct the 20-fold cross validation training as described in Charniak and Johnson (2005). In this way, we acquire the parallel parse trees for training the annotation transformer.

In this paper, we use bracketing $F1$ as the ParseVal metric provided by EVALB [1] for all experiments.

[1] http://nlp.cs.nyu.edu/evalb/

Model	F-Measure ($\leq$ 40 words)	F-Measure (all)
Self-training	86.11	83.81
Base Annotation Transformation	86.56	84.23
Iterative Annotation Transformation	86.75	84.37
Baseline	85.71	83.42

Table 1: The performance of treebank annotation transformation using iterative training.

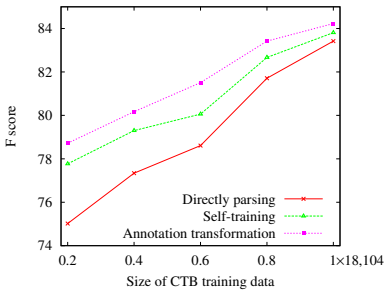

Figure 2: Parsing accuracy with different amounts of CTB training data.

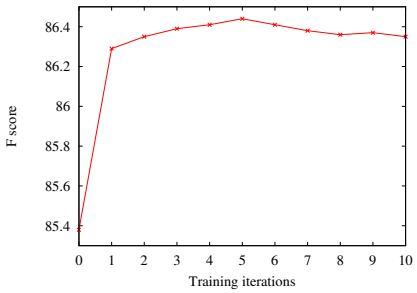

Figure 3: Learning curve of iterative transformation training.

3.2 Basic Transformation

We conduct experiments to evaluate the effect of the amount of target training data on transformation accuracy, and how much constituent parsers can benefit from our approach. An enhanced parser is trained on the CTB training data with the addition of transformed TCT by our annotation transformer. As comparison, we build a baseline system (direct parsing) using the Berkeley parser only trained on the CTB training data. In this experiment, the self-training method (McClosky et al., 2006a; McClosky et al., 2006b) is also used to build another strong baseline system, which uses unlabelled TCT as additional data. Figure 2 shows that our approach outperforms the two strong baseline systems. It achieves a 0.69% absolute improvement on the CTB test data over the direct parsing baseline when the whole CTB training data is used for training. We also can find that our approach further extends the advantage over the two baseline systems as the amount of CTB training data decreases in Figure 2. The figure confirms our approach is effective for improving parser performance, specially for the scenario where the target treebank is scarce.

3.3 Iterative Transformation

We use the iterative training method for annotation transformation. The CTB developing set is used to determine the optimal training iteration. After each iteration, we test the performance of a parser trained on the combined treebank. Figure 3 shows the performance curve with iteration ranging from 1 to 10. The performance of basic annotation transformation is also included in the curve when iteration is 1. The curve shows that the maximum performance is achieved at iteration 5. Compared to the basic annotation transformation, the iterative training strategy leads to a better parser with higher accuracy. Table 1 reports that the final optimized parsing results on the CTB test set contributes a 0.95% absolute improvement over the directly parsing baseline.

4 Related Work

Treebank transformation is an effective strategy to reuse existing annotated data. Wang et al. (1994) proposed an approach to transform a treebank into another with a different grammar using their matching metric based on the bracket information of original treebank. Jiang et al. (2009) proposed annotation adaptation in Chinese word segmentation, then, some work were done in parsing (Sun et al., 2010; Zhu et al., 2011; Sun and Wan, 2012). Recently, Jiang et al. (2012) proposed an advanced annotation transformation in Chinese word segmentation, and we extended it to the more complicated treebank annotation transformation used for Chinese constituent parsing.

Other related work has been focused on semi-supervised parsing methods which utilize labeled data to annotate unlabeled data, then use the additional annotated data to improve the original model (McClosky et al., 2006a; McClosky et

al., 2006b; Huang and Harper, 2009). The self-training methodology enlightens us on getting annotated treebank compatibal with another annotation guideline. Our approach places extra emphasis on improving the transformation performance with the help of source annotation knowledge.

Apart from constituency-to-constituency treebank transformation, there also exists some research on dependency-to-constituency treebank transformation. Collins et al. (1999) used transformed constituency treebank from Prague Dependency Treebank for constituent parsing on Czech. Xia and Palmer (2001) explored different algorithms that transform dependency structure to phrase structure. 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. Smith and Eisner (2009) and Li et al. (2012) generated rich quasi-synchronous grammar features to improve parsing performance. Some work has been done from the other direction (Daum et al., 2004; Nivre, 2006; Johansson and Nugues, 2007).

5 Conclusion

This paper propose an effective approach to transform one treebank into another with a different annotation guideline. Experiments show that our approach can effectively utilize the heterogeneous treebanks and significantly improve the state-of-the-art Chinese constituency parsing performance. How to exploit more heterogeneous knowledge to improve the transformation performance is an interesting future issue.

Acknowledgments

The authors were supported by National Natural Science Foundation of China (Contracts 61202216), National Key Technology R&D Program (No. 2012BAH39B03), and Key Project of Knowledge Innovation Program of Chinese Academy of Sciences (No. KGZD-EW-501). Qun Liu's work was partially supported by Science Foundation Ireland (Grant No.07/CE/I1142) as part of the CNGL at Dublin City University. Sincere thanks to the three anonymous reviewers for their thorough reviewing and valuable suggestions!

References

E. Charniak and M. Johnson. 2005. Coarse-to-fine n-best parsing and maxent discriminative reranking. In *Proceedings of ACL*, pages 173–180.

M. Collins and T. Koo. 2005. Discriminative reranking for natural language parsing. *Computational Linguistics*, 31(1):25–70.

M. Collins and B. Roark. 2004. Incremental parsing with the perceptron algorithm. In *Proceedings of ACL*, volume 2004.

M. Collins, L. Ramshaw, J. Hajič, and C. Tillmann. 1999. A statistical parser for czech. In *Proceedings of ACL*, pages 505–512.

M. Collins. 2002. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *Proceedings of EMNLP*, pages 1–8.

M. Daum, K. Foth, and W. Menzel. 2004. Automatic transformation of phrase treebanks to dependency trees. In *Proceedings of LREC*.

Z. Huang and M. Harper. 2009. Self-training pcfg grammars with latent annotations across languages. In *Proceedings of EMNLP*, pages 832–841.

W. Jiang, L. Huang, and Q. Liu. 2009. Automatic adaptation of annotation standards: Chinese word segmentation and pos tagging: a case study. In *Proceedings of ACL*, pages 522–530.

Wenbin Jiang, Fandong Meng, Qun Liu, and Yajuan Lü. 2012. Iterative annotation transformation with predict-self reestimation for chinese word segmentation. In *Proceedings of EMNLP*, pages 412–420.

R. Johansson and P. Nugues. 2007. Extended constituent-to-dependency conversion for english. In *Proc. of the 16th Nordic Conference on Computational Linguistics*.

R. Levy and G. Andrew. 2006. Tregex and tsurgeon: tools for querying and manipulating tree data structures. In *Proceedings of the fifth international conference on Language Resources and Evaluation*, pages 2231–2234.

Zhenghua Li, Ting Liu, and Wanxiang Che. 2012. Exploiting multiple treebanks for parsing with quasi-synchronous grammars. In *Proceedings of ACL*, pages 675–684.

D. McClosky, E. Charniak, and M. Johnson. 2006a. Effective self-training for parsing. In *Proceedings of NAACL*, pages 152–159.

D. McClosky, E. Charniak, and M. Johnson. 2006b. Reranking and self-training for parser adaptation. In *Proceedings of ACL*, pages 337–344.

Zheng-Yu Niu, Haifeng Wang, and Hua Wu. 2009. Exploiting heterogeneous treebanks for parsing. In *Proceedings of ACL*, pages 46–54.

J. Nivre. 2006. *Inductive dependency parsing*. Springer Verlag.

S. Petrov and D. Klein. 2007. Improved inference for unlexicalized parsing. In *Proceedings of NAACL*, pages 404–411.

S. Petrov, L. Barrett, R. Thibaux, and D. Klein. 2006. Learning accurate, compact, and interpretable tree annotation. In *Proceedings of ACL*, pages 433–440.

David A Smith and Jason Eisner. 2009. Parser adaptation and projection with quasi-synchronous grammar features. In *Proceedings of EMNLP*, pages 822–831.

W. Sun and X. Wan. 2012. Reducing approximation and estimation errors for chinese lexical processing with heterogeneous annotations. In *Proceedings of ACL*.

W. Sun, R. Wang, and Y. Zhang. 2010. Discriminative parse reranking for chinese with homogeneous and heterogeneous annotations. In *Proceedings of CIPS-SIGHAN*.

J.N. Wang, J.S. Chang, and K.Y. Su. 1994. An automatic treebank conversion algorithm for corpus sharing. In *Proceedings of ACL*, pages 248–254.

F. Xia and M. Palmer. 2001. Converting dependency structures to phrase structures. In *Proceedings of the first international conference on Human language technology research*, pages 1–5.

N. Xue, F. Xia, F.D. Chiou, and M. Palmer. 2005. The penn chinese treebank: Phrase structure annotation of a large corpus. *Natural Language Engineering*, 11(02):207–238.

S. Yu, J. Lu, X. Zhu, H. Duan, S. Kang, H. Sun, H. Wang, Q. Zhao, and W. Zhan. 2001. Processing norms of modern chinese corpus. *Technical Report*.

Y. Zhang and S. Clark. 2007. Chinese segmentation with a word-based perceptron algorithm. In *Proceedings of ACL*, pages 840–847.

Y. Zhang and S. Clark. 2010. A fast decoder for joint word segmentation and pos-tagging using a single discriminative model. In *Proceedings of EMNLP*, pages 843–852.

Q. Zhou. 2004. Annotation scheme for chinese treebank. *Journal of Chinese Information Processing*, 18(4).

M. Zhu, J. Zhu, and M. Hu. 2011. Better automatic treebank conversion using a feature-based approach. In *Proceedings of ACL*, pages 715–719.