Learning Grammar with Explicit Annotations for Subordinating Conjunctions

Dongchen Li, Xiantao Zhang and Xihong Wu
Key Laboratory of Machine Perception and Intelligence
Speech and Hearing Research Center
Peking University, Beijing, China
{lidc,zhangxt,wxh}@cis.pku.edu.cn

Abstract

Data-driven approach for parsing may suffer from data sparsity when entirely unsupervised. External knowledge has been shown to be an effective way to alleviate this problem. Subordinating conjunctions impose important constraints on Chinese syntactic structures. This paper proposes a method to develop a grammar with hierarchical category knowledge of subordinating conjunctions as explicit annotations. Firstly, each part-of-speech tag of the subordinating conjunctions is annotated with the most general category in the hierarchical knowledge. Those categories are human-defined to represent distinct syntactic constraints, and provide an appropriate starting point for splitting. Secondly, based on the data-driven state-split approach, we establish a mapping from each automatic refined subcategory to the one in the hierarchical knowledge. Then the data-driven splitting of these categories is restricted by the knowledge to avoid over refinement. Experiments demonstrate that constraining the grammar learning by the hierarchical knowledge improves parsing performance significantly over the baseline.

1 Introduction

Probabilistic context-free grammars (PCFGs) underlie most of the high-performance parsers (Collins, 1999; Charniak, 2000; Charniak and Johnson, 2005; Zhang and Clark, 2009; Chen and Kit, 2012; Zhang et al., 2013). However, a naive PCFG which simply takes the empirical rules and probabilities off of a Treebank does not perform well (Klein and Manning, 2003; Levy and Manning, 2003; Bansal and Klein, 2012), because its context-freedom assumptions are too strong in some cases (e.g. it assumes that subject and object NPs share the same distribution). Therefore, a variety of techniques have been developed to enrich PCFG (Klein and Manning, 2005; Matsuzaki et al., 2005; Zhang and Clark, 2011; Shindo et al., 2012).

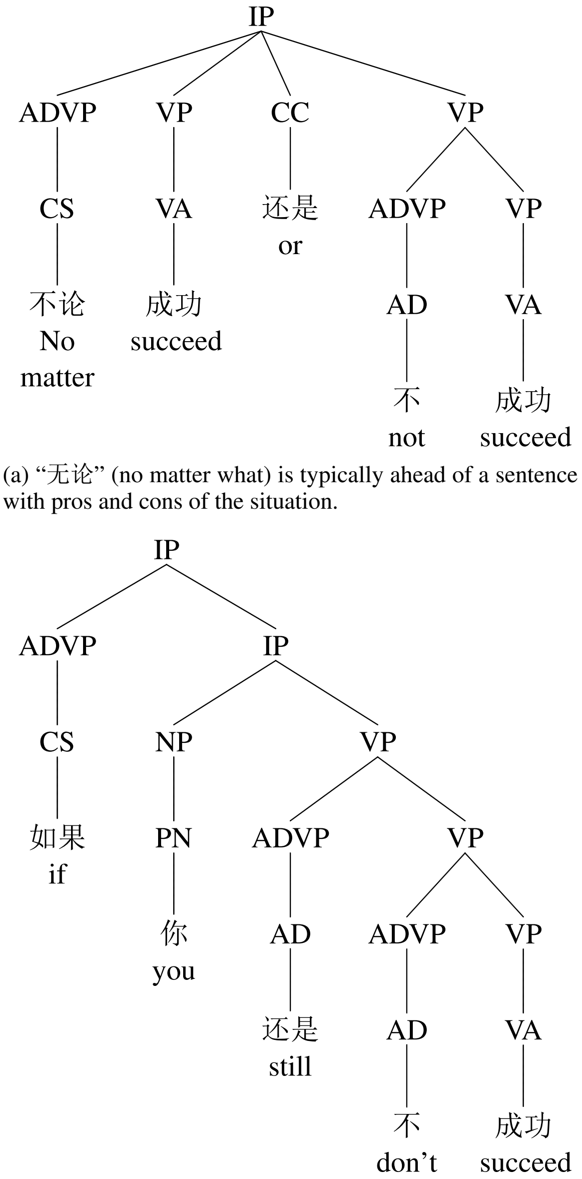
Hierarchical state-split approach (Petrov et al., 2006; Petrov and Klein, 2007; Petrov and Klein, 2008a; Petrov and Klein, 2008b; Petrov, 2009) refines and generalizes the original grammars in a data-driven manner, and achieves state-of-the-art performance. Starting from a completely markovized X-Bar grammar, each category is split into two subcategories. EM is initialized with this starting point and used to climb the highly non-convex objective function of computing the joint likelihood of the observed parse trees. Then a merging step applies a likelihood ratio test to reverse the least useful half part of the splits. Learning proceeds by iterating between those two steps for six rounds. Spectral learning of latent-variable PCFGs (Cohen et al., 2012; Bailly et al., ; Cohen et al., 2013b; Cohen et al., 2013a) is another effective manner of state-split approach that provides accurate and consistent parameter estimates. However, this two complete data-driven approaches are likely to be hindered by the overfitting issue.

Incorporating knowledge (Zhang et al., 2013; Wu et al., 2011) to refine the categories in training a parser has been proved to remedy the weaknesses of probabilistic context-free grammar (PCFG). The knowledge contains content words semantic resources base (Fujita et al., 2010; Agirre et al., 2008; Lin et al., 2009), named entity cues (Li et al., 2013) and so on. However, they are limited in that they do not take into account the knowledge about subordinating conjunctions.

Subordinating conjunctions are important indications for different syntactic structure, espe-

Proceedings of the ACL 2014 Student Research Workshop, pages 48–55,
Baltimore, Maryland USA, June 22-27 2014. ©2014 Association for Computational Linguistics

cially for Chinese. For example, the subordinating conjunction “无论” (no matter what) is typically ahead of a sentence with pros and cons of the situation; on the contrary, a sufficient condition often occurs after the subordinating conjunction “如果” (if). Those two cases are of distinct syntactic structure. Figure 1 demonstrates that although the sequences of the part-of-speech of the input words are similar, these two subordinating conjunctions exert quite different syntactic constraints to the following clauses.



(a) “无论” (no matter what) is typically ahead of a sentence with pros and cons of the situation.

(b) “如果” (if) often precedes a sufficient condition.

Figure 1: Different types of subordinating conjunctions indicate distinct syntactic structure.

Based on the hierarchical state-split approach, this paper proposes a data-oriented model supervised by our hierarchical subcategories of subordinating conjunctions. In order to constrain the automatic subcategory refinement, we firstly establish the mapping between the automatic clustered subcategories and the predefined subcategories. Then we employ a knowledge criterion to supervise the hierarchical splitting of these subordinating conjunction subcategories by the automatic state-split approach, which can alleviate over-fitting. The experiments are carried out on Penn Chinese Treebank and Tsinghua Treebank, which verify that the refined grammars with refined subordinating conjunction categories can improve parsing performance significantly.

The rest of this paper is organized as follows. We first describe our hierarchical subcategories of subordinating conjunction. Section 3 illustrates the constrained grammar learning process in details. Section 4 presents the experimental evaluation and the comparison with other approaches.

2 Hierarchical Subcategories of Subordinating Conjunction

The only tag “CS” for all the various subordinating conjunctions is too coarse to indicate the intricate subordinating relationship. The words indicating different grammatical features share the same tag “CS”, such as transition relationship, progression relationship, preference relationship, purpose relationship and condition relationship. In each case, the context is different, and the subordinating conjunction is an obvious indication for the parse disambiguation for the context. The existing resources for computational linguistic, like HowNet (Dong and Dong, 2003) and Cilin (Mei et al., 1983), have classified all subordinating conjunctions as one category, which is too coarse to capture the syntactic implication.

To make use of the indication, we subdivide the subordinating conjunctions according to its grammatical features in our scheme. Subordinating conjunctions indicating each relationship is further subdivided into two subcategories: one is used before the principal clause, the other is before the subordinate clause. For example, the conjunctions representing cause and effect contains “because” and “so”, where “because” should modify the cause, and “so” should modify the effect. In addition, we found that there are several cases in the conditional clause. Accordingly, we subdivide the conditional subordinating conjunctions into seven types: assumption, universalization,

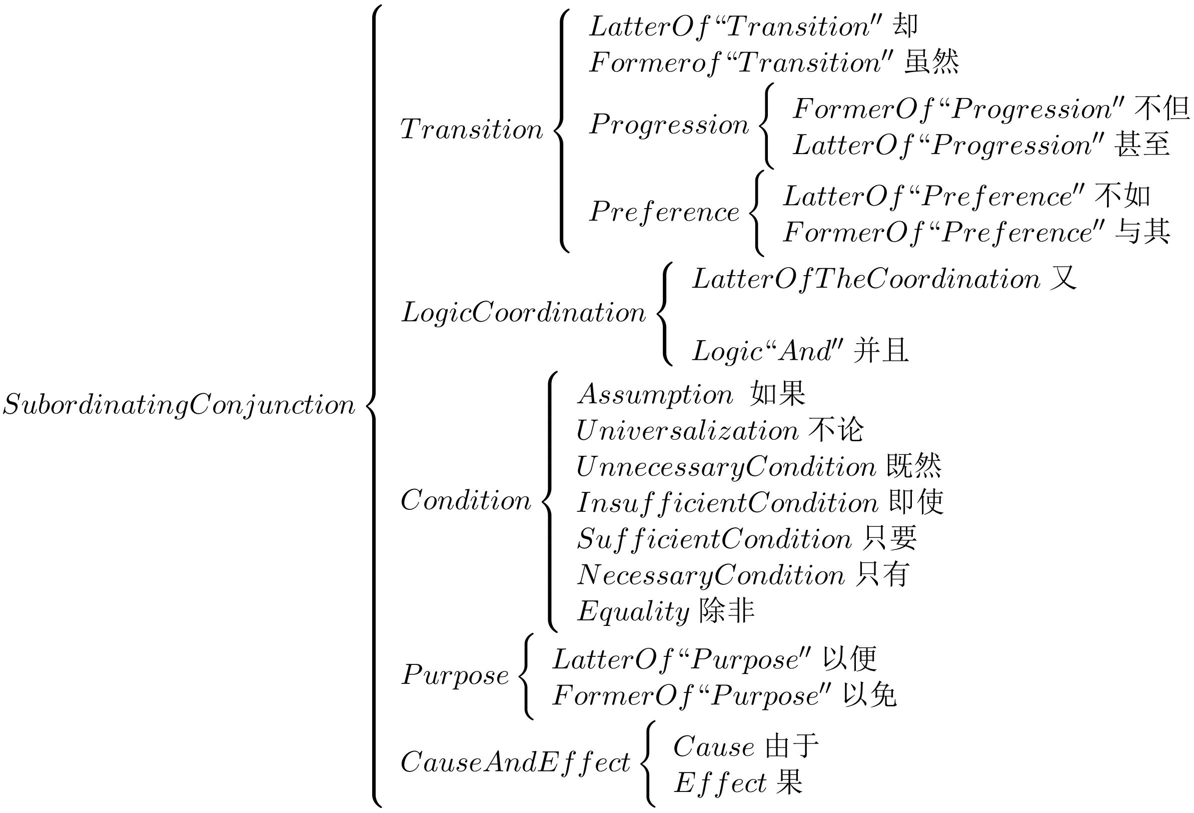


Figure 2: Hierarchical subcategories of subordinating conjunctions with examples.

equality, sufficient condition, necessary condition, sufficient but unnecessary condition and necessary but insufficient condition (concession). The detailed hierarchical subcategories of subordinating conjunctions are displayed in Figure 2.

3 Parsing with Hierarchical Categories

The automatic state-split approach is designed to refine all symbols together through a data-driven manner, which takes the over-fitting risk. Instead of splitting and merging all symbols together automatically, we employ a knowledge-based criterion with hierarchical refinement knowledge to constraint the splitting of these new refined tags for subordinating conjunctions.

At the beginning, we produce a good starting annotation with the top subcategories in the hierarchical subcategories, which is of great use to constraining the automatic splitting process. As demonstrated in Figure 4, our parser is trained on the good initialization with the automatic hierarchical state-split process, and gets improvements compared with the original training data. For example, as shown in Figure 2, the category for 却(but) and "Cause" for 由于(because) is annotated as the top category "Transition" and "Cause And Effect" respectively.

However, during this process, only the most general hypernyms are used as the semantic representation of words, and the lower subcategory knowledge in the hierarchy is not explored. Thus, we further constraint the split of the subordinating conjunctions subcategories to be consistent with the hierarchical subcategories to alleviate the overfitting issue. The top class is only used as the starting annotations of POS tags to reduce the search space for EM in our method. It is followed by the hierarchical state-split process to further refine the starting annotations based on the hierarchical subcategories.

3.1 Mapping from Automatic Subcategories to Predefined Subcategories

With the initialization proposed above, the automatically split-merge approach produces a series of refined categories for each tag. We restrict each automatically refined subcategory of subordinating conjunctions to correspond to a special node

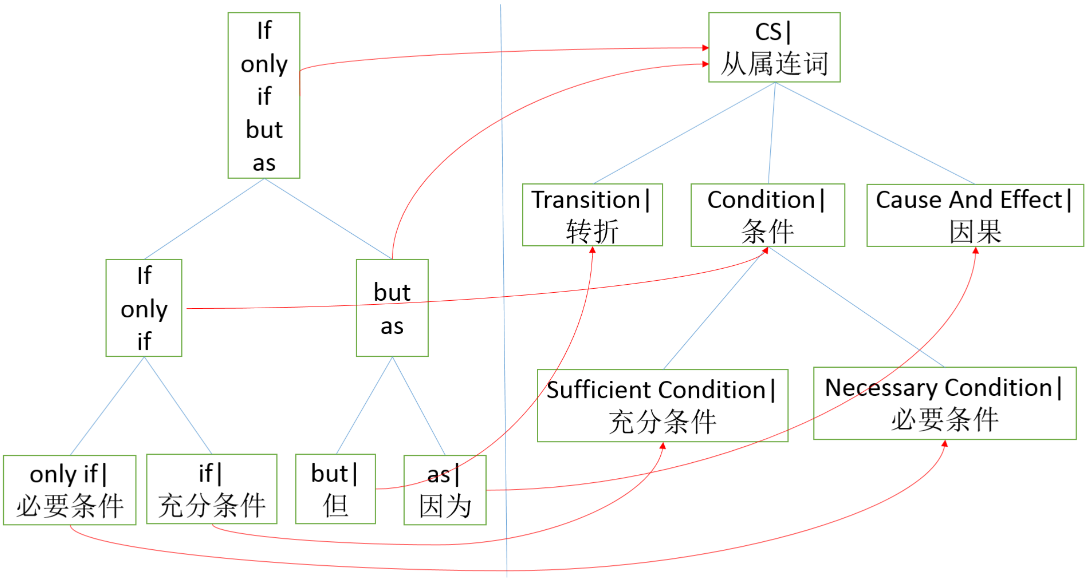


Figure 3: A schematic figure for the hierarchical state-split process of the tag "CS". Each subcategory of this tag has its own word set, and corresponds to one layer at the appropriate level in the hierarchical subcategories.

in the hierarchical subcategories, as a hyponym of "CS". The hierarchical subcategories are employed in the hierarchical state-split process to impose restrictions on the subcategory refinement.

First of all, it is necessary to establish the mapping from each subcategory in the data-driven hierarchical subcategories to the subcategory in the predefined hierarchical subcategories. We transfer the method for semantic-related labels (Lin et al., 2009) to our case here. The mapping is implemented with the word set related to each automatically refined granularity of clustered subordinating conjunctions and the node at the special level in the subcategory knowledge. The schematic in Figure 3 demonstrates this supervised splitting process for CS. The left part of this figure is the word sets of automatic clustered subcategories of the CS, which is split hierarchically. As expressed by the lines, each subcategory corresponds to one node in the right part of this figure, which is our hierarchical subcategory knowledge of subordinating conjunctions.

As it is shown in Figure 3, the original tag "CS" treats all the words it produces as its word set. Upon splitting each coarse category into two more specific subcategories, its word set is also cut into two subsets accordingly, through forcedly dividing each word in the word set into one subcategory which is most probable for this word in the lexical grammar. And each automatic refinement is mapped to the most specific subcategory (that is to say, the lowest node) that contains the entirely corresponding word set in the human-defined knowledge. On this basis, the new knowledge-based criterion is introduced to enrich and generalize these subcategories, with the purpose of fitting the refinement to the subcategory knowledge rather than the training data.

3.2 Knowledge-based Criterion for Subordinating Conjunctions Refinement

With the mapping between the automatic refined subcategories and the human-defined hierarchical subcategory knowledge, we could supervise the automatic state refinement by the knowledge.

Instead of being merged by likelihood, a knowledge-based criterion is employed, to decide whether or not to go back to the upper layer in the hierarchical subcategories and thus remove the new subcategories of these tags. The criterion is that, we assume that the bottom layer in the hierarchical subcategories is special enough to express the distinction of the subordinating conjunctions. If the subcategories of the subordinating conjunctions has gone beyond the bottom layer, then the new split subcategories are deemed to be unnecessary and should be merged back. That is to say, once the parent layer of this new subcategory is mapped onto the most special subcategory, it should be removed immediately. As illustrated

Treebank	Train Dataset	Develop Dataset	Test Dataset
CTB5	Articles 1-270	Articles 400-1151, 301-325	Articles 271-300
TCT	16000 sentences	800 sentences	758 sentences

Table 1: Data allocation of our experiment.

in Figure 3, if the node has no hyponym, this subcategory has been specialized enough according to the knowledge, and thus the corresponding subcategory will stop splitting.

By introducing a knowledge-based criterion, the issue is settled whether or not to further split subcategories from the perspective of predefined knowledge. To investigate the effectiveness of the presented approach, several experiments are conducted on both Penn Chinese Treebank and Tsinghua Treebank. They reveal that the subcategory knowledge of subordinating conjunctions is effective for parsing.

4 Experiments

4.1 Experimental Setup

We present experimental results on both Chinese Treebank (CTB) 5.0 (Xue et al., 2002) (All traces and functional tags were stripped.) and Tsinghua Treebank (TCT) (Zhou, 2004). All the experiments were carried out after six cycles of split-merge.

The data set allocation is described in Table 1. We use the EVALB parseval reference implementation (Sekine, 1997) for scoring. Statistical significance was checked by Bikel's randomized parsing evaluation comparator (Bikel, 2000).

4.2 Parsing Performance with Hierarchical Subcategories

We presented a flexible approach which refines the subordinating conjunctions in a hierarchy fashion where the hierarchical layers provide different granularity of specificity. To facilitate the comparisons, we set up 6 experiments on CTB5.0 with different strategies of choosing the subcategory layers in the hierarchical subcategory knowledge:

- baseline: Training without hierarchical subcategory knowledge
- top: Choosing the top layer in hierarchical subcategories (using "Transition", "Condition" , "Purpose" and so on)
- bottom: Choosing the bottom layer in hierarchical subcategories (the most specified subcategories)
- word: Substituting POS tag with the word itself
- knowledge criterion: Automatically choosing the appropriate layer through the knowledge criterion

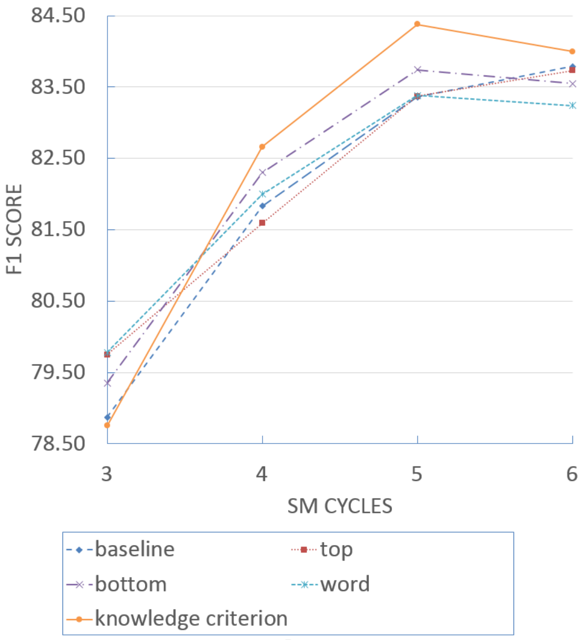


Figure 4: Comparison of parsing performance for each model in the split-merge cycles.

Figure 4 shows the F_1 scores of the last 4 cycles in the 6 split-merge cycles. The results are just as expectation, through which we can tell that the "top" model performs slightly better than the baseline owing to a better start point of the state-splitting. This result confirms the value of our initial explicit annotations. While the "bottom" model doesn't improve the performance due to excessive refinement and causes over-fitting, the "word" model behaves even worse for the same reason. In the 5th split-merge cycle, the "knowledge criterion" model picks the appropriate layer in hierarchical subcategories and achieves the best result.

We also test our method on TCT. Table 2 compares the accuracies of the baseline, initialization with top subcategories and the “knowledge criterion” model, and confirms that the subcategory knowledge helps parse disambiguation.

Parser	P	R	F_1
baseline	74.40	74.28	74.34
top	75.12	75.17	75.14
knowledge criterion	76.18	76.27	76.22

Table 2: Our parsing performance with different criterions on TCT.

4.3 Final Results

Our final results are achieved using the “knowledge criterion” model. As we can see from the table 3, our final parsing performance is higher than the unlexicalized parser (Levy and Manning, 2003; Petrov, 2009) and the parsing system in Qian and Liu (2012), but falls short of the systems using semantic knowledge of Lin et al. (2009) and exhaustive word formation knowledge of Zhang et al. (2013).

Parser	P	R	F_1
Levy(2003)	78.40	79.20	78.80
Petrov(2009)	84.82	81.93	83.33
Qian(2012)	84.57	83.68	84.13
Zhang(2013)	84.42	84.43	84.43
Lin(2009)	86.00	83.10	84.50
This paper	85.93	82.87	84.32

Table 3: Our final parsing performance compared with the best previous works on CTB5.0.

The improvement on the hierarchical state-split approach verifies the effectiveness of the subcategory knowledge of subordinating conjunctions for alleviating over-fitting. And the subcategory knowledge could be integrated with the knowledge base employed in Lin et al. (2009) and Zhang et al. (2013) to contribute more on parsing accuracy improvement.

5 Conclusion

In this paper, we present an approach to constrain the data-driven state-split method by hierarchical subcategories of subordinating conjunctions, which appear as explicit annotations in the grammar. The parsing accuracy is improved by this method owing to two reasons. Firstly, the most general hypernym of subordinating conjunctions exerts an initial restrict to the following splitting step. Secondly, the splitting process is confined by a knowledge-based criterion with the human-defined hierarchical subcategories to avoid over refinement.

Acknowledgments

We thank Baidu for travel and conference support for this paper. We thank Meng Zhang and Dingsheng Luo for their valuable advice. This work was supported in part by the National Basic Research Program of China (973 Program) under grant 2013CB329304, the Research Special Fund for Public Welfare Industry of Health under grant 201202001, the Key National Social Science Foundation of China under grant 12&ZD119, the National Natural Science Foundation of China under grant 91120001.

References

Eneko Agirre, Timothy Baldwin, and David Martinez. 2008. Improving parsing and pp attachment performance with sense information. *Proceedings of ACL-08: HLT*, pages 317–325.

Raphaël Bailly, Xavier Carreras Pérez, Franco M Luque, and Ariadna Julieta Quattoni. Unsupervised spectral learning of wcfg as low-rank matrix completion. Association for Computational Linguistics.

Mohit Bansal and Daniel Klein. 2012. An all-fragments grammar for simple and accurate parsing. Technical report, DTIC Document.

Bikel. 2000. Dan bikel’s randomized parsing evaluation comparator. In *http://www.cis.upenn.edu/dbikel/software.html.*

Eugene Charniak and Mark Johnson. 2005. Coarse-to-fine n-best parsing and maxent discriminative reranking. In *Proceedings of the 43rd annual meeting on Association for Computational Linguistics*, pages 173–180. Association for Computational Linguistics.

Eugene Charniak. 2000. A maximum-entropy-inspired parser. In *Proceedings of the 1st North*

American chapter of the association for computational Linguistics conference, pages 132–139. Association for Computational Linguistics.

Xiao Chen and Chunyu Kit. 2012. Higher-order constituent parsing and parser combination. In *Proceedings of the 50th annual meeting of the Association for Computational Linguistics: Short papers-Volume 2*, pages 1–5. Association for Computational Linguistics.

Shay B Cohen, Karl Stratos, Michael Collins, Dean P Foster, and Lyle Ungar. 2012. Spectral learning of latent-variable pcfgs. In *Proceedings of the 50th annual meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 223–231. Association for Computational Linguistics.

Shay B Cohen, Giorgio Satta, and Michael Collins. 2013a. Approximate pcfg parsing using tensor decomposition. In *Proceedings of NAACL-HLT*, pages 487–496.

Shay B Cohen, Karl Stratos, Michael Collins, Dean P Foster, and Lyle Ungar. 2013b. Experiments with spectral learning of latent-variable pcfgs. In *Proceedings of NAACL-HLT*, pages 148–157.

Michael Collins. 1999. *Head-driven statistical models for natural language parsing*. Ph.D. thesis, University of Pennsylvania.

Zhendong Dong and Qiang Dong. 2003. Hownet-a hybrid language and knowledge resource. In *Proceedings of the international conference on natural language processing and knowledge engineering*, pages 820–824. IEEE.

Sanae Fujita, Francis Bond, Stephan Oepen, and Takaaki Tanaka. 2010. Exploiting semantic information for hpsg parse selection. *Research on language and computation*, 8(1):1–22.

Dan Klein and Christopher D Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st annual meeting on Association for Computational Linguistics-Volume 1*, pages 423–430. Association for Computational Linguistics.

Dan Klein and Christopher D Manning. 2005. Parsing and hypergraphs. In *New developments in parsing technology*, pages 351–372. Springer.

Roger Levy and Christopher D Manning. 2003. Is it harder to parse chinese, or the chinese treebank? In *Proceedings of the 41st annual meeting on Association for Computational Linguistics-Volume 1*, pages 439–446. Association for Computational Linguistics.

Dongchen Li, Xiantao Zhang, and Xihong Wu. 2013. Improved chinese parsing using named entity cue. In *Proceeding of the 13th international conference on parsing technology*, pages 45–53.

Xiaojun Lin, Yang Fan, Meng Zhang, Xihong Wu, and Huisheng Chi. 2009. Refining grammars for parsing with hierarchical semantic knowledge. In *Proceedings of the 2009 conference on empirical methods in natural language processing: Volume 3-Volume 3*, pages 1298–1307. Association for Computational Linguistics.

Takuya Matsuzaki, Yusuke Miyao, and Jun'ichi Tsujii. 2005. Probabilistic cfg with latent annotations. In *Proceedings of the 43rd annual meeting on Association for Computational Linguistics*, pages 75–82. Association for Computational Linguistics.

Jia-Ju Mei, YM Li, YQ Gao, et al. 1983. Chinese thesaurus (tong-yi-ci-ci-lin).

Slav Petrov and Dan Klein. 2007. Improved inference for unlexicalized parsing. In *Human language technologies 2007: the conference of the North American chapter of the Association for Computational Linguistics*, pages 404–411.

Slav Petrov and Dan Klein. 2008a. Discriminative log-linear grammars with latent variables. *Advances in neural information processing systems*, 20:1153–1160.

Slav Petrov and Dan Klein. 2008b. Sparse multi-scale grammars for discriminative latent variable parsing. In *Proceedings of the conference on empirical methods in natural language processing*, pages 867–876. Association for Computational Linguistics.

Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. 2006. Learning accurate, compact, and interpretable tree annotation. In *Proceedings of the 21st international conference on computational linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 433–440. Association for Computational Linguistics.

Slav Orlinov Petrov. 2009. *Coarse-to-Fine natural language processing*. Ph.D. thesis, University of California.

Xian Qian and Yang Liu. 2012. Joint chinese word segmentation, pos tagging and parsing. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 501–511. Association for Computational Linguistics.

Collins Sekine. 1997. Evalb bracket scoring program. In *http://nlp.cs.nyu.edu/evalb/*.

Hiroyuki Shindo, Yusuke Miyao, Akinori Fujino, and Masaaki Nagata. 2012. Bayesian symbol-refined tree substitution grammars for syntactic parsing. In *Proceedings of the 50th annual meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 440–448. Association for Computational Linguistics.

Xihong Wu, Meng Zhang, and Xiaojun Lin. 2011. Parsing-based chinese word segmentation integrating morphological and syntactic information. In *Proceedings of 7th international conference on natural language processing and knowledge engineering (NLP-KE)*, pages 114–121. IEEE.

Nianwen Xue, Fu-Dong Chiou, and Martha Palmer. 2002. Building a large-scale annotated chinese corpus. In *Proceedings of the 19th international conference on computational linguistics-Volume 1*, pages 1–8. Association for Computational Linguistics.

Yue Zhang and Stephen Clark. 2009. Transition-based parsing of the chinese treebank using a global discriminative model. In *Proceedings of the 11th International Conference on Parsing Technologies*, pages 162–171. Association for Computational Linguistics.

Yue Zhang and Stephen Clark. 2011. Syntactic processing using the generalized perceptron and beam search. *Computational linguistics*, 37(1):105–151.

Meishan Zhang, Yue Zhang, Wanxiang Che, and Ting Liu. 2013. Chinese parsing exploiting characters. *51st annual meeting of the Association for Computational Linguistics*.

Qiang Zhou. 2004. Annotation scheme for chinese treebank. *Journal of Chinese information processing*, 18(4):1–8.