

# Learning the Taxonomy of Function Words for Parsing

Dongchen Li, Xiantao Zhang, Dingsheng Luo and Xihong Wu

Key Laboratory of Machine Perception and Intelligence  
Speech and Hearing Research Center  
Peking University, Beijing, China  
{lidc, zhangxt, dsluo, wxh}@cis.pku.edu.cn

## Abstract

Completely data-driven grammar training is prone to over-fitting. Human-defined word class knowledge is useful to address this issue. However, the manual word class taxonomy may be unreliable and irrational for statistical natural language processing, aside from its insufficient linguistic phenomena coverage and domain adaptivity. In this paper, a formalized representation of function word subcategorization is developed for parsing in an automatic manner. The function word classification representing intrinsic features of syntactic usages is used to supervise the grammar induction, and the structure of the taxonomy is learned simultaneously. The grammar learning process is no longer a unilaterally supervised training by hierarchical knowledge, but an interactive process between the knowledge structure learning and the grammar training. The established taxonomy implies the stochastic significance of the diversified syntactic features. The experiments on both Penn Chinese Treebank and Tsinghua Treebank show that the proposed method improves parsing performance by 1.6% and 7.6% respectively over the baseline.

## 1 Introduction

Probabilistic context-free grammar (PCFG) is widely used in the fields of speech recognition, machine translation, information retrieval, etc. It takes the empirical rules and probabilities from a Treebank. However, due to the context-free assumption, PCFG does not always perform well (Klein and Manning, 2003). For instance, it assumes adverbs, including temporal adverbs, degree adverbs and negation adverbs, to share the same distribution, whereas the distinction would provide useful indication for disambiguating the syntactic structure of the context.

It arose that the manual word classification in linguistic research was used to enrich PCFG and improve the performance. However, from the point of view of statistical natural language processing, there are some drawbacks for manual classification. Firstly, Linguistic phenomena covered by the manual refinement may be limited by the linguistic observations of human. Secondly, the evidence of manual refinement is often based on a particular corpus or specific sources of knowledge acquisition. As a result, its adaptivity to different domains or genres may be insufficient. As for function words, due to the ambiguity and complexity in syntactic grammar, it is more difficult to develop formalized representation than for content words. There are diversified standards for grammar refinement. Consequently, the word classification or category refinement can be conducted in distinct manners, while each of them is reasonable in some sense. A delicate hierarchical classification inevitably involves in multiple dividing standards. However, the word sets under distinct dividing standards may be overlapping. The problems come up that how to choose the set of the multiple standards to cooperate to build the taxonomy, and how to decide the priority of each standard. Regarding that the manual method is hard to overcome critical issues, manual taxonomy for function words may not be reliable for statistical natural language processing.

This article attempts to address these issues in a data-driven manner. we first manually construct a cursory and flat classification of function words. A hierarchical split-merge approach is employed to

---

This work is licenced under a Creative Commons Attribution 4.0 International License. Page numbers and proceedings footer are added by the organizers. License details: <http://creativecommons.org/licenses/by/4.0/>

introduce our classification, and the PCFG training procedure is supervised to alleviate the over-fitting issue. The priorities of the subcategorization standards are determined by the measurement of effectiveness for parsing in a greedy manner in the hierarchical classification. And the hierarchical structure of the classification is learned by data-driven approach in the course of grammar induction, so as to fit the practical usages in the Treebank. Accordingly, the grammar learning process is no longer a unilaterally supervised training by hierarchical knowledge, but an interactive process between the knowledge representation induction and the grammar training. That is, the grammar induction is supervised by the knowledge and the structure of the taxonomy is learned simultaneously. These two processes are iterated for several rounds and the hierarchical structure of the function word taxonomy is constructed. In each round, the induced grammar could benefit from the optimized taxonomy during the learning process. The category split in the early rounds take more priorities than in the late ones. Thus, the learned taxonomy implies the stochastic significance of the series of the syntactic features.

Experiments on Penn Chinese Treebank Fifth Edition (CTB5.0) (Xue et al., 2002) and Tsinghua Chinese Treebank (TCT) (Zhou, 2004) are performed. The results show that the induced grammars with refined conjunction categories gain parsing performance improvement by 1.6% on CTB and by 7.6% on TCT. During the training process, a taxonomy of function words is learned, which reflects their practical usages in the corpus.

The rest of this paper is organized as follows. We first review related work on category refinement for parsing. Then we describe our manually defined categories of function words in Section 3. The hierarchical state-split approach for introducing the the categories are presented in Section 4, and our taxonomy learning method is described in Section 5. In Section 7, experimental comparison is conducted among various methods on granularity choosing. And conclusions of this research are drawn in last section.

## 2 Related Work

A variety of techniques have been proposed to enrich PCFG in either manual (Klein and Manning, 2003; Zhang and Clark, 2011) or automatic (Petrov, 2009; Cohen et al., 2012) manner.

### 2.1 Automatic Refinement of Function Words for Parsing

One way of grammar refinement is data-driven state-split methods (Matsuzaki et al., 2005; Prescher, 2005). The part-of-speech and syntactic tags in the grammar are automatically split to encode the kinds of linguistic distinctions exhibited in the Treebank. The hierarchical state-split approach (Petrov et al., 2006) started from a bare-bones Treebank derived grammar, and iteratively refined it in a split-merge-smooth cycle with the EM-based parameter re-estimation. It achieved state of the art accuracies for many languages including English, Chinese and German.

One tag is usually heterogeneous, in the sense that its word set can be of multiple different types. Nevertheless, the automatic process tries to split the tags through a greedy data-driven manner, where multiple distinctive information is used simultaneously when dividing tags. Thus the refined tags are not intuitively interpretable. Meanwhile, considering that the EM algorithm usually gets stuck at a sub-optimal configuration, this data-driven method suffers from the risk of over-fitting. As shown in their experiments, there is little to be gained from splitting the closed part-of-speech classes (e.g. DT, CC, IN) or the nonterminal ADJP.

To alleviate the risk of over-fitting, we employ the human-defined knowledge to constrain the splitting process in this research. Based on the state-split model, our approach aims to reach a compromise between manual and automatic refinement approaches.

### 2.2 Manual Refinement of Function Words for Parsing

The other way to refine the annotation for training a parser is incorporating knowledge base. Semantic knowledge of content words has been proved to be effective in alleviate the data sparsity. Some researches utilized semantic knowledge in WordNet (Miller, 1995; Fellbaum, 1999) for English parsing (Fujita et al., 2010; Agirre et al., 2008), and Xiong et al. (2005; Lin et al. (2009) improved Chinese pars-

ing by incorporating semantic knowledge in HowNet (Dong and Dong, 2003; Dong and Dong, 2006). While WordNet and Hownet contain word classification for content words, Li et al. (2014b; Li et al. (2014a) have focused on exploiting manual classification for conjunction in parsing.

Klein and Manning (2003) examined the annotation in Penn English Treebank, manually split the majority of the part-of-speech (POS) tags. For the function words, they split the tag “IN” into subordinating conjunctions, complementizers and prepositions, and appended  $\hat{B}E$  to all forms of “be” and  $\hat{H}AVE$  to all forms of “have”. Conjunction tags are also marked to indicate whether they were “But”, “but” or “&”. The experimental results showed that the split tags of function words surprisingly make much contribution to the overall improved parsing accuracy. Levy and Manning (2003) transferred this work to Penn Chinese Treebank. They found that, in some cases, certain adverbs such as “however (然而)” and “especially (尤其是)” preferred IP modification and could help disambiguate IP coordination from VP coordination. To capture this point, they marked those adverbs possessing an IP grandparent. However, these manual refinement methods seems to split the tags in a rough way, which might account for a modest accuracy achieved. Some existing work used heuristic rules to simply split the tags of function words (Klein and Manning, 2003; Levy and Manning, 2003). They demonstrated that many function words stood out to be helpful in predicting the syntactic structure and syntactic label.

### 3 Manual Tabular Subcategories of Function Words

When subcategorizing function words, in this section, we manually list various grammatical distinctions that are commonly made in traditional and generative grammar in a fairly flat taxonomy. The grammar training procedure learns by using our manual taxonomy as a starting point, and constructs a reasonable and subtle hierarchical structure based on the distribution of function words usages in the corpus.

Based on some existing knowledge base (Xia, 2000; Xue et al., 2000; Zhu et al., 1995; Wang and Yu, 2003) and previous research work (Li et al., 2014b), we investigate and summarize the usage of function words, and come up with a hierarchical subcategories. The taxonomy of the function words is represented in a tree structure, where each subcategory of a function word corresponds to a node in the taxonomy, the nonterminals are subcategories and the terminals are words.

For the convenience and consistence, our manual classification just gives a rough and broad taxonomy. It is labor-intensive and error-prone of classifying the function words manually to produce a consistent output. Fine-grained hierarchical structure is not obligatory, but would be harmful if inappropriately classified, as it may mislead the learning process. To avoid this kind of risk, the elaboration is saved, rather than introducing unnecessary bias. The learning process would perform the hierarchical classification according to the distribution in the corpus.

For instance, the distinction within conjunctions is intricate. Conjunctions are the words that are called “connective words” in traditional Chinese grammar books. In Penn Chinese Treebank, they are tagged as coordinating conjunctions (CC), subordinating conjunctions (CS), or adverbs (AD) according to their syntactic distribution. CC conjoins two equivalent constituents (noun phrases, clauses, etc.), each of which has approximately the same function as the whole construction. CS precedes a subordinating clause, whereas conjunctive adverbs often appear in the main clause and pair with a subordinating conjunction (e.g., if (如果)/CS ... then (就)/AD). However, in Chinese, it is often hard to tell the subordinating clause from the main clause in the compound statement. As a result, in the prospective of linguistic computing, the confusion is that, CS and conjunctive adverbs both precedes the subordinating clauses or main clauses, while CC connects two phrases or precedes the main clause. In our scheme, we simply conflates the CC, CS and conjunctive adverbs together. This result in a general “conjunction” category, within which we just enumerate all the possible uses of the conjunctions. As a result, the structure of our human-defined taxonomy is fairly flat, as briefly shown in Figure 1 and Figure 2. Our scheme releases our hands from the confusing situations, by leaving them to our data-driven method described in the following section. Figure 1 and Figure 2 abbreviate the manual classification and their corresponding examples.

Many prepositions in Chinese are evolved from verbs, thus the linguistic characteristics of prepositions are somewhat similar to verbs. Therefore, this paper divides the preposition word set according to

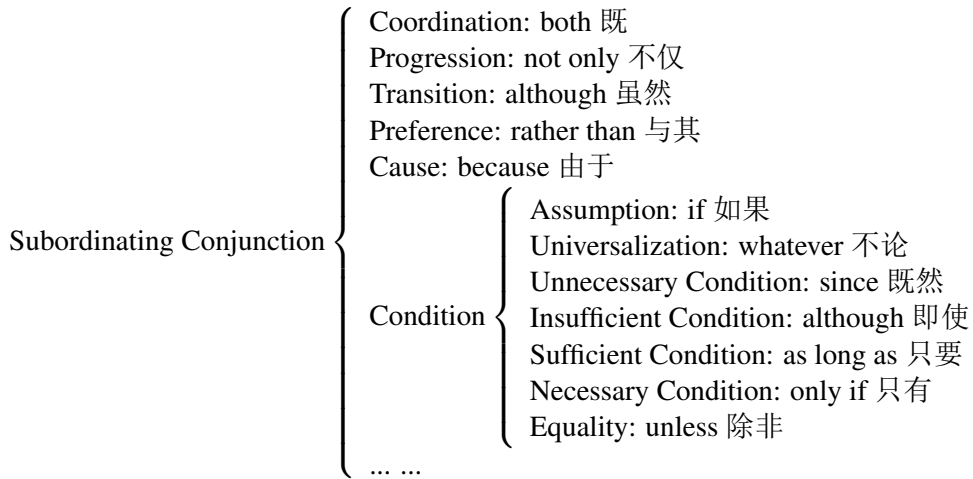


Figure 1: Abbreviated Hierarchical subcategories of subordinating conjunctions with examples.

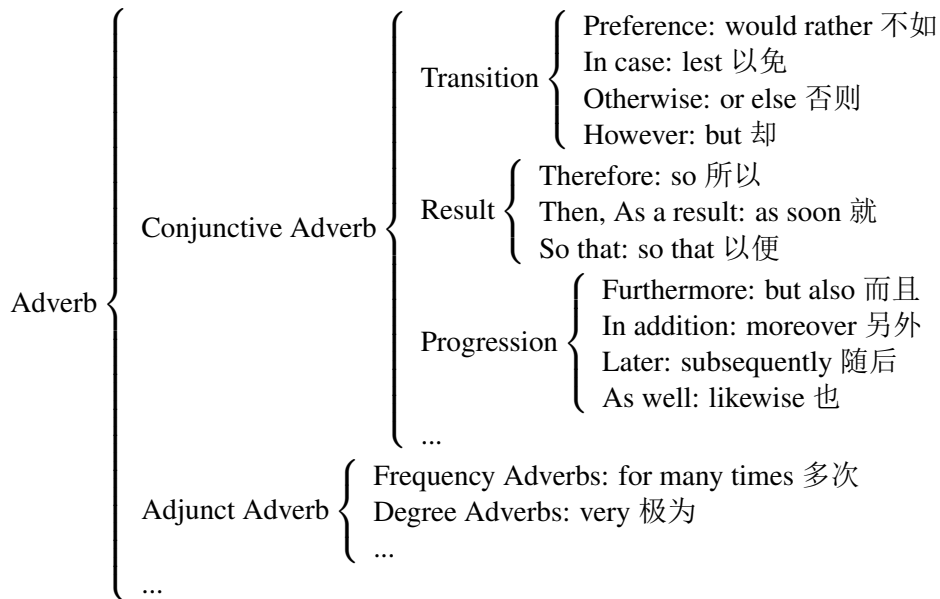


Figure 2: Abbreviated Hierarchical subcategories of adverbs with examples.

the types of their associated arguments: “benefactive”, such as “为(for)” and “给(to)”, marks the beneficiary of an action; “locative”, such as “在(in)”, marks adverbials that indicate the place of the event; “direction”, such as “向(towards)” and “由(from)”, marks adverbials that answer the questions “from where?” and “to where?”; “temporal”, such as “在(on)”, marks temporal or aspectual adverbials that answer the question “when?”, and so on.

#### 4 Refining Grammar with Hierarchical Category Refinement

In this section, we choose the appropriate granularity in a data-driven manner based on the split-merge learning method in Section 2.1. Our approach first initializes the categories with the most general subcategories in the taxonomy and then splits the categories through the hypernym-hyponym relation in the

taxonomy. Data-driven method is used to merge the overly refined subcategories.

The top category in the taxonomy is used as the starting annotations of POS tags. As we cannot predict which layer should be the most adequate one, we try to avoid applying any priori restriction on the refinement granularity, and start with the most general tags.

With the hierarchical knowledge, it turns out to be a critical issue that which granularity should be used to refine the tags for parsing. We intend to take neither too coarse subcategories nor too fine ones in the hierarchical knowledge for parsing. Instead, it would be our advantage to split the tags with the very granularity where needed, rather than splitting them all to one specific granularity in the taxonomy.

For example, “Conjunctive Adverbs” are divided into three subcategories in our taxonomy as shown in Figure 2. The evidence for the refinement may occur in very rare case, and certainly some of the context of the different subcategories are quite the same. Splitting symbols with the same context is not only unnecessary, but potentially harmful, since it unreasonably fragments observations of other symbols’ behavior.

In this paper, the hierarchical subcategory knowledge is used to refine grammars by supervising the automatic hierarchical state-split approach. In the split stage in each cycle, the function word subcategory is split along the hierarchy of the knowledge, instead of being randomly split and classified automatically. In this way, we try to alleviate the over-fitting of the greedy data-driven approach, and a new set of knowledge-related tags are generated. In the following step, we retreat some of the split subcategories to their more general layer according to its likelihood loss of merging them. In this way, we try to avoid the excessive refinement in our hierarchical knowledge without sufficient data support.

There are two issues that we have to consider in this process: a) how to deal with the polysemous words, and b) how to deal with the multi-branch situation other than binary branch in the taxonomy. Regarding to the polysemous words, they occur mostly in two situation for function words. Some are the polysemous words which can be taken as conjunctions or auxiliary words, while the others can be taken as preposition or adverbs. Fortunately there is no ambiguity for a word given its POS tag, so we could neglect this situation in the split process when training. We demonstrated the solution for the multiple branches in the Section 5.

## 5 Learning the Taxonomy of Function Words

There are multiple subcategorization criteria for building function word taxonomy, and it is difficulty for human to rank the ordering in the classification process. This section represents the method of learning the taxonomy of the function words in data-driven manner. Based on the manual tabular classification, the similar word classes are conflated to express the data distribution.

The multiple branches in the taxonomy are intractable for the original split-merge method, because it splits every category into two and merges half of them for efficiency. If we follow this scheme in our training process, it would be difficult to deal with the multi-branch situation in the taxonomy, because how to choose the first two to split among the multiple branches is another challenge. It is an equally difficult problem for us to binarize the taxonomy by hand comparing to directly choosing the granularity.

It would be our advantage to binarize the taxonomy by a data-driven method. For automatic binarization, a straightforward approach is to measure the utility of traversing all the plausible ways of cutting all the branches into two sets individually and use the best one. Then we can deal with the divided two sets in the same manner recursively. However, not only is this impractical, requiring an entire training phase for each possible binarization scheme which is exponentially expensive, but it assumes the contributions of multiple binarizations in different branches are independent. In fact, extra sub-symbols may need to be added to several nonterminals before they can cooperate to pass information along the parse tree.

Therefore, we go in the opposite direction, and propose an extended version of split-merge learning to handle the multiple branches in the taxonomy. That is, we split each state into all the subcategories in the lower layer in the taxonomy even if it has multiple branches, train, and then measure for every two sibling subcategories in the same layer the loss in likelihood incurred when merging them. If this loss is small, the new division of these two subcategories does not carry enough useful information and can be merged back. Contrary to the gain in likelihood for splitting, the loss in likelihood for merging can be

efficiently approximated (Petrov et al., 2006).

More specifically, we assume transitivity in merging multiple subcategories in one layer. Figure 3 gives an illustration. After the split stage, the category A has been split into subcategories A-1, A-2, ... to A-7. Then we compute the loss in likelihood of the training data by merging back each pair of two subcategories through A-1 to A-7. If the loss is lower than a certain threshold<sup>1</sup> set for each round of merge, this pair of newly split subcategories will be merged. We only show the sibling ones for brevity in this example. Assume the losses of merging these pairs (A-1, A-2), (A-2, A-3), (A-3, A-4) and (A-4, A-5) are below the threshold  $\varepsilon$ . Thus, A-1, A-2, A-3, A-4 and A-5 are merged to X-1 due to the transitivity of the connected points, where X-1 is the automatically generated subcategory which contains the five conflated subcategories as its descendants. At the meantime, A-6 and A-7 still remain. This scheme is an approximation because it merges subcategories that should be merged with the same subcategory. But it will leave the split of this instances to the next round when more evidence on interaction with other more refined subcategories is given.

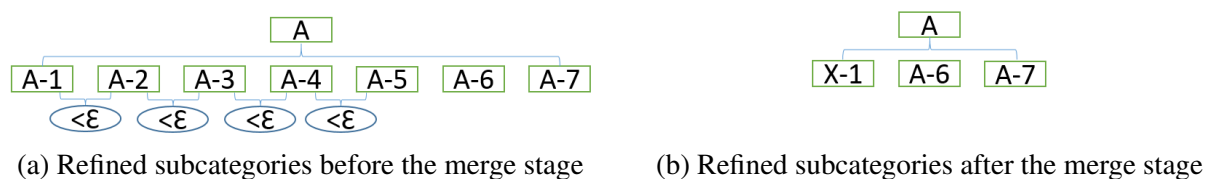


Figure 3: Illustration of merging the subcategories for multiple branches in the taxonomy. Where  $\varepsilon$  is a certain threshold below which this pair of subcategories will be merged, and X is the automatically generated subcategory which contains the conflated subcategories as its descendants.

After merging in each round, the hierarchical knowledge is reshaped to fit the practical usage in the Treebank. The split-merge cycles allow us to progressively increase the complexity of the hierarchical knowledge, and the more useful distinctions are represented as the higher level in the taxonomy, which gives priority to the most useful distinctions in return by supervising the grammar induction. Figure 4 demonstrates the transformation of the hierarchical structure from the tabular classification. Along this road, the training scheme is not a unilateral training, but an interactive process between the knowledge representation learning and the grammar training. Our learning process exerts a mutual effect to both the induced grammar and the optimized structure of the hierarchical knowledge. In this way, the set of dividing standards are chosen iteratively according to their syntactic features. The more effective divisions are conducted in the early stages. In the following stages, the divisions which interact with previous divisions to give the most effective disambiguating information are adopted. The final taxonomy are built based on manual classification in data-driven approach, and the hierarchical structure are optimized and rational in the perspective of actual data distribution. Figure 4 illustrates a concrete instance of the procedure of learning the taxonomy. On one hand, this procedure provides a more rational hierarchical subcategorization structure according to data distribution. On the other hand, the order of the division criteria represents the priorities the grammar induction takes for each criterion. The structure in the higher levels of the taxonomy are determined by the dominant syntactic characteristics. And the division in the later iterations are on the basis of minor distinctive characteristics.

## 6 Experiments and Results

### 6.1 Data Set

We present experimental results on both CTB5.0 (All traces and functional tags were stripped.) and TCT.

We ran experiments on CTB5.0 using the standard data allocation: files from CHTB\_001.fid to CHTB\_270.fid, and files from CHTB\_400.fid to CHTB\_1151.fid were used as training set. The development set includes files from CHTB\_301.fid to CHTB\_325.fid, and the test set includes files CHTB\_271.fid

<sup>1</sup>In practice, instead of setting a predefined threshold for merging, we merge a specific number of the newly split subcategories.

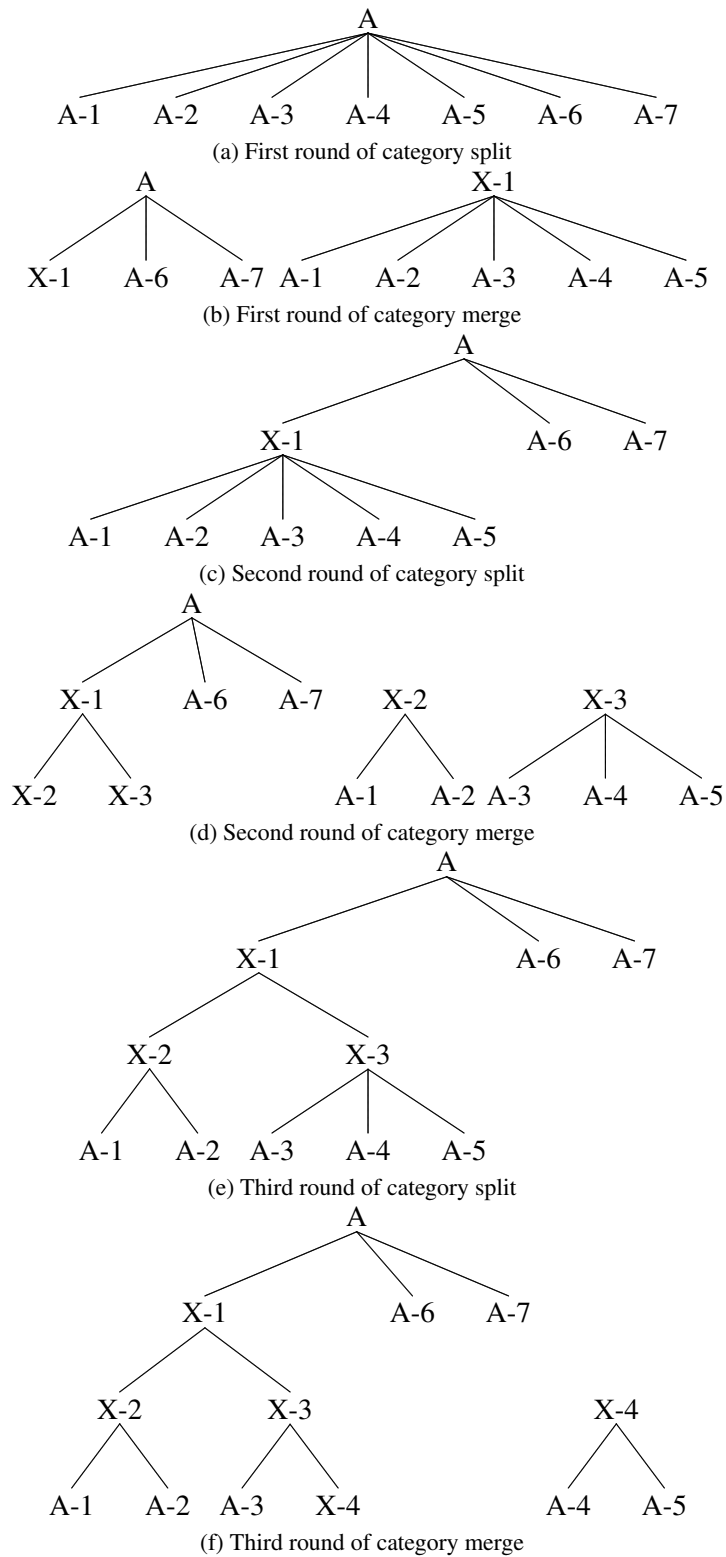


Figure 4: Iteration of grammar induction and taxonomy structure learning

to CHTB\_300.fid. Experiments on TCT use the data set as in CIPS-SIGHAN-ParsEval-2012 (Zhou, 2012). We have parsed on the segmented text in the Treebank, namely, no use of gold POS-tags, use of gold segmentations, and full-length sentences. This is the same as for other 5 parsers in Table 1 for comparison. All the experiments were carried out after six cycles of split-merge.

## 6.2 Final Results

The final results are shown in Table 1. Our final parsing performance is higher than both the manual annotation method (Levy and Manning, 2003) and the data-driven method (Petrov, 2009).

Parser	Precision	Recall	F <sub>1</sub>
Levy(2003)	78.40	79.20	78.80
Petrov(2009)	84.82	81.93	83.33
Lin(2009)	86.00	83.10	84.50
Qian(2012)	84.57	<b>83.68</b>	84.13
Zhang(2013)	84.42	84.43	84.43
This paper	<b>86.55</b>	83.41	<b>84.95</b>

Table 1: Our final parsing performance compared with the best previous work on CTB5.0.

On test set TCT, the method achieves the best precision, recall and F-measure in the CIPS-SIGHAN-ParsEval-2012 competition, and table 2 compares our results with the system of Beijing Information Science and Technology University (BISTU) which got the second place in the competition.

Parser	Precision	Recall	F <sub>1</sub>
BISTU	70.10	68.08	69.08
This paper	<b>76.81</b>	<b>76.66</b>	<b>76.74</b>

Table 2: Our final parsing performance compared with the best previous works on TCT.

Given the manual labor required for generating the taxonomy (and in languages where there is a taxonomy, determining whether it is suitable), this first study focuses on a language where there is quite a bit of under- and over-specification in the Treebanks' tag sets. So this work is only implemented on Chinese. We regard it as future work to transfer this approach to other languages.

## 6.3 Analysis

The outline of constructing the taxonomy of function words are as follows. Firstly, the function words are manually subcategorized in a rough and cursory way. When dealing with subcategories hard to resolve their relation of subordination, we simply treat them as siblings in the tree in a rather flat structure, and leave the elaboration of exquisite clustering to the algorithms. The data-driven approach in Section 4 automatically choose the appropriate granularity of refinement for our grammar. Moreover, the split-merge learning for multiple branches in the hierarchical subcategories in Section 5 exploits the relationship between the sibling nodes in the same layer, making use of the Treebank data to adjust and optimize the hierarchy.

During the split-merge process, the hierarchical subcategories are learned to fit the data, which is a transformation of our manually defined hierarchy. The transformed hierarchy is just the route map of subcategories employed in our model. As abbreviated in Figure 5 and Figure 6, many distinctions between word sets of the subcategories have been exploited by our approach, and the learned taxonomy is interpretable. For instance, It shows that the learned structure of the taxonomy is reasonable.

## 6.4 Comparison with Previous Work

Although the taxonomy of function words are learned in the grammar training process, the grammar is trained on the Treebank in supervised manner. Thus, this work is not directly relevant with unsupervised grammar induction literature (Headden III et al., 2009; Berant et al., 2007; Mareček and Žabokrtský, 2014).



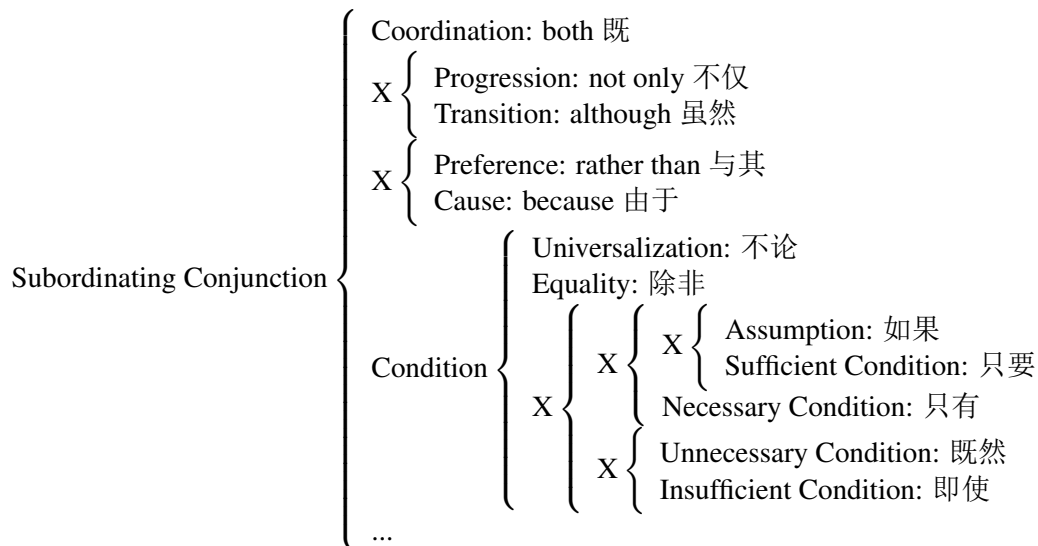


Figure 5: Abbreviated automatically learned hierarchical subcategories of subordinating conjunctions with examples. Where “X” represents the automatically generated subcategory.

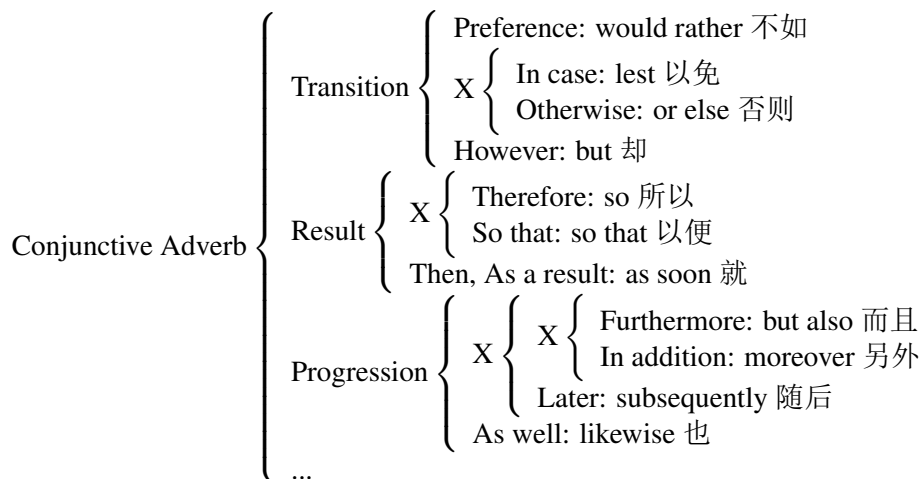


Figure 6: Abbreviated automatically learned hierarchical subcategories of adverbs with examples.

Lin et al. (2009) and Li et al. (2014b) presented ideas of using either hierarchical semantic knowledge from HowNet for content words or grammar knowledge for subordinating conjunctions. They introduced hierarchical subcategory knowledge in a different stage. They split the original Treebank categories in split-merge process according to the data, and then find a method to map the subcategories to the node in the taxonomy, and constrain their further splitting. Comparing to their work, our approach is more delicate, which is splitting the categories according to the knowledge, and learning the knowledge structure according to data during the training course. Lin et al. (2009) incorporated semantic knowledge of content words into the data-driven method. It would be promising if this work stacks with the content word knowledge. However, the work with content word knowledge have to handle the polysemous words in the semantic taxonomy, so they split the categories according to the data, and then find a way to map the subcategories to the node in the taxonomy, and constrain their further splitting. It is our goal to make these two methods compatible with each other.

Incorporating word formation knowledge achieved higher parsing accuracy according to Zhang and

Clark (2011). However, they ran their experiment on gold POS-tags and a different data set split, which is different from the setup of work in Table 1 including this work. They also presented their result on automatically assigned POS-tags and the same data set split as in the work in Table 1 to facilitate the performance comparison. It gave  $F_1$  score of 81.45% for sentences with less than 40 words and 78.3% for all sentences, significantly lower than Petrov and Klein (2007).

Zhang et al. (2013) exhaustively exploited character-level syntactic structures for words, and achieved 84.43% on  $F_1$  measure. They placed more emphasis on the word-formation of content words, which our model highlights the value of the function words. The complementary intuitions make it possible to integrate these approaches together in the future work.

## 7 Conclusion

This paper presents an approach for inducing finer syntactic categories while learning the taxonomy for function words. It used linguistic insight to guide the state-split process, and the hierarchical structure representing syntactic features of function word usages was established during the grammar training process. Empirical evidence has been provided that automatically subcategorizing function words contributes to high parsing performance. The induced grammar supervised by the taxonomy outperformed previous approaches, which benefited from both the knowledge and the data-driven method. The proposed approach for learning the structure of the taxonomy could be generalized to construct semantic knowledge base.

## Acknowledgments

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.
- Jonathan Berant, Yaron Gross, Matan Mussel, Ben Sandbank, Eytan Ruppín, and Shimon Edelman. 2007. Boosting unsupervised grammar induction by splitting complex sentences on function words. In *Proceedings of the Boston University Conference on Language Development*.
- 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.
- 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.
- Zhengdong Dong and Qiang Dong. 2006. *HowNet and the computation of meaning*. World Scientific Publishing Co. Pte. Ltd.
- Christiane Fellbaum. 1999. *WordNet*. Wiley Online Library.
- 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.
- William P Headden III, Mark Johnson, and David McClosky. 2009. Improving unsupervised dependency parsing with richer contexts and smoothing. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 101–109. Association for Computational Linguistics.
- 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.

- 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. 2014a. Improved parsing with taxonomy of conjunctions. In *2014 IEEE China Summit & International Conference on Signal and Information Processing*. IEEE.
- Dongchen Li, Xiantao Zhang, and Xihong Wu. 2014b. Learning grammar with explicit annotations for subordinating conjunctions in chinese. In *Proceedings of the 52th annual meeting of the Association for Computational Linguistics Student Research Workshop*. Association for Computational Linguistics.
- 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.
- David Mareček and Zdeněk Žabokrtský. 2014. Dealing with function words in unsupervised dependency parsing. In *Computational Linguistics and Intelligent Text Processing*, pages 250–261. Springer.
- 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.
- George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.
- 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, 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.
- Detlef Prescher. 2005. Inducing head-driven pcfgs with latent heads: Refining a tree-bank grammar for parsing. In *Machine Learning: ECML 2005*, pages 292–304. Springer.
- Hui Wang and Shiwen Yu. 2003. The semantic knowledge-base of contemporary chinese and its applications in wsd. In *Proceedings of the second SIGHAN workshop on Chinese language processing-Volume 17*, pages 112–118. Association for Computational Linguistics.
- Fei Xia. 2000. The part-of-speech tagging guidelines for the penn chinese treebank (3.0). *Technical report*.
- Deyi Xiong, Shuanglong Li, Qun Liu, Shouxun Lin, and Yueliang Qian. 2005. Parsing the penn chinese treebank with semantic knowledge. In *Natural language processing-IJCNLP 2005*, pages 70–81. Springer.
- Nianwen Xue, Fei Xia, Shizhe Huang, and Anthony Kroch. 2000. The bracketing guidelines for the penn chinese treebank (3.0). *Technical report*.
- 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. 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.
- Qiang Zhou. 2012. Evaluation report of the third chinese parsing evaluation: Cips-sighan-parseval-2012. In *Proceedings of the second CIPS-SIGHAN joint conference on Chinese language processing*, pages 159–167.
- Xuefeng Zhu, Shiwen Yu, and Hui Wang. 1995. The development of contemporary chinese grammatical knowledge base and its applications. *International journal of asian language processing*, 5(1,2):39–41.