Refining Grammars for Parsing with Hierarchical Semantic Knowledge

Xiaojun Lin, Yang Fan, Meng Zhang, Xihong Wu*, Huisheng Chi
Speech and Hearing Research Center
Key Laboratory of Machine Perception (Ministry of Education)
School of Electronics Engineering and Computer Science
Peking University, Beijing, 100871, China
{linxj, fanyang, zhangm, wxh}@cis.pku.edu.cn, chi@pku.edu.cn

Abstract

This paper proposes a novel method to refine the grammars in parsing by utilizing semantic knowledge from HowNet. Based on the hierarchical state-split approach, which can refine grammars automatically in a data-driven manner, this study introduces semantic knowledge into the splitting process at two steps. Firstly, each part-of-speech node will be annotated with a semantic tag of its terminal word. These new tags generated in this step are semantic-related, which can provide a good start for splitting. Secondly, a knowledge-based criterion is used to supervise the hierarchical splitting of these semantic-related tags, which can alleviate overfitting. The experiments are carried out on both Chinese and English Penn Treebank show that the refined grammars with semantic knowledge can improve parsing performance significantly. Especially with respect to Chinese, our parser achieves an F_1 score of 87.5%, which is the best published result we are aware of.

1 Introduction

At present, most high-performance parsers are based on probabilistic context-free grammars (PCFGs) in one way or another (Collins, 1999; Charniak and Johnson, 2005; Petrov and Klein, 2007). However, restricted by the strong context-free assumptions, the original PCFG model which simply takes the grammars and probabilities off a treebank, does not perform well. Therefore, a variety of techniques have been developed to enrich and generalize the original grammar, ranging from lexicalization to symbol annotation.

Lexicalized PCFGs use the structural features on the lexical head of phrasal node in a tree, and get significant improvements for parsing (Collins, 1997; Charniak, 1997; Collins, 1999; Charniak, 2000). However, they suffer from the problem of fundamental sparseness of the lexical dependency information. (Klein and Manning, 2003).

In order to deal with this limitation, a variety of unlexicalized parsing techniques have been proposed. Johnson (1998) annotates each node by its parent category in a tree, and gets significant improvements compared with the original PCFGs on the Penn Treebank. Then, some manual and automatic symbol splitting methods are presented, which get comparable performance with lexicalized parsers (Klein and Manning, 2003; Matsuzaki et al., 2005). Recently, Petrov et al. (2006) introduces an automatic hierarchical state-split approach to refine the grammars, which can alternately split and merge the basic nonterminals by the Expectation-Maximization (EM) algorithm. In this method, the nonterminals are split to different degrees, as appropriate to the actual complexity in the data. The grammars refined in this way are proved to be much more accurate and compact than previous work on automatic annotation. This data-driven method still suffers from the overfitting problem, which may be improved by integrating other external information.

In this paper, we propose a novel method that combines the strengths of both data-driven and knowledge-driven strategies to refine grammars. Based on the work proposed by Petrov et al. (2006), we use the semantic knowledge from HowNet (Dong and Dong, 2000) to supervise the hierarchical state-split process at the part-of-speech(POS) level. At first, we define the most general hypernym in HowNet as the semantic class of a word, and then use this semantic class to initialize the tag of each POS node. In this way, a new set of semantic-related tags is generated, and

*Corresponding author: Xihong Wu.

Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, pages 1298–1307,
Singapore, 6-7 August 2009. ©2009 ACL and AFNLP

a good starting annotation is provided to reduce the search space for the EM algorithm in the splitting process. Then, in order to mitigate the overfitting risk, the hierarchical hypernym-hyponym relation between hypernyms in HowNet is utilized to supervise the splitting of these new semantic-related tags. By introducing a knowledge-based criterion, these new tags are decided whether or not to split into subcategories from a semantic perspective. To investigate the effectiveness of the presented approach, several experiments are conduced on both Chinese and English. They reveal that the semantic knowledge is potentially useful to parsing.

The remainder of this paper is organized as follows. Section 2 reviews some closely related works, including the lexical semantic related parsing and the hierarchical state-split unlexicalized parsing. In section 3, the presented method for grammar refining is described in detail, and several experiments are carried out for evaluation in Section 4. Conclusions are drawn in Section 5.

2 Background

This paper tries to refine the grammars through an improved hierarchical state-split process integrated with semantic knowledge. The related works are reviewed as follows.

2.1 Lexical Semantic Related Parsing

Semantic knowledge is useful to resolving syntactic ambiguities, and a variety of researches focus on how to utilize it. Especially in recent years, a conviction arose that semantic knowledge could be incorporated into the lexicalized parsing.

Based on the lexicalized grammars, Bikel (2000) attempts at combining parsing and word sense disambiguation in a unified model, using a subset of SemCor (Miller et al., 1994). Bikel (2000) evaluates this model in a parsing context with sense information from WordNet, but does not get improvements on parsing performance.

Xiong et al. (2005) combines word sense from CiLin and HowNet (two Chinese semantic resources) in a generative parsing model, which generalizes standard bilexical dependencies to word-class dependencies, and indeed help to tackle the sparseness problem in lexicalized parsing. The experiments show that the parse model combined with word sense and the most special hypernyms achieves a significant improvement on Penn Chinese Treebank. This work only considers the most special hypernym of a word, rather than other hypernyms at different levels of the hypernym-hyponym hierarchy.

Then, Fujita et al. (2007) uses the Hinoki treebank as training data to train a discriminative parse selection model combining syntactic features and word sense information. Instead of utilizing the most special hypernym, the word sense information in this model is embodied with more general concepts. Based on the hand-craft sense information, this model is proved to be effective for parse selection.

Recently, Agirre et al. (2008) train two lexicalized models (Charniak, 2000; Bikel, 2004) on preprocessed inputs, where content words are substituted with semantic classes from WordNet. By integrating the word semantic classes into the process of parser training directly, these two models obtain significant improvements in both parsing and prepositional phrase attachment tasks. Zhang (2008) does preliminary work on integrating POS with semantic class of words directly, which can not only alleviate the confusion in parsing, but also infer syntax and semantic information at the same time.

2.2 The Hierarchical State-split Parsing

In order to alleviate the context-free assumptions, Petrov et al. (2006) proposes a hierarchical state-split approach to refine and generalize the original grammars, and achieves state-of-the-art performance. Starting with the basic nonterminals, this method repeats the split-merge (SM) cycle to increase the complexity of grammars. That is, it splits every symbol into two, and then re-merges some new subcategories based on the likelihood computation.

Splitting

In each splitting stage, the previous syntactic symbol is split into two subcategories, and the EM algorithm is adopted to learn probability of the rules for these latent annotations to maximize the likelihood of trees in the training data. Finally, each symbol generates a series of new subcategories in a hierarchical fashion. With this method, the splitting strategy introduces more context information, and the refined grammars cover more linguistic information which helps resolve the syntactic ambiguities.

However, it is worth noting that the EM algorithm does not guarantee a global optimal solution, and often gets stuck in a suboptimal configuration. Therefore, a good starting annotation is expected to help alleviate this problem, as well as reduce the search space for EM.

Merging

It is obvious that using more derived subcategories can increase accuracy, but the refined grammars fit tighter to the training data, and may lead to overfitting to some extent. In addition, different symbols should have their specific numbers of subcategories. For example, the comma POS tag should have only one subcategory, as it always produces the terminal comma. On the contrary, the noun POS tag and the verb POS tag are expected to have much more subcategories to express their context dependencies. Therefore, it is not reasonable to split them in the same way.

The symbol merging stage is introduced to alleviate this defect. This approach splits symbols only where needed, and it is implemented by splitting each symbol first and then measure the loss in likelihood incurred when removing this subcategory. If the loss is small, it means that this subcategory does not take enough information and should be removed. In general it is hard to decide the threshold of the likelihood loss, and this merging stage is often executed by removing a certain proportion of subcategories, as well as giving priority to the most informative subcategories.

By splitting and merging alternately, this method can refine the grammars step by step to mitigate the overfitting risk to some extent. However, this data-driven method can not solve this problem completely, and we need to find other external information to improve it.

Analysis

The hierarchical state-split approach is used to split all the symbols in the same way. Table 1 cites the subcategories for several POS tags, along with their two most frequent words. Results show that the words in the same subcategory of POS tags are semantic consistent in some cases. Therefore, it is expected to optimize the splitting and merging process at the POS level with semantic knowledge.

NR

NR-0	大甲溪(Daja river)	尼泊尔(Nepal)
NR-1	新力(Sony)	伯乐网(Bole Co.)
NR-2	华诚(C. Hua)	文天祥(T. Wen)
NR-3	乐绍延(S. Yue)	商(Shang)

LC

LC-0	当中(middle)	右侧(right)
LC-1	以前(before)	以来(since)
LC-2	开始(start)	止(end)
LC-3	为止(till)	末(end)

P

P-0	每当(whenever)	至于(as for)
P-1	犹如(like)	仿佛(as)
P-2	朝着(look to)	照着(according to)
P-3	傍(be close to)	比照(contrast)

Table 1: The two most frequent words in the subcategories of several POS tag.

3 Integration with Semantic Knowledge

In this paper, the semantic knowledge is used to refine grammars by improving the automatic hierarchical state-split approach. At first, in order to provide good starting annotations to reduce the search space for the EM algorithm, we try to annotate the tag of each POS node with the most general hypernym of its terminal word. In this way, we generate a new set of semantic-related tags. And then, instead of splitting and merging all symbols together automatically, we propose a knowledge-based criterion with hierarchical semantic knowledge to supervise the splitting of these new semantic-related tags.

3.1 HowNet

The semantic knowledge resource we use is HowNet, which is a common sense knowledge base unveiling concepts and inter-conceptual relations in Chinese and English.

As a knowledge base of graph structure, HowNet is devoted to demonstrating the properties of concepts through sememes and relations between sememes. Broadly speaking, a sememe refers to the smallest basic semantic unit that cannot be reduced further, which can be represented in English and their Chinese equivalents, such as the sememe *institution|机构*. The relations explicated in HowNet include hypernym-hyponym relations, location-event relations, time-event relations and so on. In this work, we mainly focus on

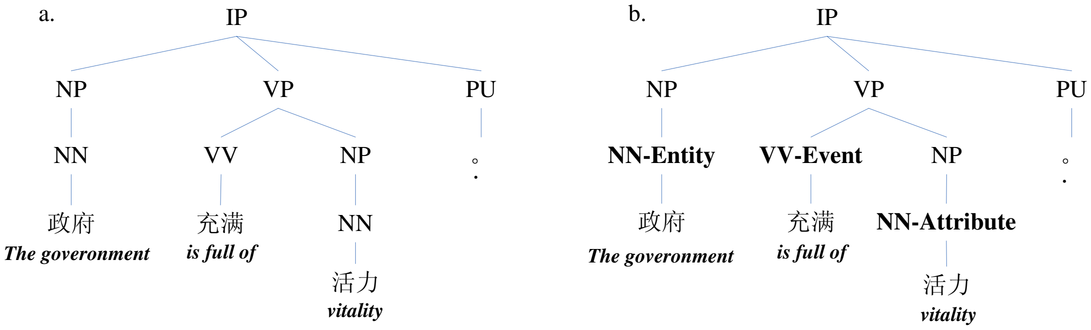

Figure 1: The two syntax trees of the sentence "*The government is full of vitality*". a. is the original syntax tree, b. is the syntax tree in which each tag of the POS node is annotated with the most general hypernym of its terminal word.

the hypernym-hyponym relations. Take the word 政府(government) as an example, its hypernyms with the hierarchical hypernym-hyponym relations are listed below from speciality to generality, which we call hierarchical semantic information in this paper.

institution|机构→group|群体→thing|万物→entity|实体

It is clear that this word 政府(government) has hypernyms from the most special hypernym *institution|机构* to the most general hypernym *entity|实体* in a hierarchical way.

In HowNet(Update 2008), there are 173535 concepts, with 2085 sememes. The sememes are categorized into entity, event, attribute, attribute value, etc., each corresponding to a sememe hierarchy tree.

3.2 Annotating the Training Data

One of the original motivations for the grammar refinement is that the original symbols, especially the POS tags, are usually too general to distinguish the context dependencies. Take the sentence in Figure 1 for example, the word 政府(government) should have different context dependencies compared with the word 活力(vitality), although both of them have the same POS tag "NN". In fact, the two words are defined in HowNet with different hypernyms. The word 政府(government) is defined as a kind of objective things, while the word 活力(vitality) is defined as a property that is often used to describe things. It is obvious that the different senses can represent their different syntax structures, and we expect to refine the POS tags with semantic knowledge.

In the automatic hierarchical state-split approach introduced above, the EM algorithm is used to search for the maximum of the likelihood during the splitting process, which can generate subcategories for POS tags to express the context dependencies. However, this method often gets stuck in a suboptimal configuration, which varies depending on the start point. Therefore, a good start of the annotations is very important. As it is displayed in Figure 1, we annotate the tag of each POS node with the hypernym of its terminal word as the starting annotation. There are two problems that we have to consider in this process: a) how to choose the appropriate semantic granularity, and b) how to deal with the polysemous words.

As mentioned above, the semantic information of each word can be represented as hierarchical hypernym-hyponym relations among its hypernyms. In general, it is hard to decide the appropriate level of granularity to represent the word. The semantic 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 structural information. If more special kinds of semantic classes are chosen, it will make the structural information weaker. As annotations with the special hypernym always defeat some of the advantage of automatically latent annotations learning, we annotate the training data with the most general hypernym. For example, as shown in Figure 1, the POS tag "NN" of 政府(government) is annotated as "NN-Entity", and "NN" of 活力(energy) is annotated as "NN-Attribute".

Another problem is how to deal with the polysemous words in HowNet. In fact, when we choose the most general hypernym as the word's semantic

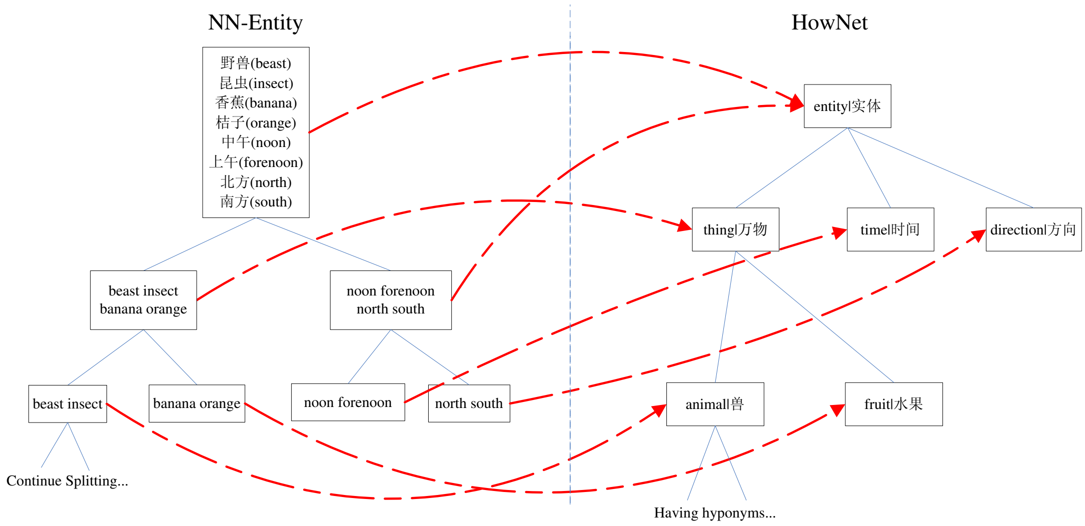

Figure 2: A schematic figure for the hierarchical state-split process of the semantic-related tag "NN-Entity". Each subcategory of this tag has its own word set, and corresponds to one hypernym at the appropriate level in HowNet.

representation, this problem has been alleviated to a large extent. In this paper we adopt the first sense option as our word sense disambiguation (WSD) strategy to determine the sense of each token instance of a target word. That is to say, all token instances of a given word are tagged with the sense that occurs most frequently in HowNet. In addition, we keep the tag of the POS node whose terminal word is not defined in HowNet unchanged.

3.3 Supervising the Hierarchical State-split Process

With the method proposed above, we can produce a good starting annotation with semantic knowledge, which is of great use to constraining the automatic splitting process. Our parser is trained on the good starting annotations with the automatic hierarchical state-split process, and gets improvements compared with the original training data. However, during this process, only the most general hypernyms are used as the semantic representation of words, and the hierarchical semantic knowledge is not explored. In addition, the automatic process tries to refine all symbols together through a data-driven manner, which suffers the overfitting risk.

After annotating the training data with hypernyms, a new set of semantic-related tags such as "NN-Entity" is produced. We treat the refining process of these semantic-related tags as the specializing process of hypernym with hierarchical semantic knowledge. Each subcategory of these tags corresponds to a appropriate special level of hypernym in the HowNet. For example, every subcategory of "NN-Entity" could corresponds to a appropriate hyponym of *entity*|实体.

We integrate the hierarchical semantic knowledge into the original hierarchical state-split process to refine these semantic-related tags. First of all, it is necessary to establish the mapping from each subcategory of these semantic-related tags to the hypernym at the appropriate level in HowNet. Then, instead of likelihood judgment, a knowledge-based criterion is proposed, to decide whether or not to remove the new subcategories of these tags. That is to say, once the parent tag of this new subcategory is mapped onto the most special hypernym without any hyponym, it should be removed immediately.

The schematic Figure 2 demonstrates this semantically supervised splitting process. The left part of this figure is the subcategories of the semantic-related tag "NN-Entity", which is split hierarchically. As expressed by the dashed line, each subcategory corresponds to one hypernym in the right part of this figure. If the hypernym node has no hyponym, the corresponding subcategory will stop splitting.

The mapping from each subcategory of these semantic-related tags to the hypernym at the appropriate level is implemented with the word set related to this subcategory. As it is shown in Fig-

DataSet	Chinese	English
	Xue et al. (2002)	Marcus et al. (1993)
TrainSet	Art. 1-270,400-1151	Sections 2-21
DevSet	Articles 301-325	Section 22
TestSet	Articles 271-300	Section 23

Table 2: Experimental setup.

ure 2, the original tag "NN-Entity" treats all the words it products as its word set. Once the original category is split into two subcategories, its word set is also split, through forcedly dividing each word in the word set into one subcategory which is most frequent with this word. And then, each subcategory is mapped onto the most specific hypernym that contains its related word set entirely in HowNet. On this basis, a new knowledge-based criterion is introduced to enrich and generalize these semantic-related tags, with purpose of fitting to the hierarchical semantic structure rather than the training data.

4 Experiments

In this section, we designed several experiments to investigate the validity of refining grammars with semantic knowledge.

4.1 Experimental Setup

We did experiments on Chinese and English. In order to make a fair comparison with previous works, we split the standard corpora as shown in Table 2. Our parsers were evaluated by the EVALB parseval reference implementation[1]. The Berkeley parser[2] was used to train the models with the original automatic hierarchical state-split process. The semantic resource we used to improve parsing was HowNet, which has been introduced in Subsection 3.1. Statistical significance was checked using Dan Bikel' s randomized parsing evaluation comparator with the default setting of 10,000 iterations[3].

4.2 Semantic Representation Experiments

First of all, we ran experiments with different semantic representation methods on Chinese. The polysemous words in the training set were annotated with the WSD strategy of first sense option, which was proved to be useful in Agirre et al. (2008).

As mentioned in Subsection 3.1, the semantic information of each word can be represented as a hierarchical relation among its hypernyms from specialty to generalization in HowNet. In order to choose the appropriate level of granularity to represent words, we annotated the training set with different levels of granularity as semantic representation. In our experiments, the automatic hierarchical state-split process is used to train models on these training sets with different level of semantic representation.

We tried two kinds of semantic representations, one is using the most general hypernym, and the other is using the most special hypernym. Results in Figure 3 proved the effectiveness of our method in Subsection 3.2. When we annotated the tag of each POS node with the most general hypernym of its terminal word, the parser performs much better than both the baseline and the one annotated with the most special hypernym. Moreover, the F_1 score starts dropping after 3 training iterations on the training set annotated with the most special hypernyms, while it is still improving with the most general one, indicating overfitting.

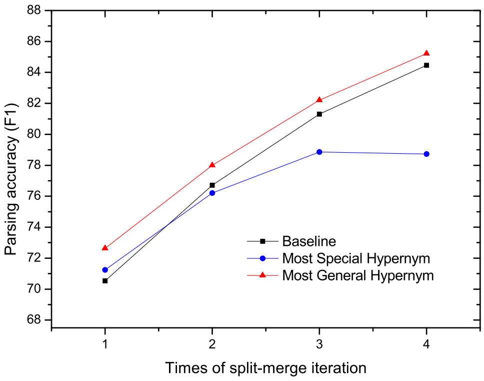

Figure 3: Performances on Chinese with different semantic representations: the training set without semantic representation, the training set annotated with the most special hypernyms, and the training set annotated with the most general hypernyms.

When the training set was annotated with the most general hypernyms, there were only 57 new semantic-related tags such as "NN-Entity", "NN-Attribute" and so on. However, when the training set was annotated with the most special hypernyms, 4313 new tags would be introduced. Ob-

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

[2] http://code.google.com/p/berkeleyparser/.

[3] http://www.cis.upenn.edu/ dbikel/software.html.

viously, it introduces too many tags at once and is difficult to refine appropriate grammars in the subsequent step starting with this over-splitting training set.

4.3 Grammar Refinement Experiments

Several experiments were carried out on Chinese and English to verify the effectiveness of refining grammars with semantic knowledge. We took the most general hypernym as the semantic representation, and the polysemous words in the training set were annotated with the WSD strategy of first sense option.

In our experiments, three kinds of method were compared. The baseline was trained on the raw training set with the automatic hierarchical state-split approach. Then, we improved it with the semantic annotation, which annotated the raw training set with the most general hypernyms as semantic representations, while keeping the training approach used in the baseline unchanged. Further, our knowledge-based criterion was introduced to supervise the automatic hierarchical state-split process with semantic knowledge.

In this section, since most of the parsers (including the baseline parser and our advanced parsers) had the same behavior on development set that the accuracy continued increasing in the five beginning iterations and then dropped at the sixth iteration, we chose the results at the fifth iteration as our final test set parsing performance.

Performances on Chinese

Figure 4 shows that refining grammars with semantic knowledge can help improve parsing performance significantly on Chinese (sentences of length 40 or less). Benefitting from the good starting annotations, our parser achieved significant improvements compared with the baseline (86.8% vs. 86.1%, p<.08). It proved that the good starting annotations with semantic knowledge were effective in the splitting process. Further, we supervised the splitting of the new semantic-related tags from the semantic annotations, and achieved the best results at the fifth iteration. The best F_1 score reached 87.5%, with an error rate reduction of 10.1%, relative to the baseline (p<.004).

Table 3 compared our methods with the best previous works on Chinese. The result showed that refining grammars integrated with semantic knowledge could resolve syntactic ambiguities remarkably and achieved the state-of-the-art performance on Chinese.

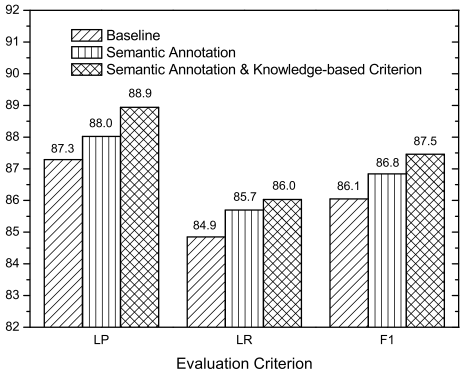

Figure 4: Performances at the fifth iteration on Chinese (sentences of length 40 or less) with three methods: the baseline, the parser trained on the semantic annotations with automatic method, and the parser trained on the semantic annotations with knowledge-based criterion.

Parser	≤ 40 words			all		
	LP	LR	F_1	LP	LR	F_1
Chiang and Bikel (2002)	81.1	78.8	79.9	78.0	75.2	76.6
Petrov and Klein (2007)	86.9	85.7	86.3	84.8	81.9	83.3
This Paper	**88.9**	**86.0**	**87.5**	**86.0**	**83.1**	**84.5**

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

Performances on English

In order to verify the effectiveness of our method on other languages, we carried out some experiments on English. HowNet is a common sense knowledge base in Chinese and English, therefore, it was still utilized as the knowledge source in these experiments.

The same three methods were compared on English (sentences of length 40 or less), and the results were showed in Table 4. Compared with the baseline (90.1%), the parsers trained with the semantic annotation, while using different splitting methods introduced in Section 3, achieved an F_1 score of 90.2% and 90.3% respectively. The results showed that our methods could get a small but stable improvements on English (p<.08).

Subcategory Refined from the Original Training Set

PN-0	**援外(aid foreign)**, *叔婶(aunt)*, *自家(self)*, *汝(you)*, **予(donate)**, 那些(those), 相貌(appearence), *自个儿(self)*, *咱们(we)*, 彼(that), 以上(the above), 那边(there), 其他(other), 以下(below)

Subcategories Fefined from the Good Starting Annotations

PN-0	*叔婶(aunt)*, *自家(self)*, *自个儿(self)*, *咱们(we)*, *汝(you)*
PN-Event-0	**援外(aid foreign)**, **予(donate)**
PN-AttributeValue-2	以上(the above), 那些(those), 彼(that), 其他(other), 以下(below)

Table 5: Several subcategories that generated from the original training set and the good starting annotations respectively.

Method	F_1
Baseline	90.1
Semantic Annotations	90.2
Semantic Annotations & Knowledge-based Criterion	**90.3**

Table 4: Performances at the fifth iteration on English (sentences of length 40 or less) with three methods: the baseline, the parser trained on the semantic annotations with automatic method, and the parser trained on the semantic annotations with knowledge-based criterion.

These results on English were preliminary, and we did not introduce any language dependent operation such as morphological processing. Since only the lemma of English words can be found in HowNet, we just annotated two kinds of POS tags "VB"(Verb, base form) and "NN"(Noun, singular or mass) with semantic knowledge, on the contrary, we annotated almost all POS tags whose corresponding words could be found in HowNet on Chinese. This might be the reason that the improvement on the English Treebank was much smaller than that of Chinese. It is expected to achieve more improvements through some morphological analysis in the future.

4.4 Results and Analysis

So far, a new strategy has been introduced to refine the grammars in two steps, and achieved significant improvements on parsing performance. In this section, we analyze the grammars learned at different steps, attempting to explain how the semantic knowledge works.

It is hard to inspect all the grammars by hand. Since the semantic knowledge is mainly used for generating and splitting new semantic-related tags in our method, we focus on the refined subcategories of these tags.

First, we examine the refined subcategories of POS tags, which are generated from the original training set and the good starting annotations respectively. Several subcategories are listed and compared in Table 5, along with their frequent words. It can be seen that the subcategories refined with semantic knowledge are more consistent than the previous one. For example, the subcategory "PN-0", which is refined from the original training set, produces a lot of words without semantic consistence. In contrast, we refine the subcategories "PN-0", "PN-Event-0" and "PN-AttributeValue-2" from the good starting annotations. Each of them produces a small but semantic consistent word set.

In order to inspect the difference between the automatic splitting process and the semantic based one, we compare the numbers of subcategories refined in these two processes. Since it is hard to list all the semantic-related tags here, three parts of the semantic-related tags were selected and listed in Table 6, along with the number of their subcategories. The first part is the noun and verb related tags, which are most heavily split in both two processes. It is clear that the semantic based splitting process can generate more subcategories than the automatic one, because the semantic structures of noun and verb are sophisticated. The second part lists the tags that have much more subcategories ($\geq$ 4) from the automatic splitting process than the semantic based one, and the third part vice verse. It can be seen that most of the subcategories in the second part are functional categories, while most of the subcategories in the third part are content categories. It means that the semantic based splitting process is prone to generating less subcategory for the functional categories, but more subcategories for the content categories. This tendency is in accordance with the linguistic intuition. We believe that it is the main effect

Semantic-related tag	Automatic split number	Semantic based split numebr
NN-Attribute	30	30
NN-AttributeValue	25	**27**
NN-Entity	32	32
NN-Event	**31**	30
VV-Attribute	2	2
VV-AttributeValue	27	27
VV-Entity	22	**26**
VV-Event	29	**32**
BA-event	**13**	5
CS-AttributeValue	**29**	16
CS-entity	**22**	15
OD-Attribute	**13**	7
PN-Attribute	**26**	22
AS-AttributeValue	2	**7**
JJ-event	4	**8**
NR-AttributeValue	9	**13**
NT-event	12	**18**
VA-AttributeValue	22	**27**
VA-event	7	**11**

Table 6: The number of subcategories learned from two approaches: the automatic hierarchical state-splitting, and the semantic based splitting.

of our knowledge-based criterion, because it adjusts the splitting results dynamically with semantic knowledge, which can alleviate the overfitting risk.

5 Conclusions

In this paper, we present a novel approach to integrate semantic knowledge into the hierarchical state-split process for grammar refinement, which yields better accuracies on Chinese than previous methods. The improvements are mainly owing to two aspects. Firstly, the original treebank is initialized by annotating the tag of each POS node with the most general hypernym of its terminal word, which reduces the search space for the EM algorithm and brings an initial restrict to the following splitting step. Secondly, the splitting process is supervised by a knowledge-based criterion with the new semantic-related tags. Benefitting from the hierarchical semantic knowledge, the proposed approach alleviates the overfitting risk in a knowledge-driven manner. Experimental results reveal that the semantic knowledge is of great use to syntactic disambiguation. The further analysis on the refined grammars shows that, our method tends to split the content categories more often than the baseline method and the function classes less often.

Acknowledgments

We thank Yaozhong Zhang for the enlightening discussions. We also thank the anonymous reviewers who gave very helpful comments. The work was supported in part by the National Natural Science Foundation of China (60535030; 60605016), the National High Technology Research and Development Program of China (2006AA010103), the National Key Basic Research Program of China (2004CB318005, 2004CB318105).

References

E. Agirre, T. Baldwin and D. Martinez. 2008. Improving parsing and PP attachment performance with sense information. In *Proc. of ACL'08*, pages 317-325.

D. Bikel. 2000. A statistical model for parsing and word-sense disambiguation. In *Proc. of EMNLP/VLC'2000*, pages 155-163.

D. Bikel. 2004. Intricacies of Collins’ parsing model. *Computational Linguistics*, 30(4):479-511.'

E. Charniak. 1997. Statistical parsing with a context-free grammar and word statistics. In *Proc. of AAAI'97*, pages 598-603.

E. Charniak. 2000. A maximum-entropy-inspired parser. In *Proc. of NAACL'00*, pages 132-139.

E Charniak and M. Johnson. 2005. Coarse-to-fine n-best parsing and maxEnt discriminative reranking. In *Proc. of ACL'05*, pages 173-180.

D. Chiang and D. Bikel. 2002. Recovering latent information in treebanks. In *Proc. of COLING'02*, pages 183-189.

M. Collins. 1997. Three generative, lexicalised models for statistical parsing. In *Proc. of ACL'97*, pages 16-23.

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

Z. Dong and Q. Dong. 2000. HowNet Chinese-English conceptual database. Technical Report Online Software Database, Released at ACL. http://www.keenage.com.

S. Fujita, F. Bond, S. Oepen and T. Tanaka 2007. Exploiting semantic information for HPSG parse selection. *In ACL 2007 Workshop on Deep Linguistic Processing*, pages 25-32.

M. Johnson. 1998. PCFG models of linguistic tree representations. *Computational Linguistics*, 24(4):613-631.

D. Klein and C. Manning. 2003. Accurate unlexicalized parsing. In *Proc. of ACL'03*, pages 423-430.

M. Marcus, B. Santorini, and M. Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313-330.

T. Matsuzaki, Y. Miyao, and J. Tsujii. 2005. Probabilistic CFG with latent annotations. In *Proc. of ACL'05*, pages 75-82.

George A. Miller, Martin Chodorow, Shari Landes, Claudia Leacock, and Robert G. Thomas. 1994. Using a semantic concordance for sense identification. In *Proc. of ARPA-HLT Workshop.*, pages 240-243.

S. Petrov, L. Barrett, R. Thibaux, and D. Klein. 2006. Learning accurate, compact, and interpretable tree annotation. In *Proc. of COLING-ACL'06*, pages 443 – 440.

S. Petrov and D. Klein. 2007. Improved inference for unlexicalized parsing. In *Proc. of HLT-NAACL'07*, pages 404-411.

D. Xiong, S. Li, Q. Liu, S. Lin, and Y. Qian. 2005. Parsing the Penn Chinese treebank with semantic knowledge. In *Proc. of IJCNLP'05*, pages 70-81.

N. Xue, F.-D. Chiou, and M. Palmer. 2002. Building a large scale annotated Chinese corpus. In *Proc. of COLING'02*, pages 1-8.

Y. Zhang. 2008. The Study and Realization of Chinese Parsing with Semantic and Sentence Type Information. Master thesis, Peking University.