Parsing the Penn Chinese Treebank with Semantic Knowledge

Deyi Xiong[1,2], Shuanglong Li[1,3],
Qun Liu[1], Shouxun Lin[1], and Yueliang Qian[1]

[1] Institute of Computing Technology, Chinese Academy of Sciences,
P.O. Box 2704, Beijing 100080, China
{dyxiong, liuqun, sxlin}@ict.ac.cn
[2] Graduate School of Chinese Academy of Sciences
[3] University of Science and Technology Beijing

Abstract. We build a class-based selection preference sub-model to incorporate external semantic knowledge from two Chinese electronic semantic dictionaries. This sub-model is combined with modifier-head generation sub-model. After being optimized on the held out data by the EM algorithm, our improved parser achieves 79.4% (F1 measure), as well as a 4.4% relative decrease in error rate on the Penn Chinese Treebank (CTB). Further analysis of performance improvement indicates that semantic knowledge is helpful for nominal compounds, coordination, and N◇V tagging disambiguation, as well as alleviating the sparseness of information available in treebank.

1 Introduction

In the recent development of full parsing technology, semantic knowledge is seldom used, though it is known to be useful for resolving syntactic ambiguities. The reasons for this may be twofold. The first one is that it can be very difficult to add additional features which are not available in treebanks to generative models like Collins (see [1]), which are very popular for full parsing. For smaller tasks, like prepositional phrase attachment disambiguation, semantic knowledge can be incorporated flexibly using different learning algorithms (see [2,3,4,5]). For full parsing with generative models, however, incorporating semantic knowledge may involve great changes of model structures. The second reason is that semantic knowledge from external dictionaries seems to be noisy, ambiguous and not available in explicit forms, compared with the information from treebanks. Given these two reasons, it seems to be difficult to combine the two different information sources–*treebank* and *semantic knowledge*–into one integrated statistical parsing model.

One feasible way to solve this problem is to keep the original parsing model unchanged and build an additional sub-model to incorporate semantic knowledge from external dictionaries. The modularity afforded by this approach makes it easier to expand or update semantic knowledge sources with the treebank

R. Dale et al. (Eds.): IJCNLP 2005, LNAI 3651, pp. 70–81, 2005.
© Springer-Verlag Berlin Heidelberg 2005

unchanged or vice versa. Further, the combination of the semantic sub-model and the original parsing model can be optimized automatically.

In this paper, we build a class-based selection preference sub-model, which is embedded in our lexicalized parsing model, to incorporate external semantic knowledge. We use two Chinese electronic dictionaries and their combination as our semantic information sources. Several experiments are carried out on the Penn Chinese Treebank to test our hypotheses. The results indicate that a significant improvement in performance is achieved when semantic knowledge is incorporated into parsing model. Further improvement analysis is made. We confirm that semantic knowledge is indeed useful for nominal compounds and coordination ambiguity resolution. And surprisingly, semantic knowledge is also helpful to correct Chinese N$\diamond$V mistagging errors mentioned by Levy and Manning (see [12]). Yet another great benefit to incorporating semantic knowledge is to alleviate the sparseness of information available in treebank.

2 The Baseline Parser

Our baseline parsing model is similar to the history-based, generative and lexicalized Model 1 of Collins (see [1]). In this model, the right hand side of lexicalized rules is decomposed into smaller linguistic objects as follows:

$$P(h) \rightarrow \#L_n(l_n)...L_1(l_1)H(h)R_1(r_1)...R_m(r_m)\# \ .$$

The uppercase letters are delexicalized nonterminals, while the lowercase letters are lexical items, e.g. head word and head tag (part-of-speech tag of the head word), corresponding to delexicalized nonterminals. $H(h)$ is the head constituent of the rule from which the head lexical item h is derived according to some head percolation rules.[1] The special termination symbol # indicates that there is no more symbols to the left/right. Accordingly, the rule probability is factored into three distributions. The first distribution is the probability of generating the syntactic label of the head constituent of a parent node with label P, head word Hhw and head tag Hht:

$$Pr_H(H|P, Hht, Hhw) \ .$$

Then each left/right modifier of head constituent is generated in two steps: first its syntactic label M_i and corresponding head tag M_iht are chosen given context features from the parent (P), head constituent (H, Hht, Hhw), previously generated modifier ($M_{i-1}, M_{i-1}ht$) and other context information like the direction (dir) and distance[2] (dis) to the head constituent:

[1] Here we use the modified head percolation table for Chinese from Xia (see [6]).

[2] Our distance definitions are different for termination symbol and non-termination symbol, which are similar to Klein and Manning (see [7]).

$$Pr_M(M_i, M_iht | HC_M) \ .$$

where the history context HC_M is defined as the joint event of

$$P, H, Hht, Hhw, M_{i-1}, M_{i-1}ht, dir, dis \ .$$

Then the new modifier's head word M_ihw is also generated with the probability:

$$Pr_{M_w}(M_ihw | HC_{M_w}) \ .$$

where the history context HC_{M_w} is defined as the joint event of

$$P, H, Hht, Hhw, M_{i-1}, M_{i-1}ht, dir, dis, M_i, M_iht \ .$$

All the three distributions are smoothed through Witten-Bell interpolation just like Collins (see [1]). For the distribution Pr_M, we build back-off structures with six levels, which are different from Collins' since we find our back-off structures work better than the three-level back-off structures of Collins. For the distribution Pr_{M_w}, the parsing model backs off to the history context with head word Hhw removed, then to the modifier head tag M_iht, just like Collins. Gildea (see [9]) and Bikel (see [10]) both observed that the effect of bilexical dependencies is greatly impaired due to the sparseness of bilexical statistics. Bikel even found that the parser only received an estimate that made use of bilexical statistics a mere 1.49% of the time. However, according to the wisdom of the parsing community, lexical bigrams, the word pairs (M_ihw, Hhw) are very informative with semantic constraints. Along this line, in this paper, we build an additional class-based selectional preference sub-model, which is described in section 3, to make good use of this semantic information through selectional restrictions between head and modifier words.

Our parser takes segmented but untagged sentences as input. The probability of unknown words, $Pr(uword|tag)$, is estimated based on the first character of the word and if the first characters are unseen, the probability is estimated by absolute discounting.

We do some linguistically motivated re-annotations for the baseline parser. The first one is marking non-recursive noun phrases from other common noun phrases without introducing any extra unary levels (see [1,8]). We find this basic NP re-annotation very helpful for the performance. We think it is because of the annotation style of the Upenn Chinese Treebank (CTB). According to Xue et al. (see [11]), noun-noun compounds formed by an uninterrupted sequence of words POS-tagged as NNs are always left flat because of difficulties in determining which modifies which. The second re-annotation is marking basic VPs, which we think is beneficial for reducing multilevel VP adjunction ambiguities (see [12]).

To speed up parsing, we use the beam thresholding techniques in Xiong et al. (see [13]). In all cases, the thresholding for completed edges is set at $ct = 9$ and incomplete edges at $it = 7$. The performance of the baseline parser is 78.5% in terms of F1 measure of labeled parse constituents on the same CTB training and test sets with Bikel et al. (see [14])

3 Incorporating Semantic Knowledge

In this section, we describe how to incorporate semantic knowledge from external semantic dictionaries into parsing model to improve the performance. Firstly, we extract semantic categories through two Chinese electronic semantic dictionaries and some heuristic rules. Then we build a selection preference sub-model based on extracted semantic categories. In section 3.3, we present our experiments and results in detail. And finally, we compare parses from baseline parser with those from the new parser incorporated with semantic knowledge. We empirically confirm that semantic knowledge is helpful for nominal compound, coordination and POS tagging ambiguity resolution. Additionally, we also find that semantic knowledge can greatly alleviate problems caused by data sparseness.

3.1 Extracting Semantic Categories

Semantic knowledge is not presented in treebanks and therefore has to be extracted from external knowledge sources. We have two Chinese electronic semantic dictionaries, both are good knowledge sources for us to extract semantic categories. One is the HowNet dictionary[3], which covers 67,440 words defined by 2112 different sememes. The other is the "TongYiCi CiLin" expanded version (henceforth CiLin)[4], which represents 77,343 words in a dendrogram.

HowNet (HN): Each sememe defined by the HowNet is regarded as a semantic category. And through the hypernym-hyponym relation between different categories, we can extract semantic categories at various granularity levels. Since words may have different senses, and therefore different definitions in HowNet, we just use the first definition of words in HowNet. At the first level HN1, we extract the first definitions and use them as semantic categories of words. Through the hypernym ladders, we can get HN2, HN3, by replacing categories at lower level with their hypernyms at higher level. Table 1 shows information about words and extracted categories at different levels.

CiLin (CL): CL is a branching diagram, where each node represents a semantic category. There are three levels in total, and from the top down, 12 categories in the first level (CL1), 97 categories in the second level (CL2), 1400 categories in the third level (CL3). We extract semantic categories at level CL1, CL2 and CL3.

HowNet+CiLin: Since the two dictionaries have different ontologies and representations of semantic categories, we establish a strategy to combine them: HowNet is used as a primary dictionary, and CiLin as a secondary dictionary. If a word is not found in HowNet but found in Cilin, we will look up other words from its synset defined by CiLin in HowNet. If HowNet query succeeds, the corresponding semantic category in HowNet will be assigned to this word.

[3] http://www.keenage.com/.

[4] The dictionary is recorded and expanded by Information Retrieval Laboratory, Harbin Institute of Technology.

Table 1. Sizes and coverage of words and semantic categories from different semantic knowledge sources

	Data	HN1	HN2	HN3	CL1	CL2	CL3
words in train	9522	6040			6469		
words in test	1824	1538			1581		
words in both	1412	1293			1310		
classes in train	-	1054	381	118	12	92	1033
classes in test	-	520	251	93	12	79	569
classes in both	-	504	248	93	12	79	552

According to our experimental results, we choose HN2 as the primary semantic category set and combine it with CL1, CL2 and CL3.

Heuristic Rules (HR): Numbers and time expressions are recognized using simple heuristic rules. For a better recognition, one can define accurate regular expressions. However, we just collect suffixes and feature characters to match strings. For example, Chinese numbers are strings whose characters all come from a predefined set. These two classes are merged into HowNet and labelled by semantic categories from HowNet.

In our experiments, we combine HN2, CL1/2/3, and HR as our external sources. In these combinations {HN2+CL1/2/3+HR}, all semantic classes come from the primary semantic category set HN2, therefore we get the same class coverage that we obtain from the single source HN2 but a bigger word coverage. The number of covered words of these combinations in {*train, test, both*} is {7911, 1672, 1372} respectively.

3.2 Building Class-Based Selection Preference Sub-model

There are several ways to incorporate semantic knowledge into parsing model. Bikel (see [15]) suggested a way to capture semantic preferences by employing bilexical-class statistics, in other words, dependencies among head-modifier word classes. Bikel did not carry it out and therefore greater details are not available. However, the key point, we think, is to use classes extracted from semantic dictionary, instead of words, to model semantic dependencies between head and modifier. Accordingly, we build a similar bilexical-class sub-model as follows:

$$Pr_{class}(C_{M_i hw}|C_{Hhw}, Hht, M_i ht, dir) \ .$$

where $C_{M_i hw}$ and C_{Hhw} represent semantic categories of words $M_i hw$ and Hhw, respectively. This model is combined with sub-model Pr_{M_w} to form a mixture model P_{mix}:

$$Pr_{mix} = \lambda Pr_{M_w} + (1 - \lambda) Pr_{class} \ . \tag{1}$$

λ is hand-optimized, and an improvement of about 0.5% in terms of F1 measure is gained. However, even a very slight change in the value of λ, e.g. 0.001, will have a great effect on the performance. Besides, it seems that the connection between entropy, i.e. the total negative logarithm of the inside probability of trees, and F1 measure, is lost, while this relation is observed in many experiments. Therefore, automatic optimization algorithms, like EM, can not work in this mixture model. The reason, we guess, is that biclass dependencies among head-modifier word classes seem too coarse-grained to capture semantic preferences between head and modifier. In most cases, a head word has a strong semantic constraints on the concept κ of *mw*, one of its modifier words, but that doesn't mean other words in the same class with the head word has the same semantic preferences on the concept κ. For example, the verb *eat* impose a selection restriction on its object modifier[5]: it has to be solid food. On the other hand, the verb *drink* specifies its object modifier to be liquid beverage. At the level HN2, verb *eat* and *drink* have the same semantic category *metabolize*. However, they impose different selection preferences on their PATIENT roles.

To sum up, bilexical dependencies are too fine-grained when being used to capture semantic preferences and therefore lead to serious data sparseness. Biclass dependencies, which result in an unstable performance improvement, on the other hand, seem to be too coarse-grained for semantic preferences. We build a class-based selection preference model:

$$Pr_{sel}(C_{M_i hw}|Hhw, P) \ .$$

This model is similar to Resnik (see [2]). We use the parent node label P to represent the grammatical relation between head and modifier. Besides, in this model, only modifier word is replaced with its semantic category. The dependencies between head word and modifier word class seem to be just right for capturing these semantic preferences.

The final mixture model is the combination of the class-based selection preference sub-model Pr_{sel} and modifier-head generation sub-model Pr_{M_w}:

$$Pr_{mix} = \lambda Pr_{M_w} + (1-\lambda) Pr_{sel} \ . \tag{2}$$

Since the connection between entropy and F1 measure is observed again, EM algorithm is used to optimize λ. Just like Levy (see [12]), we set aside articles 1-25 in CTB as held out data for EM algorithm and use articles 26-270 as training data during λ optimization.

3.3 Experimental Results

We have designed several experiments to check the power of our class-based selection preference model with different semantic data sources. In all experiments, we first use the EM algorithm to optimize the parameter λ. As mentioned above, during parameter optimization, articles 1-25 are used as held out data and articles 26-270 are used as training data. Then we test our mixture model with optimized parameter λ using the training data of articles 1-270 and test data of articles 271-300 of length at most 40 words.

[5] According to Thematic Role theory, this modifier has a PATIENT role.

Table 2. Results for incorporating different semantic knowledge sources. The baseline parser is described in Sect. 2. in detail.

	Baseline	HN1	HN2	HN3	CL1	CL2	CL3
F1(%)	78.5	78.6	79.1	78.9	77.5	78.7	78.8

Table 3. Results for combinations of different semantic knowledge sources

	Baseline	HN2+CL1+HR	HN2+CL2+HR	HN2+CL3+HR
F1(%)	78.5	79.2	79.4	79.3

Firstly, we carry out experiments on HowNet and CiLin, separately. Experimental results are presented in Table 2. As can be seen, CiLin has a greater coverage of words than that of HowNet, however, it works worse than HowNet. And at the level CL1, coarse-grained classes even yield degraded results. It's difficult to explain this, but the main reason may be that HowNet has a fine-grained and substantial ontology while CiLin is designed only as a synset container.

Since HowNet has a better semantic representation and CiLin better coverage, we want to combine them. The combination is described in Sect. 3.1, where HN2 is used as the primary semantic category set. Words found by CiLin and heuristic rules are labelled by semantic categories from HN2. Results are shown in Table 3. Although external sources HN2+CL1/2/3+HR have the identical word coverage and yield exactly the same number of classes, the different word-class distributions in them lead to the different results.

Due to the combination of HN2, CL2 and HR, we see that our new parser with external semantic knowledge outperforms the baseline parser by 0.9% in F1 measure. Given we are already at the 78% level of accuracy, an improvement of 0.9% is well worth obtaining and confirms the importance of semantic dependencies on parsing. Further, we do the significance test using Bikel's significance tester[6] which is modified to output p-value for F1. The significance level for F-score is at most $(43376 + 1)/(1048576 + 1) = 0.041$. A second 1048576 iteration produces the similar result. Therefore the improvement is statistically significant.

3.4 Performance Improvement Analysis

We manually analyze parsing errors of the baseline parser (BP) as well as performance improvement of the new parser (IP) with semantic knowledge from the combination of HN2, CL2 and HR. Improvement analysis can provide an additional valuable perspective: how semantic knowledge helps to resolve some ambiguities. We compare BP and IP on the test data parse by parse. There are 299 sentences of length at most 40 words among the total 348 test sentences. The two parsers BP and IP found different parses for 102 sentences, among which

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

Table 4. Frequency of parsing improvement types. *AR* represents *ambiguity resolution.*

Type	Count	Percent(%)
Nominal Compound *AR*	19	38
Coordination *AR*	9	18
N◇V *AR* in *N◇V+noun*	6	12
Other *AR*	16	32

IP yields better parse trees for 47 sentences according to the gold standard trees. We have concentrated on these 47 sentences and compared parse trees found by *IP* with those found by *BP*. Frequencies of major types of parsing improvement is presented in Table 4. Levy and Manning (see [12])(henceforth L&M) observed the top three parsing error types: NP-NP modification, Coordination and N◇V mistagging, which are also common in our baseline parser. As can be seen, our improved parser can address these types of ambiguities to some extent through semantic knowledge.

Nominal Compounds (NCs) Disambiguation: Nominal compounds are notorious "every way ambiguous" constructions.[7] The different semantic interpretations have different dependency structures. According to L&M, this ambiguity will be addressed by the dependency model when word frequencies are large enough to be reliable. However, even for the treebank central to a certain topic, many very plausible dependencies occur only once.[8] A good technique for resolving this conflict is to generalize the dependencies from word pairs to word-class pairs. Such generalized dependencies, as noted in section 3.2, can capture semantic preferences, as well as alleviate the data sparseness associated with standard bilexical statistics. In our class-based selection preference model, if the frequency of pair $[C_{Mhw}, Hhw]$[9] is large enough, the parser can interpret nominal compounds correctly, that is, it can tell which modify which.

NCs are always parsed as flatter structures by our baseline parser, just like the tree a. in Figure 1. This is partly because of the annotation style of CTB, where there is no NP-internal structure. For these NCs without internal analysis, we re-annotated them as basic NPs with label NPB, as mentioned in section 2. This re-annotation really helps. Another reason is that the baseline parser, or the modifier word generating sub-model P_{M_w}, can not capture hierarchical semantic dependencies of internal structures of NCs due to the sparseness of bilexical dependencies. In our new parser, however, the selection preference model is able to build semantically preferable structures through word-class dependency statistics. For NCs like (n_1, n_2, n_3), where n_i is a noun, dependency structures

[7] "Every way ambiguous" constructions are those for which the number of analyses is the number of binary trees over the terminal elements. Prepositional phrase attachment, coordination, and nominal compounds are all "every way ambiguous" constructions.

[8] Just as Klein et al. (see [8]) said, one million words of training data just isn't enough.

[9] Henceforth, $[s_1, s_2]$ denotes a dependency structure, where s_1 is a modifier word or its semantic class (C), and s_2 is the head word.

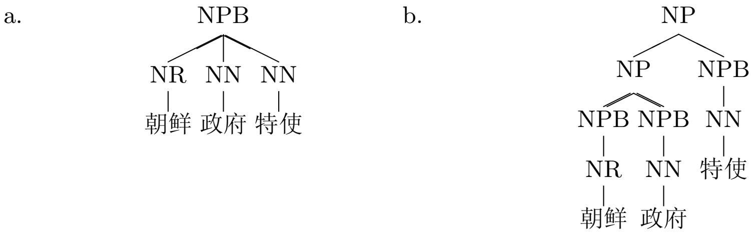

Fig. 1. Nominal Compounds: The North Korean government's special envoy. a. is the incorrect flat parse, b. is the right one in corpus

$\{[C_{n_1}, n_2], [C_{n_1}, n_3], [C_{n_2}, n_3]\}$ will be checked in terms of semantic acceptability and semantically preferable structures will be built finally. For more complicated NCs, similar analysis follows.

In our example (see Fig. 1.), the counts of word dependencies *[朝鲜/North Korea, 政府/government]* and *[朝鲜/North Korea,特使/special envoy]* in the training data both are 0. Therefore, it is impossible for the baseline parser to have a preference between these two dependency structures. On the other hand, the counts of word-class dependencies *[来源值,政府/government]*, where 来源值 is the semantic category of 朝鲜 in HN2, is much larger than the counts of *[来源值,特使/special envoy]* and *[组织,特使/special envoy]*, where 组织 is the semantic category of 政府 in the training data. Therefore, the dependency structure of *[朝鲜/North Korea, 政府/government]* will be built.

Coordination Disambiguation: Coordination is another kind of "every way ambiguous" construction. For coordination structures, the head word is meaningless. But that doesn't matter, since semantic dependency between the spurious head and modifier will be used to measure the meaning similarity of coordinated structures. Therefore, our selection preference model still works in coordination constructions. We have also found VP coordination ambiguity, which is similar to that observed by L&M. The latter VP in coordinated VPs is often parsed as an IP due to *pro*-drop by the baseline parser. That is, the coordinated structure *VP* is parsed as: $VP^0 \rightarrow VP^1 IP^2$. This parse will be penalized by the selection preference model because the hypothesis that the head word of IP^2 has a similar meaning to the head word of VP^1 under the grammatical relation VP^0 is infrequent.

N◇V-ambiguous Tagging Disambiguation: The lack of overt morphological marking for transforming verbal words to nominal words in Chinese results in ambiguity between these two categories. L&M argued that the way to resolve this ambiguity is to look at more external context, like some function words, e.g. adverbial or prenominal modifiers, co-occurring with N◇V-ambiguous words. However, in some cases, N◇V-ambiguous words can be tagged correctly without external context. Chen et al. (see [16]) studied the pattern of *N◇V+noun*, which will be analyzed as a *predicate-object* structure if *N◇V* is a verb and a *modifier-noun* structure if *N◇V* is a noun. They found that in most cases, this pattern can

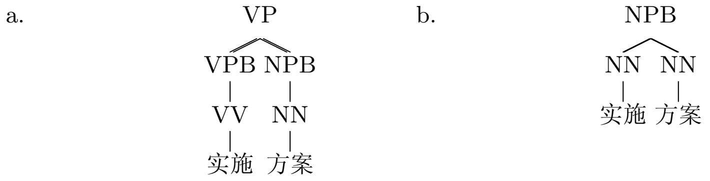

Fig. 2. N◇V-ambiguity: a. implement plans (incorrect parse) versus b. implementation plans (corpus)

Table 5. Previous Results on CTB parsing for sentences of length at most 40 words

	LP	LR	F1
Bikel and Chiang 2000	77.2	76.2	76.7
Levy and Manning 2003	78.4	79.2	78.8
Present work	80.1	78.7	79.4
Bikel Thesis 2004	81.2	78.0	79.6
Chiang and Bikel 2002	81.1	78.8	79.9

be parsed correctly without any external context. Furthermore, they argued that semantic preferences are helpful for the resolution of ambiguity between these two different structures. In our selection preference model, semantic preferences interweave with grammatical relations. These semantic dependencies impose constraints on the structure of the pattern *N◇V+noun* and therefore on the POS tag of *N◇V*. Figure 2 shows our new parser can correct N◇V mistagging errors occurring in the pattern of *N◇V+noun*.

Smoothing: Besides the three ambiguity resolution noted above, semantic knowledge indeed helps alleviate the fundamental sparseness of the lexical dependency information available in the CTB. For many word pairs *[mod,head]*, whose count information is not available in the training data, the dependency statistics of head and modifier can still work through the semantic category of *mod*. During our manual analysis of performance improvement, many other structural ambiguities are addressed due to the smoothing function of semantic knowledge.

4 Related Work on CTB Parsing

Previous work on CTB parsing and their results are shown in table 5. Bikel and Chiang (see [14]) used two different models on CTB, one based on the modified BBN model which is very similar to our baseline model, the other on Tree Insertion Grammar (TIG). While our baseline model used the same unknown word threshold with Bikel and Chiang but smaller beam width, our result outperforms theirs due to other features like distance, basic NP re-annotation used by our baseline model. Levy and Manning (see [12]) used a factored model with rich re-annotations guided by error analysis. In the baseline model, we also used several re-annotations but find most re-annotations they suggested do not fit our model. The three parsing error types expounded above are also found by L&M. However, we used more efficient measures to keep our improved model from these errors.

The work of Bikel thesis (see [10]) emulated Collins' model and created a language package to Chinese parsing. He used subcat frames and an additional POS tagger for unseen words. Chiang and Bikel (see [17]) used the EM algorithm on the same TIG-parser to improve the head percolation table for Chinese parsing. Both these two parsers used fine-tuned features recovered from the treebank that our model does not use. This leads to better results and indicates that there is still room of improvement for our model.

5 Conclusions

We have shown that how semantic knowledge may be incorporated into a generative model for full parsing, which reaches 79.4% in CTB. Experimental results are quite consistent with our intuition. After the manual analysis of performance improvement, the working mechanism of semantic knowledge in the selection preference model is quite clear:

1. Using semantic categories extracted from external dictionaries, the class-based selection preference model first generalizes standard bilexical dependencies, some of which are not available in training data, to word-class dependencies. These dependencies are neither too fine-grained nor too coarse-grained compared with bilexical and biclass dependencies, and really help to alleviate fundamental information sparseness in treebank.
2. Based on the generalized word-class pairs, semantic dependencies are captured and used to address different kinds of ambiguities, like nominal compounds, coordination construction, even N◇V-ambiguous words tagging.

Our experiments show that generative models have room for improvement by employing semantic knowledge. And that may be also true for discriminative models, since these models can easily incorporate richer features in a well-founded fashion. This is the subject of our future work.

Acknowledgements

This work was supported in part by National High Technology Research and Development Program under grant #2001AA114010 and #2003AA111010. We would like to acknowledge anonymous reviewers who provided helpful suggestions.

References

1. Michael Collins. 1999. Head-Driven Statistical Models for Natural Language Parsing. PhD thesis, University of Pennsylvania.
2. Philip Stuart Resnik. 1993. Selection and Information: A Class-Based Approach to Lexical Relationships. PhD thesis, University of Pennsylvania, Philadelphia, PA, USA.

3. Sanda Harabagiu. 1996. An Application of WordNet to Prepositional Attachement. In *Proceedings of ACL-96*, June 1996, Santa Cruz CA, pages 360-363.
4. Yuval Krymolowski and Dan Roth. 1998. Incorporating Knowledge in Natural Language Learning: A Case Study. In *COLING-ACL'98 Workshop on Usage of WordNet in Natural Language Processing Systems*,Montreal, Canada.
5. Mark McLauchlan. 2004. Thesauruses for Prepositional Phrase Attachment. In *Proceedings of CoNLL-2004*,Boston, MA, USA, 2004, pp. 73-80.
6. Fei Xia. 1999. Automatic Grammar Generation from Two Different Perspectives. PhD thesis, University of Pennsylvania.
7. Dan Klein and Christopher D. Manning. 2002. Fast Exact Natural Language Parsing with a Factored Model. In *Advances in Neural Information Processing Systems 15 (NIPS-2002).*
8. Dan Klein and Christopher D. Manning. 2003. Accurate Unlexicalized Parsing. In *Proceedings of ACL-03.*
9. Daniel Gildea. 2001. Corpus variation and parser performance. In *Proceedings of EMNLP-01*, Pittsburgh, Pennsylvania.
10. Daniel M. Bikel. 2004a. On the Parameter Space of Generative Lexicalized Statistical Parsing Models. PhD thesis, University of Pennsylvania.
11. Nianwen Xue and Fei Xia. 2000. The Bracketing Guidelines for Chinese Treebank Project. Technical Report IRCS 00-08, University of Pennsylvania.
12. Roger Levy and Christopher Manning. 2003. Is it harder to parse Chinese, or the Chinese Treebank? In *Proceedings of ACL-03.*
13. Deyi Xiong, Qun Liu and Shouxun Lin. 2005. Lexicalized Beam Thresholding Parsing with Prior and Boundary Estimates. In *Proceedings of the 6th Conference on Intelligent Text Processing and Computational Linguistics (CICLing)*, Mexico City, Mexico, 2005.
14. Daniel M. Bikel and David Chiang. 2000. Two statistical parsing models applied to the chinese treebank. In *Proceedings of the Second Chinese Language Processing Workshop*, pages 1-6.
15. Daniel M. Bikel. 2004b. Intricacies of Collins' Parsing Model. to appear in *Computational Linguistics.*
16. Kejian Chen and Weimei Hong. 1996. Resolving Ambiguities of Predicate-object and Modifier-noun Structures for Chinese V-N Patterns. in Chinese. In *Communication of COLIPS*, Vol.6, #2, pages 73-79.
17. David Chiang and Daniel M. Bikel. 2002. Recovering Latent Information in Treebanks. In *proceedings of COLING*,2002.