Improving Dependency Parsing with Fined-Grained Features

Guangyou Zhou, Li Cai, Kang Liu, and Jun Zhao
National Laboratory of Pattern Recognition
Institute of Automation, Chinese Academy of Sciences
95 Zhongguancun East Road, Beijing 100190, China
{gyzhou,lcai,kliu,jzhao}@nlpr.ia.ac.cn

Abstract

In this paper, we present a simple and effective fine-grained feature generation scheme for dependency parsing. We focus on the problem of grammar representation, introducing fine-grained features by splitting various POS tags to different degrees using HowNet hierarchical semantic knowledge. To prevent the oversplitting, we adopt a threshold-constrained bottom-up strategy to merge the derived subcategories. We conduct the experiments on the Penn Chinese Treebank. The results show that, with the fine-grained features, we can improve the dependency parsing accuracies by 0.52% (absolute) for the unlabeled first-order parser, and in the case of second-order parser, we can improve the dependency parsing accuracies by 0.61% (absolute).

1 Introduction

In natural language parsing, part-of-speech (POS) information is seen as crucial to resolving ambiguous relationships, yet POS tags are usually too general to encapsulate a word's syntactic behavior. It is therefore attractive to consider intermediate entities which exist at a finer level than the POS tags, and the relationship between specific words and their syntactic contexts may be best modeled.

In this paper, we introduce the fine-grained features by splitting various POS tags to different degrees. First, we split the POS tags of each word in the Treebank using HowNet hypernym-hyponymy hierarchical semantic knowledge (Dong and Dong, 2000). Then we adopt a threshold-constrained bottom-up strategy to merge the semantic-related subcategories which are plagued by the oversplitting problems. Finally, we use
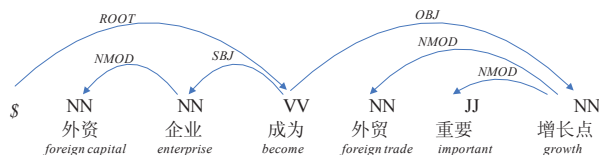

Figure 1: An example of a labeled dependency tree. The tree contains a special token "$" which is always the root of the tree. Each arc is directed from head to modifier and has a label describing the function of the attachment.

the generated sub-categories to construct a new fine-grained feature mapping for a discriminative learner. We are thus relying on the ability of discriminative learning methods to identify and exploit informative features.

To demonstrate the effectiveness of our approach, we conduct the dependency parsing experiments on the Penn Chinese Treebank (CTB) (Xue et al., 2005). The results show that, with the fine-grained features, we can obtain mildly significant improvements both for first-order and second-order parsing (e.g., the absolute improvements are 0.52% and 0.61%, respectively) (see Section 6).

The remainder of this paper is organized as follows. Section 2 introduces the Motivation. Section 3 gives background on dependency parsing and HowNet hierarchical semantic knowledge. Section 4 describes the fine-grained feature generation scheme. Section 5 presents fine-grained features. Experimental evaluation and results are reported in Section 6. Section 7 discusses related work. Finally, in Section 8 we draw conclusion.

2 Motivation

In dependency parsing, we attempt to build head-modifier (or head-dependent) relations between words in a sentence. A simple example is shown in Figure 1, where NN, VV, and JJ are POS tags.

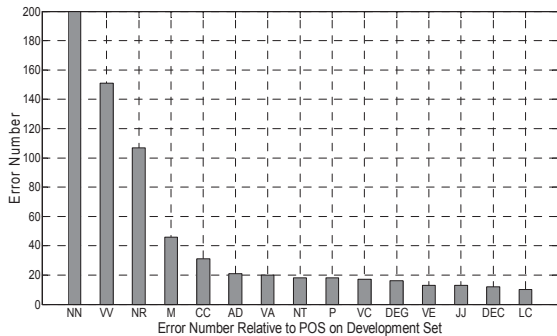Currently, a variety of statistical methods have

228

Figure 2: The number of the most frequent errors relative to POS types on development set for first-order parsing.

been developed for dependency parsing, such as **graph-based** (McDonald et al., 2005; McDonald and Pereira, 2006), **transition-based** (Yamada and Matsumoto, 2003; Hall et al., 2006), or **hybrid methods** (Nivre and McDonald, 2008; Martins et al., 2008; Zhang and Clark, 2008). These methods mainly rely on the POS information as important features, but the POS tags are usually too general to encapsulate a word's syntactic behavior, especially for Chinese dependency parsing on CTB (e.g., it assumes that all the words with the POS tag NN share the same syntactic behavior). In the limit, each word may well have its own unique syntactic behavior (Petrov and Klein, 2006). However, in practice, given limited data, the relationships between the specific words and their context dependencies may be best modeled at a level finer than the POS tags but coarser than the words themselves. Take the sentence in Figure 1 for example, although the words 外资(foreign capital) and 增长点(growth) have the same POS tag NN, they should have different context dependencies in dependency parsing tree. In HowNet, the two words are defined with different hypernyms. The word 外资(foreign capital) is defined as a kind of objective things, while the word 增长点(growth) is defined as an event role feature. Intuitively, the different senses can represent their different syntactic behavior, and we attempt to split the POS tags to different degrees based on hierarchical semantic knowledge.

Figure 2 shows the number of the most frequent errors relative to POS types on the development set for the first-order parsing. From the figure, it is seen that the main errors are nominal and verbal categories. Therefore, we may suspect that whether the complex and frequent categories like

NN and VV should be split heavily while barely split rare or simple ones. Our experiments demonstrate that this strategy can be quite effective in Chinese dependency parsing task (see Table 2 in Section 4 for empirical results).

3 Background

3.1 Dependency Parsing

In dependency parsing, we attempt to build head-modifier (or head-dependent) relations between words in a sentence. The discriminative parser we used in this paper is based on the *part-factored* model and features of the MSTParser (McDonald et al., 2005; McDonald and Pereira, 2006; Carreras, 2007). The parsing model can be defined as a conditional distribution $p(y|\mathbf{x}; \mathbf{w})$ over each projective parse tree y for a particular sentence $\mathbf{x}$, parameterized by a vector $\mathbf{w}$. The probability of a parse tree is

$$p(y|\mathbf{x}; \mathbf{w}) = \frac{1}{Z(\mathbf{x}; \mathbf{w})} \exp\left\{ \sum_{\rho \in y} \mathbf{w} \cdot \mathbf{\Phi}(\mathbf{x}, \rho) \right\} \quad (1)$$

where $Z(\mathbf{x}; \mathbf{w})$ is the partition function and $\mathbf{\Phi}$ are *part-factored* feature functions that include *head-modifier* parts, *sibling* parts and *grandchild* parts. Given the training set $\{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$, parameter estimation for log-linear models generally resolve around optimization of a regularized conditional log-likelihood objective $\mathbf{w}^* = \arg\min_{\mathbf{w}} L(\mathbf{w})$ where

$$L(\mathbf{w}) = -C \sum_{i=1}^{N} \log p(y_i|\mathbf{x}_i; \mathbf{w}) + \frac{1}{2}||\mathbf{w}||^2 \quad (2)$$

The parameter $C > 0$ is a constant dictating the level of regularization in the model. Since objective function $L(\mathbf{w})$ is smooth and convex, which is convenient for standard gradient-based optimization techniques. In this paper we use the dual exponentiated gradient (EG)[1] descent, which is a particularly effective optimization algorithm for log-linear models (Collins et al., 2008).

3.2 HowNet Semantic Knowledge

HowNet is a bilingual general knowledge-base describing relations between concepts and relations between the attributes of concepts in Chinese and their English equivalents (Gan and Wong, 2000).

HowNet constructs a hierarchical structure of its knowledge base from hypernym-hyponymy

---

[1]http://groups.csail.mit.edu/nlp/egstra/

229

relations. The unit of meaning is called sememe that can not be further decomposed, which can be represented in Chinese and their English equivalents, such as the sememe **fund|资金**. The explicated relations of HowNet include hypernym-hyponymy, synonymy, metonymy, antonymy, part-whole, attribute-host, material-product, dynamic role and concept co-occurrence, and so on. In this paper, we only consider the hypernym-hyponymy relations at different levels of granularities. Since a word may have different senses, and therefore different definitions in HowNet, we just use the first definition as the semantic-related tag of the word. Take the concept 外资(foreign capital) for example, its definition and the hypernym-hyponymy relations are listed below from speciality to generality, which we call the hierarchical semantic information in this paper.

Definition:

$DEF = \{$fund|资金:modifier= $\{$foreign|外国$\}\}$

Hierarchy:

fund|资金→wealth|钱财→artifact|人工物→inanimate|无生物→physical|物质→thing|万物→entity|实体

In the definition, HowNet decomposes the concept into sememes 'fund|资金', 'wealth|钱财', 'artifact|人工物', 'inanimate|无生物', 'physical|物质', 'thing|万物', 'entity|实体'. The sememe appearing in the first position of Definition ('fund|资金') is the categorical attribute, which names the hypernym of the concept 外资(foreign capital). Those sememes appearing in other positions (e.g., 'foreign|外国') are additional attributes, which give more specific information to the concept.

It is clear that the word 外资(foreign capital) has hypernyms from the most special hypernym fund|资金 to the most general hypernym entity|实体 in a hierarchical way.

HowNet contains very limited words, so there are many words which cannot be found in HowNet. In this paper, we extend HowNet with Chinese Knowledge base "TongYiCiLin" (abbreviation: CiLin) (Mei et al., 1983), which represents 77,343 words in a dendrogram (or tree).

CiLin is organized as a hierarchical tree structure, each node represents a semantic category. To balance the words coverage, we extract semantic categories at level 3, which covers 1,400 subcate-

gories.

HowNet and CiLin have different ontologies and representations of semantic categories (Xiong et al., 2005), we combine the two dictionaries: given a word w, if we cannot find in HowNet, but found in CiLin, we try to replace w with a synonym s in the synset defined by CiLin. If the synonym s can be found in HowNet, the corresponding semantic-related tag in HowNet will be assigned to w.

4 Fine-Grained Feature Generation

4.1 Splitting the POS Tags

In this subsection, we split the original POS tags to different degrees based on HowNet hierarchical semantic knowledge. The challenge is how to deal with the problem of polysemous words. Since each word may have multiple senses, and therefore different definitions in HowNet. Following Xiong et al. (2005) and Lin et al. (2009), we just use the first sense to determine the sense of each token instance of a target word (e.g., all token instances of a given word are tagged with the sense that occurs most frequently in HowNet).

As mentioned in Section 3.2, the semantic information of each word can be represented as hierarchical hypernym-hyponymy relations. In this paper, we attempt to establish the mapping from top to down and split the words into different subcategories based on hypernym-hyponymy relations defined in HowNet. For easy explanation of the splitting process, we take the words with POS tag NN for example; the fine-grained feature generation is shown in Figure 3. The left part of the figure is the word subcategories, which is split based on HowNet hierarchy. As shown by the dashed line from left to right, we generate each subcategory with the hierarchical semantic-related tag, such as NN-event, NN-entity, NN-thing, NN-time and so on. If the hypernym node has no hyponym, the corresponding subcategory will stop splitting (e.g., at the level 3 in figure 3, "fruit" is the most speciality hypernym of the corresponding words "banana" and "apple" in HowNet hierarchy, which cannot be further decomposed). The details of HowNet hierarchy were presented in Dong and Dong (2000).

As shown in Figure 3, the original relationships between the words and their syntactic contexts are modeled by the POS tag NN, after hierarchically split, the relationships can be best modeled at the
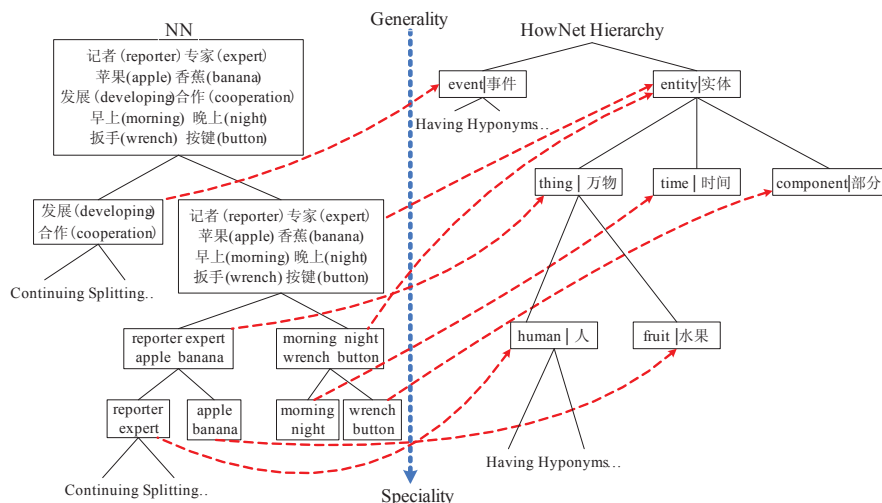
Figure 3: Fine-grained feature generation for words with POS tag NN. The left part is word subcategory and the right part is HowNet hierarchy from generality to speciality.
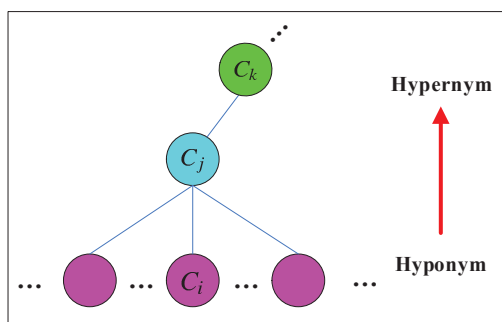

Figure 4: The merging procedure based on hypernym-hyponymy relations from bottom-up.

different levels of the fine-grained subcategories. By this observation, the fine-grained feature generation is just a hierarchical clustering of words themselves with the fine-grained semantic-related tags. Unlike the previous work, such as word cluster technique (Koo et al., 2008), data-driven split manner (Matsuzaki et al., 2005; Petrov and Klein, 2006), our approach does not exploit unlabeled data, and the splitting is based on hierarchical semantic knowledge instead of maximizing posterior probability, which is much simpler than their methods.

4.2 Merging Based on Threshold Constraint

Intuitively, creating more subcategories can increase parsing accuracy. On the other hand, oversplitting can be a serious problem, the details were presented in Klein and Manning (2003). To prevent oversplitting, we merge the subcategories based on the threshold constraint. After the splitting, each subcategory contains a group of

words which share the same semantic-related tag. Then we measure the size of each subcategory to determine whether the subcategory should be further merged. For easy explanation, we show an example in Figure 4, where each node C_i denotes a subcategory, C_j is the nearest hypernym subcategory of C_i, C_k is the nearest hypernym subcategory of C_j, and so on. Assuming that $f(C_i)$, $f(C_j)$ and $f(C_j)$ denote the number of the words contained in the subcategory C_i, C_j and C_k respectively, f is the threshold. We judge C_i should be further merged into C_j if $f(C_i) < f$, and update the number of the words contained in C_j using the following formula:

$$f^{update}(C_j) = \begin{cases} f(C_i) + f(C_j) & \text{if } f(C_i) < f \\ f(C_j) & \text{other} \end{cases}$$

(3)

where $f^{update}(C_j)$ is the number of the words contained in the updated subcategory C_j. In this way, we repeatedly merge each subcategory from bottom-up through the hypernym ladders according to the formula (3). Finally, we generate appropriate granularity of the fine-grained subcategories by splitting and merging approach.

In our approach, each POS tag is divided into several subcategories. The subcategories of some POS tags with the words are shown in Table 1. The categories compose of the original POS tags and the subcategories derived from HowNet. For example, NN is split into NN-InstitutePlace, NN-aValue, and so on. The subcategories' number of each POS tag is shown in Table 2. Nominal categories are the most heavily split. For ex-

NN		VV	
NN-InstitutePlace	企业(enterprise) 公司(company)	VV-event	猜到(guess) 预见(foresee)
NN-aValue	经济(economy) 国际(international)	VV-aValue	小心(care) 可以(can)
NN-organization	国家(country) 政府(government)	VV-SelfMoveInDirection	进行(conduct) 扩散(spread)
NN-event	发展(developing) 合作(cooperation)	VV-change	增长(increase) 涨价(deform)
NN-human	记者(reporter) 专家(expert)	VV-attribute	简称(abbreviation) 库容(storage capacity)
NN-affairs	贸易(trading) 金融(financial)	VV-entity	经历(experience) 考虑(consider)
NN-mental	情绪(mood) 感受(feelings)	VV-AlterRelation	围困(siege) 脱离(separate)
NN-entity	后者(latter) 机会(opportunity)	VV-AlterPossession	借用(borrow) 购进(buy)
NN-artifact	棉花(cotton) 维生素(vitamin)	VV-AlterPhysical	建造(build) 制成(make)
· · ·	· · ·	· · ·	· · ·
AD		**JJ**	
AD-aValue	以后(after) 唯有(only)	JJ-aValue	共同(together) 特别(special)
AD-event	还(also) 不管(no matter)	JJ-event	继续(continue) 相对(relatively)
· · ·	· · ·	· · ·	· · ·

Table 1: The two words with their English translations in the subcategories of some POS tags.

NN	24	VC	2	MSP	1
VV	17	VE	1	OD	1
NR	5	ON	1	DEV	1
JJ	8	P	4	BA	1
CC	7	NT	3	LJ	1
DEG	1	CS	3	LB	1
M	5	AD	5	DER	1
VA	4	SB	1	SP	1
LC	1	CD	1	IJ	1
PN	1	DEC	1	ETC	1
DT	1	AS	1	PU	1

Table 2: The number of subcategories generated by our hierarchical semantic knowledge based split-merge procedure.

ample, common noun (NN) category is divided into the maximum number of subcategories (24). One subcategory consists primarily of objective things, whose typical semantic knowledge is an entity. Another subcategory is defined as an attribute, and so on. These kinds of semantic-related subcategories are typical, and give a division similar to the distributional clustering results like those of Schuetze (1998). The proper noun (NR) category is split into the 5 subcategories, including entity, institute-Places, attribute, aValue, and so on, which are defined in HowNet. The temporal noun (NT) category is also split into 3 subcategories.

Verbal categories are also heavily split. Verbal subcategories sometimes reflect syntactic selectional preferences, and sometimes reflect other aspects of verbal syntax (Petrov and Klein, 2006). For example, the common verb (VV) category is divided into the number of 17 subcategories based on hierarchical split-merge procedure. The predictive adjective (VA) category is also split into 4 subcategories.

Functional categories generally have fewer splits shown in Table 2. Intuitively, those categories are known to be strongly correlated with syntactic behavior. For example, determiner (DT), interjection (IJ), onomatopoeia (ON), and so on.

5 Feature Design

Key to the success of our approach is the use of HowNet hierarchical semantic knowledge to generate the fine-grained features to assist the dependency parsers. The feature sets we used in this paper are similar to other feature sets in the literature (McDonald et al., 2005; McDonald and Pereira, 2006; Carreras, 2007), so we will not attempt to give an exhaustive description of the features in this Section. Rather, we describe our fine-grained features at a high level and concentrate on our motivations. In the experiments, our employed two different feature sets: a baseline feature set which draws upon "normal" information sources such as word forms and POS, and a fine-grained feature set that also information derived from the HowNet hierarchical semantic knowledge.

Our first-order baseline feature set is similar to the feature set of McDonald et al. (2005) and McDonald and Pereira (2006). The second-order baseline features are the same as those of Carreras (2007) and include indicators for triples of POS tags for sibling interactions and grandparent interactions, as well as additional bigram features based on pairs of words involved these higher order interactions.

The first- and second-order fine-grained features are complementary with the baseline features. We generate the fine-grained features by mimicking the word-to-tag and tag-to-tag interactions between the head and modifier of a dependency. Also, We include indicators for triples of fine-grained subcategory tags for sibling and grandparent interactions. Examples of these features are provided in Table 3.

Till now, we have demonstrated our fine-grained generation scheme using HowNet hierarchical semantic knowledge. With the derived subcategories, we can construct a new fine-grained

Baseline	Fine-grained
ht, mt	hf, mf
hw, mw	hw, mf
hw, ht, mt	hf, mw
hw, ht, mw, mt	hw, hf, mf
ht, mt, gt	hw, hf, mw, mf
ht, mt, st	hf, mf, gf
ht, mt, gt, st	hf, mf, sf
. . .	hf, mf, gf
gt, ht, mt, st	. . .
	gf, hf, mf, sf

Table 3: Baseline (left) and fine-grained (right) feature templates. Abbreviation: ht=head POS, hw= head word, hf=fine-grained POS of head, mf=fineg-grained POS of modifier. st, gt, sf, gf= likewise for sibling and grandchild.

feature mapping for a discriminative learner, similar to (Koo et al., 2008). We are relying on the ability of discriminative learning methods to identify and exploit informative features.

6 Experiments

In order to evaluate the effectiveness of the proposed approach, we conducted dependency parsing experiments in Chinese. The experiments were performed on the Penn Chinese Treebank (CTB) version 5.0 (Xue et al., 2005), using a set of head-selection rules (Zhang and Clark, 2008) to convert the phrase structure syntax of the Treebank to a dependency tree representation, dependency labels were obtained via the "Malt" hard-coded setting.[2] We split the data into training set (files 1-270 and files 400-931), development set (files 301-325) and test set (files 271-300). The development and test set were used gold-standard segmentation and POS tags in CTB.

We measured the parser quality by the unlabeled attachment score (UAS), e.g., the percentage of tokens (excluding all punctuation tokens) with the correct HEAD. And we also evaluated on complete dependency analysis (CM).

6.1 Splitting Experiments

In this subsection, we conduct the experiments only using the splitting operator. The results are shown in Table 4, where Ord1/Ord2 refers to a first-/second-order parsers (McDonald et al., 2005; McDonald and Pereira, 2006; Carreras, 2007) with baseline features. Ord1f/Ord2f refers to a first-/second-order parsers with baseline+fine-grained features, and the im-

Models	UAS	CM
Ord1	86.57	42.24
Ord1f	86.72 (+0.15)	42.81
Ord2	88.27	46.84
Ord2f	88.51 (+0.24)	47.99

Table 4: Dependency parsing results on the test set only using the splitting operator.

provements by the fine-grained features over the baseline features are shown in parentheses. There are some clear trends in the results.

First, the performance increases with the order of the parser: the first-order model (Ord1) has the lowest performance, adding sibling and grandparent interactions (Ord2) yield better performance. Similar observations regarding the effect of model order have also been made by Carreras (2007) and Koo et al. (2008).

Second, note that the parsers using the fine-grained features outperform the baseline, regardless of model order. Moreover, the benefits of the fine-grained features can improve the performance with the increasing of the model order. For example, increasing the model order from Ord1 to Ord1f results in a relative reduction in error of roughly 1.12%, while introducing fine-grained features from Ord2 to Ord2f yields an additional relative error reduction of roughly 2.05%.

6.2 Merging Experiments

To prevent oversplitting, we merge the subcategories based on the threshold constraint. For parameter f in equation (3), different POS tags (e.g., NN, VV, JJ, ON, $\cdots$) need different values. We do the experiments on the development set to determine the best value among 10, 20, 50, 100, 200, 300, $\cdots$, 1,000 in terms of UAS for each POS tag. The number of the subcategories are shown in Table 2 (in Section 4). Our experiments corresponding to the best parameter values are evaluated on the test set of CTB 5.0.

Table 5 shows the results. The performances can be further increased after using the merging operator. Such a fact validates the effectiveness of merging operator. Overall, for the first-order parser, we find that there is an absolute improvement of 0.52 points (UAS) by adding fine-grained features. For the second-order parser, we get an absolute improvement of 0.61 points (UAS) by including fine-grained features. The improvements of parsing with fine-grained features are mildly significant using the Z-test of Collins et al. (2005).

Models	UAS	CM
Ord1	86.57	42.24
Ord1f	**87.09**	43.39
improvement	+0.52	-
significant level	$p < 0.08$	-
Ord2	88.27	46.84
Ord2f	**88.88**	48.85
improvement	+0.61	-
significant level	$p < 0.05$	-

Table 5: Dependency parsing results on the test set after using the merging operator.

Systems	≤40 words (UAS)	Full (UAS)
Wang et al. (2007)	86.6	-
Yu at al. (2008)	-	87.26
Zhao et al. (2009)	88.9	87.0
Chen et al. (2009)	**92.34**	**89.91**
Ours	90.86	88.88

Table 6: Dependency parsing results on this data set for our second-order model and the previous work.

Systems	UAS
Duan et al. (2007)	84.38
Zhang and Clark (2008)	**86.21**
Huang and Sagae (2010)	85.20
Ours	85.45

Table 7: Comparison of our final results with other best-performing systems on this data set.
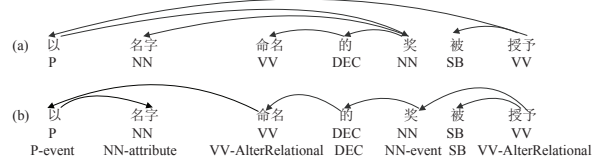

Figure 5: Dependency trees of an example sentence "以(with) 名字(name) 命名(named) 的(of) 奖(prize) 被(by) 授予(award)" as its English translation "··· prize named by ··· name is awarded ···". (a) Dependency tree produced by the baseline model; (b) Dependency tree produced by the proposed approach.

6.3 Comparison with Previous Work

To put our results in perspective, we also compare our second-order system with other best systems: Wang et al. (2007), Yu at al. (2008), Zhao et al. (2009) and Chen et al. (2009), respectively. The results are shown in Table 6, our approach outperforms the first three systems. Chen et al. (2009) reports a very high performance using subtree features from auto-parsed data. In our systems, our do not use such knowledge.

Some researchers conducted experiments on CTB with a different data split: files 1-815 and files 1001-1136 for training, files 816-885 and files 1137-1147 for test, files 886-931 and 1148-1151 for development. The development and test sets were also performed using the gold-standard assigned POS tags. We report the experimental results as well as the performance of previous work on this data set shown in Table 7. Our results are better than most previous work, although Zhang and Clark (2008) achieved an even higher accuracy (86.21) by combining both graph-based and transition-based parsing into a single system for training and decoding. Moreover, their technique is orthogonal to ours, and we suspect that integrating the fine-grained features into the combined parsers might get an even better performance.

6.4 Discussion

Our purpose in this paper is to incorporate the fine-grained features to assist the dependency parsing.

Figure 5 shows an example of dependency trees produced by the baseline parser and our proposed approach.

In Figure 5(a), the baseline parser incorrectly assigned 奖/NN (prize) as the modifier of 以/P (with) and the head of 以/P was also incorrectly recognized as 授予/VV (award). The reason may be that the POS features (**P→NN** and **VV→P**) are too general to model the syntactic dependencies. However, after introducing the fine-grained features **P-event→NN-attribute** and **VV-AlterRelational→P-event**, 名字/NN (name) was selected as modifier of 以/P (with) and the head of 以/P (with) was correctly recognized (Figure 5(b)).

Besides, there exist a large number of neighborhood ambiguities in Chinese dependency parsing, such as "NN NN NN", "JJ NN NN", "AD VV VV", "JJ NN CC NN" and so on they have possible parsing trees as shown in Figure 6. For those ambiguities, our approach can provide the fine-grained features as additional information for the parser. For example, we have the following case in the data set: "外商NN(foreign tradesman)/投资NN(investment) /企业NN(enterprise)/". We can provide additional information about the relations of "外商**NN-human**(foreign tradesman)/ 企业**NN-InstitutePlace**(enterprise)" and "外商**NN-human**(foreign tradesman)/投资**NN-event**(investment)", which can be used to help the parser make the correct decision. Our approach
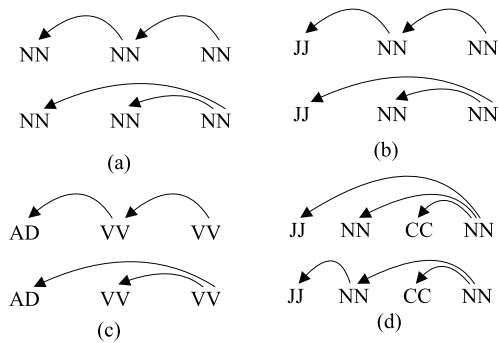
Figure 6: Neighborhood ambiguities in Chinese dependency parsing.

can also help the longer dependencies, such as "JJ NN NN NN" and "NN NN NN NN". For "JJ NN1 CC NN2" ambiguity, we can provide the additional information about the relations of JJ/NN1 and JJ/NN2. In this case, the dependency parser can correctly differentiate the ambiguity.

Our proposed approach is only a preliminary work. Despite the success, there are still some problems which should be extensively discussed in the future work.

(1) In this paper we split the POS tags by using the gold-standard POS tags of CTB. However, in many real application problems, the sentences to be parsed are often from the plain text and the POS tagging is an inevitable phase before dependency parsing. But the split of POS tags will bring great difficulty to the POS tagging phase. Whether the increase the parsing performance will cover the decrease of the POS tagging, this is a very appealing and challenging task in practice. We will leave it for future research.

(2) To deal with the problem of polysemous words, we just use the first definitions in HowNet. A natural avenue for further research would be the development of word sense disambiguation (WSD) technology to solve this problem.

7 Related Work

In this paper, we have focused on developing new representations for POS information. The idea of exploiting different granularities of information for dependency parsing has previously been investigated.

Liu et al. (2007) subdivided verbs according to their grammatical functions and integrated the information of verb subclasses into the dependency parsing model. They regarded the verb subdividing process as a classification task. In contrast, we

split the POS tags based on HowNet hierarchical semantic knowledge and relax the subdivision to be all types of POS tags, which is much simpler than the classification-based method.

Koo et al. (2008) introduced lexical intermediaries at a coarser level than words themselves via a cluster method. Our approach is similar to theirs in that we used the fine-grained feature generation scheme based on HowNet hierarchical semantic knowledge, and the fine-grained features can be viewed as being a kind of "back-off" version of the baseline features. However, we focus on the problem of POS representation instead of lexical representation.

Recently, there are some studies focusing on parsing task using semantic knowledge. Agirre et al. (2008) used word sense information to improve English parsing and PP attachment. Xiong et al. (2005) and Lin et al. (2009) extracted hypernym features from HowNet semantic knowledge and integrated the features into a generative model for Chinese constituent parsing. As with their work, we also use semantic knowledge for parsing. However, our gold is to employ HowNet hierarchical semantic knowledge to generate fine-grained features to dependency parsing, rather than to PCFGs, requiring a substantially different model formulation. Besides, Bansal and Klein (2011) and Zhou et al. (2011) exploited web-scale semantic information for parsing.

8 Conclusion

In this paper, we focus on the problem of grammar representation, introducing fine-grained features by splitting various POS tags to different degrees using HowNet hierarchical semantic knowledge. To prevent the oversplitting, we adopt a threshold-constrained bottom-up strategy to merge the derived subcategories. The results show that, with the fine-grained features, we can improve the dependency parsing accuracies by 0.52% (absolute) for the unlabeled first-order parser, and in the case of second-order parser, we can improve the dependency parsing accuracies by 0.61% (absolute).

Acknowledgments

References

E. Agirre, T. Baldwin, and D. Martinez. 2008. Improving parsing and PP-attachment performance with sense information. In *Proceedings of ACL-08: HLT*, pages 317-325.

M. Bansal and D. Klein. 2011. Web-Scale Features for Full-Scale Parsing. In *Proceedings of ACL-HLT*, pages 693-702.

X. Carreras. 2007. Experiments with a higher-order projective dependency parser. In *Proceedings of EMNLP-CoNLL*, pages 957-961.

W. Chen, D. Kawahara, K. Uchimoto, and Torisawa. 2009. Improving dependency parsing with subtrees from auto-parsed data. In *Proceedings of EMNLP*, pages 570-579.

M. Collins, A. Globerson, T. Koo, X. Carreras, and P. L. Bartlett. 2008. Exponentiated gradient algorithm for conditional random fields and max-margin markov networks. *Journal of Machine Learning Research*, pages 1775–1822.

M. Collins, P. Koehn, and I. Kucerova. 2005. Clause restructuring for statistical machine translation. In *Proceedings of ACL*, pages 531-540.

Z. Dong and Q. Dong. 2000. HowNet Chinese-English conceptual database. *Technical report online software database, released at ACL*, http://www.keenage.com.

X. Duan, J. Zhao, and B. Xu. 2007. Probabilistic Models for action-based Chinese dependency parsing. In *Proceedings of ECML/PKDD*.

K. W. Gan and P. W. Wong. 2000. Annotating information structures in Chinese texts using HowNet. In *Proceedings of ACL*.

J. Hall, J. Nivre, and J. Nilsson. 2006. Discriminative classifier for deterministic dependency parsing. In *Proceedings of ACL*, pages 316-323.

L. Huang and K. Sagae. 2010. Dynamic Programming for Linear-Time Incremental Parsing. In *Proceedings of ACL*, pages 1077-1086.

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

T. Koo, X. Carreras, and M. Collins. 2008. Simple semi-supervised dependency parsing. In *Proceedings of ACL-08: HLT*, pages 595-603.

X. Lin, Y. Fan, M. Zhang, X. Wu, H. Chi. 2009. Refining grammars for parsing with hierarchical semantic knowledge. In *Proceedings of EMNLP*, pages 1298-1307.

T. Liu, J. Ma, H. Zhang, and S. Li. 2007. Subdivided verbs to improve syntactic parsing. *Journal of electronics*, 24(3).

T. Matsuzaki, Y. Miyao, and J. Tsujii. 2005. Probabilistic CFG with latent annotation. In *Proceedings of ACL*, pages 75-82.

A. F. T. Martins, D. Das, N. A. Smith, and E. P. Xing. 2008. Stacking dependency parsers. In *Proceedings of EMNLP*, pages 157-166.

R. McDonald and F. Pereira. 2006. Online learning of approximate dependency parsing algorithms. In *Proceedings of EACL*, pages 81-88.

R. McDonald, K. Crammer, and F. Pereira. 2005. Online large-margin training of dependency parsers. In *Proceedings of ACL*, pages 91-98.

J. Mei, Y. Zhu, Y. Gao, and H. Yin. 1983. TongYiCiLin. *Shanghai Lexicographical Publishing House..*

J. Nivre and R. McDonld. 2008. Integrating graph-based and transition-based dependency parsing. In *Proceedings of ACL-08: HLT*, pages 950-958.

S. Petrov and D. Klein. 2006. Learning accurate, compact, and interpretable tree annotation. In *Proceedings of COLING-ACL*, pages 433-440.

H. Schuetze. 1998. Automatic word sense discrimination. *Computational Linguistics*, 24(1): 97-124.

Q. I. Wang, D. Lin, and D. Schuurmans. 2007. Simple training of dependency parsers via structured boosting. In *Proceedings of IJCAI*, pages 1756-1762.

D. Xiong, S. Li, Q. Liu, S. Lin, and Y. Qian. 2005. Parsing the Penn Chinese Treebank with semantic knowledge. In *Proceedings of IJCNLP*.

N. Xue, F. Xia, F.-D. Chiou, and M. Palmer. 2005. The Penn Chinese Treebank: Phrase structure annotation of a large corpus. *Natural Language Engineering*, 10(4):1-30.

Yamada and Matsumoto. 2003. Statistical dependency analysis with support vector machines. In *Proceedings of IWPT*, pages 195-206.

K. Yu, D. Kawahara, and S. Kurohashi. 2008. Chinese dependency parsing with large scale automatically constructed case structures. In *Proceedings of COLING*, pages 1049-1056.

Y. Zhang and S. Clark. 2008. A tale of two parsers: investigating and combining graph-based and transition-based dependency parsing using beam-search. In *Proceedings of EMNLP*, pages 562-571.

H. Zhao, Y. Song, C. Kit and G. Zhou. 2009. Cross language dependency parsing using a bilingual lexicon. In *Proceedings of ACL*, pages 55-63.

G. Zhou, J. Zhao, K. Liu and L. Cai. 2011. Exploiting web-derived selectional preference to imporve statistical dependency parsing. In *Proceedings of ACL-HLT*, pages 1556-1665.