# Improving Chinese Dependency Parsing with Lexical Semantic Features

Lvexing Zheng[1(✉)], Houfeng Wang[1], and Xueqiang Lv[2]

[1] Key Laboratory of Computational Linguistics, Peking University, Ministry of Education, Beijing, China
zhengulxin@163.com
[2] Beijing Key Laboratory of Internet Culture and Digital Dissemination Research, Beijing, China

**Abstract.** Lexical semantic information plays an important role in supervised dependency parsing. In this paper, we add lexical semantic features to the feature set of a parser, obtaining improvements on the Penn Chinese Treebank. We extract semantic categories of words from HowNet, and use them as semantic information of words. Moreover, we investigate the method to compute semantic similarity between Chinese compound words, and obtain semantic information of words which did not record in HowNet. Our experiments show that unlabeled attachment scores can increase by 1.29%.

**Keywords:** Lexical semantic features · HowNet · Semantic similarity · Chinese compound words

## 1 Introduction

Due to the data sparseness problem, the lexical information from a Treebank for a lexicalized parser could be insufficient, and the parser is mainly based on part-of speech (POS) information. However, the low granularity of POS limits the performance of the parser. Using lexical semantic information to solve data sparseness problem has become an interesting research means with the emergence of lexical semantic resources such as WordNet (Fellbaum,1998), HowNet (Dong et al., 2003), CiLin (Che et al., 2010). Several research works have tried to test the intuition that lexical semantic information should help parsing, as a word can be generalized to semantic classes in a lexicalized parser (i.e. Bengoetxea et al., 2014; Agirre et al., 2011; Xiong et al., 2005).

In general, a word would be more syntactically similar to the other if they were more semantically similar to each other. For example, we may observe in our training data that "*大学生*" (undergraduate student) often occurs as the subject of word "*阅读*" (read). We assume that the word pair "*小学生*" (elementary school student) and "*阅读*" (read) do not appear in the training data, but "*小学生*" is semantically similar to word


J. Li et al. (Eds.): NLPCC 2015, LNAI 9362, pp. 36–46, 2015.
DOI: 10.1007/978-3-319-25207-0_4

"*大学生*". If we incorporate their semantic information into the parser, it can help the parser to predict that they have the same attachment preferences in dependency tree.

In this paper, we present a simple and effective method for incorporating lexical semantic information into the parser. Instead of substituting words with their semantic classes (Agirre et al., 2011), we add lexical semantic information to the feature set of the parser. When lexical information cannot help recognize relation of two words, our approach makes it back off to semantic information. For example, word bi-gram information <word$_i$, word$_j$> can back off to <sense$_i$, word$_j$> , <word$_i$, sense$_j$> and < sense$_i$, sense$_j$>.

Moreover, many Chinese words (called **UNword** for short) hardly recorded in a semantic dictionary, so we cannot directly obtain their semantic information. We investigate the construction of Chinese compound words, and propose an approach to computing semantic similarity between the **UNword** and the word recorded in the dictionary. Then, we obtain the most similar semantic information of the **UNword**, and incorporate it into the parser.

As our baseline parser, we use MSTParser (McDonald et al., 2005; McDonald and Pereira, 2006). We extract semantic categories at various granularity levels in HowNet. We present a set of experiments in dependency parsing of the Penn Chinese Treebank 5.1 (Xue et al., 2000). The results show that a significant improvement in performance is achieved when lexical semantic information is incorporated into the parser.

## 2 Framework

In this section, we extract semantic categories in HowNet (subsection 2.1). In subsection 2.2, we describe the semantic feature templates used by the parser. In subsection 2.3, we present an approach to computing semantic similarity between the **UNword** and the word recorded in HowNet.

### 2.1 Extracting Semantic Categories

We use the HowNet3.0 dictionary to extract semantic categories, which covers 66,181 words defined by sememes.

HowNet (HN): Each sememe defined in the HowNet is regarded as a semantic category. The typical relation between different categories is hypernym-hyponym. Through the hypernym ladders, we can extract semantic categories at various granularity levels HN1, HN2, and HN3. HN1 stands for semantic categories at the first level, so HN2 and HN3 is the second and third level. Since a word may have many senses in HowNet, we obtain semantic category of the word by two ways. The first one is to choose the first sense of the word in HowHet by following the work (Xiong et al., 2005, Agirre et al, 2011), and call them HN1-1st, HN2-1st, and HN3-1st. The second method is to automatically rank senses (Agirre et al, 2011), but the experiments show that improvement is worse than the first method. Instead of that, we extract all senses of a word to form a new category, and call them HN1-all, HN2-all, and HN3-all. Table 1 shows information about words and semantic categories.

**Table 1.** Size and coverage of words and semantic categories.

| | Data | HN1-1st | HN2-1st | HN3-1st | HN1-all | HN2-all | HN3-all |
|---|---|---|---|---|---|---|---|
| Words in train | 437990 | 372974 | | | | | |
| Words in test | 50319 | 43240 | | | | | |
| Words in both | 47172 | 42218 | | | | | |
| Categories in train | | 910 | 3985 | 5838 | 3766 | 7383 | 9043 |
| Categories in test | | 714 | 1921 | 2669 | 2077 | 3398 | 3983 |
| Categories in both | | 714 | 1817 | 2498 | 2018 | 3201 | 3714 |

In Table 1, words of train data have a great coverage of test data, so atomic features (McDonald et al., 2005) such as word unigrams are less likely to be sparse, but the higher-order features are poor for parsing. For example, the dependency coverage of word bi-grams feature (<word$_i$, word$_j$>) is only 50.09%. As can be seen, HowNet has a great coverage of train and test data, and lexical semantic information can help recognize relation of two words. In our experiments, we choose HN1/2/3-st and HN1/2/3-all as our external lexical semantic sources.

## 2.2 Semantic Feature Templates

We extend the baseline 1-order and second-order features in (McDonald et al., 2005; McDonald and Pereira, 2006) by introducing lexical semantic information into the parser. The feature templates are shown in table 2.

In Table 2, we incorporate lexical semantic information into the parser by using bi-gram and surrounding features which almost follow the bastline feature set. For example, we change the baseline features (<p-word, c-word>) into features <p-word, c-sense>, <p-sense, c-word> and <p-sense, c- sense>. Then <p-word, c-word> can back off to <p-word, c-sense>, <p-sense, c-word> and <p-sense, c-sense>, when it do not exist in train data.

In Table 3, Following the work (McDonald and Pereira, 2006), we also include conjunctions between these lexical semantic features and the direction and distance from sibling *j* to sibling *k*.

It is notable that if one word is not recorded in HowNet, we substitute the sense of this word with its POS tag. We try to substitute all of this word with “no-sense”, but the dependency accuracy of the parser cannot increase. We also try to incorporate single-sense features into the parser, but it works worse than baseline. The main reason may be that the features induce dependency ambiguities of single-word.

**Table 2.** Lexical semantic features used by MSTParser-1-order. p-word: word of parent node in dependency tree. c-word: word of child node. p-pos: POS of parent node. c-pos: POS of child node. p-sense: the semantic class of parent node. c-sense: the semantic class of child node. p-sense+1: sense to the right of parent in sentence. p-sense-1: sense to the left of parent. c-sense+1: sense to the right of child. c-sense-1: sense to the left of child.

| | |
|---|---|
| Bi-gram Semantic Features | *<p-sense/p-pos, c-sense/c-pos>;*<br>*<p-word/p-sense, c-word/c-sense>;*<br>*<p-sense, c-word>; <p-sense, c-sense>;*<br>*<p-sense/p-pos, c-pos>; <p-pos, c-sense/c-pos>;*<br>*<p-word, c-word/c-sense>; <p-word/p-sense, c-word>*<br>*<p-sense, c-pos>; <p-pos, c-sense>; <p-word, c-sense>;* |
| Surrounding Semantic Features | *<p-sense, p-sense+1, c-sense-1, c-sense>;*<br>*<p-sense-1, p-sense, c-sense-1, c-sense>;*<br>*<p-sense, p-sense+1, c-sense, c-sense+1>;*<br>*<p-sense-1, p-sense, c-sense, c-sense+1>;* |

**Table 3.** Lexical semantic features used by MSTParser-2-order. $x_i$-sense: the sense of the $i^{th}$ word in sentence. $x_k$: the sibling node *k* of $x_i$. $x_j$: the sibling node *j* of $x_i$.

| | |
|---|---|
| second-order Semantic Features | <$x_k$-*sense*, $x_j$-*sense*>; <$x_k$-*sense*, $x_j$-*pos*>;<br><$x_k$-*word*, $x_j$-*sense*>; <$x_k$-*sense*, $x_j$-*word*>;<br><$x_k$-*pos*, $x_j$-*sense*>; <$x_i$-*sense*, $x_k$-*sense*, $x_j$-*sense*>; |

### 2.3 Semantic Similarity

The next problem is to incorporate **UNword** (not recorded in HowNet) semantic information into dependency parsing. Many researchers measure semantic similarity between two words with a large number of contexts in which the two words occurring, what is called “distributional similarity” (McCarthy et al., 2004). However, due to the data sparseness problem, there is obvious limitation for this measure (Agirre et al, 2011).

Instead, we investigate the construction of Chinese words, and try to compute semantic similarity between **UNword** and word recorded in HowNet. Then, we can obtain the most similar semantic class of **UNword**.

Compounding is the most important method to form new words in Chinese word-formation system (Yan, 2007). The compound words follow the endocentric principles, and a basic framework for the semantic construction of compound words is studied (Yan, 2007), that is *one supplementary semantic component (affix) + one head semantic component (root) = one word meaning*. For the sake of convenience, we call them “affix” and “root”. The root could be consistent with word meaning, and two words containing the same root could have similar meanings.

The problem we face is to automatically split a compound word into the affix and the root. There are many types of compound words in Chinese, but modification-center is the most common structure and its root is at end of the word (Yan, 2007).

For convenience, we regard the shared part of two words as a root, and look for it at the end of two words. Thus, after the root of a word is found, the remaining part becomes its affix. So, we denote a word $w_i$ as $(a_{w_i}, r_{w_i})$, where $a_{w_i}$ stands for the affix of the word $w_i$, and $r_{w_i}$ stands for the root. For a pair of words, we present them as follows:

*(小学生(elementary school student)/$w_1$); (大学生(undergraduate school student)/$w_2$) ->*

*(小(small)/$a_{w_1}$，学生(student)/$r_{w_1}$); (大(big)/$a_{w_2}$，学生(student)/$r_{w_2}$);*

*(学生(student)/$w_3$); (大学生(undergraduate school student)/$w_2$)->*

*(Ø/$a_{w_3}$，学生(student)/$r_{w_3}$); (大(big)/$a_{w_2}$，学生(student)/$r_{w_2}$);*

Taking the principle of compound word segmentation, we need to consider two situations:

1. Both the affixes of two word are not ∅, such as "*小学生 (*elementary school student*), 大学生 (*undergraduate school student*)*". Intuitively, the word pair "*小学生, 大学生*" are likely to have the same attachment preferences. There are two reasons for it. One reason is that the meaning of the root "*学生*" (student) with any prefix is very similar to the word "*学生*" (student), and belonging to the semantic class *"human_人|*study_学"* in HowNet. So, we reckon that the more words in a semantic class share a root, the word with this root is the more likely to belong to this semantic class.
   The other reason is that the affixes "*小 (small)*" and "*大 (big)*" both show the age of the root "*学生*" (student), and they have very semantic similar that the majority of pairs of compound words are synonymous, which have the same root and the affixes *"小"* and *"大"*. So, we can use synonyms relation in a semantic dictionary to measure the similarity between them. We predict that the more pairs of compound words with same root and affixes ($a_{w_i}$, $a_{w_j}$) are synonymous in a semantic dictionary, thus affixes $a_{w_i}$ and $a_{w_j}$ are the more semantic similar.
2. One word is the root of another, such as "*学生 (student), 大学生 (undergraduate school student)*". Unlike (1), the affix of one word must to be ∅. But, if we regard ∅ as a special affix, the situation 2 is the similar to 1. A pair of compound words with affixes $a_{w_i}$ and ∅ is the more semantic similar, if the more pairs of compound words with same root and affixes ($a_{w_i}$, ∅) are synonymous.

So, to compute semantic similarity between two compound words, there are two factors need to be considered: the effect of the root on the whole similarity in a semantic class and the similarity between two affixes of a pair of words. In this paper, we integrate two factors to similarity evaluation which is shown in equation (1).

$$\begin{aligned} Sim(uw_k, w_{ij}) &= Sim((a_{uw_k}, r_{uw_k}), (a_{w_{ij}}, r_{w_{ij}})) \\ &= \lambda(r_{w_{ij}}) * AffixSim(a_{uw_k}, a_{w_{ij}}) \end{aligned}$$

$$where: r_{uw_k} = r_{w_{ij}} \; and \; r_{w_{ij}} \neq \emptyset \tag{1}$$

Here $Sim(uw_k, w_{ij})$ represents semantic similarity between **UNword** $k$ and word $j$ with the semantic class $i$. $a_{uw_k}$ stands for the affix of the word $uw_k$, and $r_{uw_k}$ stands for the root. $\lambda(r_{w_{ij}})$ represents the degree of effect of the root $r_{w_{ij}}$ in the semantic class $i$. $AffixSim(a_{uw_k}, a_{w_{ij}})$ represents the semantic similarity between the affixes $a_{uw_k}$ and $a_{w_{ij}}$.

The $\lambda(r_{w_{ij}})$ is estimated in equation (2).

$$\lambda\left(r_{w_{ij}}\right) = \frac{C^2_{Sum_{class_i}\left(r_{w_{ij}}\right)}}{\sum_x C^2_{Sum_{class_x}\left(r_{w_{ij}}\right)}} \tag{2}$$

Here $Sum_{class_i}(r_{w_{ij}})$ indicates the number of words containing the *root* $r_{w_i}$ in the semantic class i. $C^2_{Sum_{class_i}(r_{w_{ij}})}$ means the number of word pairs which root are $r_{w_{ij}}$ in the semantic class $i$. $\sum_x C^2_{Sum_{class_x}(r_{w_{ij}})}$ indicates the number of all pairs of synonyms which roots are $r_{w_{ij}}$. So, if $\lambda(r_{w_{ij}})$ is closer to 1, the fewer effects have on the whole similarity, and the word $uw_k$ is more likely to belong to the semantic class $i$.

The $AffixSim(a_{uw_k}, a_{w_{ij}})$ is estimated in equation (3).

$$AffixSim(a_{uw_k}, a_{w_{ij}}) = \frac{\sum_{r_{w_x} \neq \emptyset} count((a_{uw_k}, r_{w_x}) = (a_{w_{ij}}, r_{w_x}))}{\sum_{r_{w_x} \neq \emptyset} count((a_{uw_k}, r_{w_x}) = (a_{w_{ij}}, r_{w_x})) + \sum_{r_{w_x} \neq \emptyset} count((a_{uw_k}, r_{w_x}) \neq (a_{w_{ij}}, r_{w_x}))}$$

$$where\ words\ (a_{uw_k}, r_{w_x})\ and\ (a_{w_{ij}}, r_{w_x}) \in HowNet \tag{3}$$

Here $(a_{uw_k}, r_{w_x})$ means the word is split into the affix $a_{uw_k}$ and the root $r_{w_x}$. $count((a_{uw_k}, r_{w_x}) = (a_{w_{ij}}, r_{w_x}))$ indicates the number of the pairs in which the words $(a_{uw_k}, r_{w_x})$ and $(a_{w_{ij}}, r_{w_x})$ are synonymous. $count((a_{uw_k}, r_{w_x}) \neq (a_{w_{ij}}, r_{w_x}))$ means the number of the pairs in which the words are not synonymous.

Therefore, we obtain the most similar semantic information of **UNword** $k$ by equation (4), and respectively extend our external lexical semantic sources (HN1/2/3-st, HN1/2/3-all) with the semantic class of **UNword** k. Here $SemClass(w_{ij})$ means the semantic class $i$ of the word $w_{ij}$, $POS(uw_k)$ means part-of speech of word $uw_k$ recorded in CTB, and $POS(w_{ij})$ is recorded in HowNet. NR means proper nouns, and CD means cardinal numbers. In many cases, NR which includes personal names and transliterated words cannot fit the semantic construction of compound words. Recognizing the dependency relation of CD is already used by simple rules in the baseline parser.

$$SemClass(uw_k) = SemClass(argmax_{(w_{ij})} Sim(uw_k, w_{ij}))$$

$$where\ \ Sim(uw_k,\ w_{ij}) \geq \theta, POS(uw_k) = POS(w_{ij}),$$
$$and\ POS(uw_k) \neq "NR" and\ "CD";\ otherwise\ \ SemClass(uw_k) = \emptyset \qquad (4)$$

# 3 Experiment

## 3.1 Data set

We use Penn Chinese Treebank 5.1 (Xue et al., 2000) as data set in the experiments. The Penn Chinese Treebank 5.1 (CTB) is phrase structure Treebank, and we use the toolkit Penn2Malt (Johansson and Nugues, 2007) to transfer them to dependency treebank. To balance each resource in train set, development set and test set, we follow Duan's work (Duan, 2007), and split the data set as in table 4.

**Table 4.** The division of CTB data set

| | CTB files | Number of sentences |
|---|---|---|
| Train set | 001-815, 1001-1136 | 16,091 |
| Development set | 886-931, 1148-1151 | 803 |
| Test set | 816-885, 1137-1147 | 1,910 |

## 3.2 Experimental Results

We use MSTParser (McDonald et al., 2005; McDonald et al., 2006) as our basic parser. It represents global, exhaustive graph-based parsing that finds the highest scoring directed spanning tree in a graph. The parser can be trained using first or second order models, and we use default options.

We use both labeled attachment score (LAS) and unlabeled attachment score (UAS) to evaluate the all experiments, and punctuation is included in all evaluation metrics. We consider three options:

1. We substitute words with their semantic classes in the process of training and testing the parser using our external lexical semantic sources (HN1/2/3-st, HN1/2/3-all). The results for parsing are given in Table 5.

**Table 5.** Parsing results by substituting words with their semantic classes.

| | LAS | UAS |
|---|---|---|
| MSTParser-1order(baseline) | 78.86 | 80.95 |
| HN1-st | 77.71 | 79.96 |
| HN2-st | 78.23 | 80.42 |
| HN3-st | 78.44 | 80.65 |
| HN1-all | 78.68 | 80.89 |
| HN2-all | 78.69 | 80.82 |
| HN3-all | 78.69 | 80.87 |

In Table 5, by substituting words with their semantic classes, the results are not superior to the baseline. Though the parser knows that words are synonymous, it can help the parser to predict that they have the same attachment preferences in dependency tree. However, there are many words belonging to the same semantic class in a semantic dictionary, but they have not the same attachment preferences or have some different attachment preferences in dependency parsing. For example, the words "*执法*" (enforce the law) and "*执法必严*" (strictly enforce the law) are in the same semantic class "*conduct_实施*" in HowNet. But the word "*执法必严*" (strictly enforce the law) has different attachment preference compared to "*执法*" (enforce the law), and do not have an adverb. So, this method causes lexical information loss and cannot improve parsing performance.

2. We extend the baseline features with the semantic feature template (Table 2), then train and test MSTParser-1order by using our sources (HN1/2/3-st, HN1/2/3-all). Experimental results show that the best performance of MSTParser-1order model is obtained with HN2-all. Due to the fact that the MSTParser-2order model extends the MSTParser-1order with the second-order features, we only incorporate HN2-all into 2order model with the semantic feature templates (Table 2 and 3). Moreover, based on section 3.3, we can obtain the most similar semantic information of **UNwords** by equation (4), and extend our external lexical semantic source HN2-all, and call it HN2-all-E for short. Then, the MSTParser-1order and 2order model need to retrain by using HN2-all-E. In equation 4, we set threshold $\theta = 0.9$. The results of semantic class of **UNwords** are shown in Table 6 and the experimental results are given in Table 7.

**Table 6.** Examples of semantic class of UNwords using HN2-all

| **UNword** | The most similar Word | Semantic class in HN2-all | Similarity |
|---|---|---|---|
| 西南郊 (southwestern suburbs) | 西郊 (western suburbs) | part_部件 /%place_地方 | 1.0 |
| 系列赛 (series of competitions) | 公开赛 (open championship) | fact_事情 /compete_比赛 | 0.9848 |
| 乍看 (glance) | 观看 (watch) | look_看 | 0.6000 |
| 乐于 (be happy to) | 位于 (locate) | situated_处于 | 0.1429 |

Table 7 shows the performance of the baseline that is extended with the semantic feature templates. We can see that in all cases our external lexical semantic sources improve over the baseline. Using HN2-all, UAS of MSTParser-1order and 2order respectively increase by 1.21%, 1.16%. The main purpose of adding the semantic feature into the parser is that if lexical dependency information of two words is sparse, and it can back off to lexical semantic dependency information, and the experimental results prove its validity. Compare with the method of substituting words with their semantic classes, this method do not loss high-frequency word and word-pair dependency information in general, and it can help recognize relation of lower-frequency word and word-pair. For example, if a high-frequency verb always does not have an object (OBJ), the parser can also consider this situation.

**Table 7.** Parsing results by extending the baseline features with the semantic feature templates.

| | $\theta$ | LAS | UAS |
|---|---|---|---|
| MSTParser-1order(baseline) | – | 78.86 | 80.95 |
| 1order + HN1-st | – | 79.79 | 81.85 |
| 1order + HN2-st | – | 79.95 | 82.05 |
| 1order + HN3-st | – | 79.98 | 82.00 |
| 1order + HN1-all | – | 80.06 | 82.09 |
| **1order + HN2-all** | – | 80.09 | **82.16(+1.21)** |
| 1order + HN3-all | – | 79.92 | 82.00 |
| 1order + HN2-all-E | 0.9 | 80.18 | 82.28 |
| MSTParser-2order(baseline) | – | 80.85 | 83.04 |
| 2order + HN2-all | – | 82.08 | **84.20(+1.16)** |
| 2order + HN2-all-E | 0.9 | **82.19(+1.34)** | **84.33(+1.29)** |

In Table 6 and 7, the number of **UNwords** is 11,071 in train data and 1,467 in test data, and our approach (section 3.3) adds 1,199 (10.83%) words with semantic class in train data, and 192 (13.09%) words in test data. Compared with HN2-all, table 7 shows UAS of MSTParser-2order increases by 0.13% using HN2-all-E. So, it proves our approach can effectively obtain the semantic information of **UNwords**. Due to the fact that the lexical semantic information is auxiliary information for parsing, the improvement is not significant.

3. We include the results of ZPar-dep (Zhang and Nivre,2011) and neural network model of Chen and D.Manning (2014) for comparison, and the experimental results are given in Table 8. As we can see, compared with ZPar-dep (Zhang and Nivre, 2011) and neural network model (Chen and D.Manning, 2014), our approach gets the best UAS.

**Table 8.** Parsing results with HN2-all-E and comparion with high performance models

| | LAS | UAS | LAS(excluding punctuations) | UAS(excluding punctuations) |
|---|---|---|---|---|
| 2order + HN2-all-E | 82.19 | **84.33** | 84.27 | **86.36** |
| ZPar-dep (Zhang and Nivre, 2011) | – | – | 84.40 | 86.00 |
| Neural Network Model (Chen and D.Manning, 2014) | 82.40 | 83.90 | – | – |

## 4 Related Work

Agirre et al. (2008) used semantic classes to help parsing. Later, they extended the test and successfully introduced WordNet classes in a dependency parser (Agirre et al., 2011). MacKinlay et al. (2012) investigated the addition of semantic annotations in the form of word sense hypernyms, in HPSG parse ranking.

Ciaramita and Attardi (2007) showed that adding semantic features extracted by a named entity tagger (such as PERSON or MONEY) improved the accuracy of a dependency parser. Candito and Seddah (2010) studied statistical parsing of French, where terminal forms were replaced by more general symbols, particularly clusters of words obtained through unsupervised clustering. The results showed that word clusters had a positive effect.

Apart from these, there have been other attempts to solve the data sparseness problem, Koo et al. (2008) and Suzuki et al. (2009) presented a semi-supervised method for training dependency parsers, using word clusters derived from a large unannotated corpus as features.

## 5 Conclusion

In this paper, we present a simple and effective method for incorporating lexical semantic information into the parser. We can relieve sparse data problem by extending with the semantic feature sets, and obtain the most similar semantic information of words which are not recorded in the lexical semantic resource. Experiments on CTB dataset show our approach achieves significant improvement. Our approach is only a preliminary work and has much future work to do. The considered future work includes incorporating word sense disambiguation method and deep research on basic framework for the semantic structure of Chinese words.

**Acknowledgement.** Our work is supported by National High Technology Research and Development Program of China (863 Program) (No. 2015AA015402), National Natural Science Foundation of China (No.61370117 & No.61333018), Major National Social Science Fund of China (No.12&ZD227) and the Opening Project of Beijing Key Laboratory of Internet Culture and Digital Dissemination Research (ICDD201501).

## References

1. Fellbaum, C. (ed.): WordNet: An Electronic Lexical Database. MIT press, Cambridge (1998)
2. Dong, Z., Dong, Q.: Hownet [EB/OL]. http://www.keenage.com, 2003201211
3. Che, W., Li, Z., Liu, T.: LTP: a Chinese language technology platform. In: Proceedings of the Coling 2010: Demonstrations, Beijing, China, 2010.08, pp. 13–16 (2010)
4. Bengoetxea, K., Agirre, E., Nivre, J., Zhang, Y., Gojenol, K.: On WordNet semantic classes and dependency parsing. In: Proceedings of ACL 2014, Baltimore, Maryland, pp. 649–655 (2014)

5. Agirre, E., Bengoetxea, K., Gojenola, K., Nivre, J.: Improving dependency parsing with semantic classes. In: Proceedings of ACL 2011, Poorland, Oregon, pp. 699–703 (2011)
6. Xiong, D., Li, S., Liu, Q., Lin, S.-X., Qian, Y.: Parsing the Penn Chinese treebank with semantic knowledge. In: Dale, R., Wong, K.-F., Su, J., Kwong, O.Y. (eds.) IJCNLP 2005. LNCS (LNAI), vol. 3651, pp. 70–81. Springer, Heidelberg (2005)
7. McDonald, R., Crammer, K., Pereira, F.: Online large-margin training of dependency parsers. In: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, pp. 91–98. Association for Computational Linguistics (2005)
8. McDonald, R., Pereira, F.: Online learning of approximate dependency parsing algorithms. In: EACL. Citeseer (2006)
9. Xue, N., Xia, F., Huang, S., Kroch, A.: The bracketing guidelines for the perm Chinese treebank (3.0) (2000)
10. McCarthy, D., Koeling, R., Weeds, J., Carroll, J.: Finding predominant senses in untagged text. In: Proc. of the 42nd Annual Meeting of the ACL, Barcelona, Spain, pp. 280–287 (2004)
11. Yan, H.: The Study on the Semantic structure of Compound Words in Contemporary Chinese. Capital Normal University, Beijin (2007)
12. Johansson, R., Nugues, P.: Extended constituent-to-dependency conversion for English. In: Proc.of 16th Nordic Conference on Computational Linguistics (NODALIDA), pp. 105–112 (2007)
13. Duan, X., Zhao, J., Xu, B.: Probabilistic models for action-based Chinese dependency parsing. In: Kok, J.N., Koronacki, J., Lopez de Mantaras, R., Matwin, S., Mladenič, D., Skowron, A. (eds.) ECML 2007. LNCS (LNAI), vol. 4701, pp. 559–566. Springer, Heidelberg (2007)
14. Zhang, Y., Nivre, J.: Transition-based dependency parsing with rich non-local features. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Langauge Technologies, pp. 188–193. Association for Computational Linguistics, Portland (2011)
15. Chen, D., Manning, C.: A fast and accurate dependency parser using neural networks. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 740–750. Association for Computational Linguistics, Doha, Qatar, October 2014
16. Agirre, E., Baldwin, T., Martinez, D.: Improving parsing and PP attachment performance with sense information. In: Proceedings of ACL08: HIT, Columbus, Ohio, pp. 317–325 (2008)
17. MacKinlay, A., Dridan, R., McCarthy, D., Baldwin, T.: The effects of semantic annotations on precision parse ranking. In: First Joint Conference on Lexical and Computational Semantics (*SEM), Montreal, Canada, pp. 228–236 (2012)
18. Ciaramita, M., Attardi, G.: Dependency parsing with second-order feature maps and annotated semantic information. In: Proceedings of the 10th International Conference on Parsing Technology (2007)
19. Candito, M., Seddah, D.: Parsing word clusters. In: Proceedings of the NAACLHLT 2010 First Workshop on Statistical Parsing of Morphologically-Rich Languages, LosAngeles, CA, USA, pp. 76–84 (2010)
20. Koo, T., Cameras, X., Collins, M.: Simple semi-supervised dependency parsing. In: Proceedings of ACL-08: HLT, Columbus, Ohio, pp. 595–603 (2008)
21. Suzuki, J., Isozaki, H., Cameras, X., Collins, M.: An empirical study of semi-supervised structured conditional models for dependency parsing. In: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, Singapore, pp. 551–560 (2009)