Integrating Structural Context with Local Context for Disambiguating Word Senses

Qianlong Du † Chengqing Zong † Keh-Yih Su‡

†National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Science, Beijing, China
†University of Chinese Academy of Sciences, Beijing, China
‡Institute of Information Science, Academia Sinica, Taiwan
†{qianlong.du, cqzong}@nlpr.ia.ac.cn
‡kysu@iis.sinica.edu.tw

Abstract. A novel word sense disambiguation (WSD) discriminative model is proposed in this paper to handle *long distance sense dependency* and *multi-reference lexicon dependency* (i.e., the sense of a lexicon might depend on several other non-local lexicons under the same subtree) within the sentence. Many WSD systems only adopt local context to independently decide the sense of each lexicon in a sentence. However, the sense of a target word actually also depends on those structure related sense/lexicons that might be far away from it. Therefore, we propose a supervised approach which integrates *structural context* (for long distance sense dependency and multi-reference lexicon dependency) with the local context (for local dependency) to handle the problems mentioned above. As the result, the sense of each word is decided not only based on the local lexicons, but also based on various reference sense/lexicons (might be *non-local*) specified by all its associated syntactic subtrees. Experimental results show that the proposed approach significantly outperforms other state-of-art WSD systems.

1 Introduction

Word sense disambiguation (WSD) is a process of determining the appropriate sense of a word in the given context. It is a fundamental task in natural language processing. Usually, we regard word sense disambiguation as an intermediate step, which could help high-level applications in NLP, such as machine translation (Carpuat and Wu, 2005; Carpuat and Wu, 2007; Chan et al., 2007), information retrieval (Stokoe et al., 2003) and content analysis (Berendt and Navigli, 2006). With the help of WSD, it is expected to get a higher performance in these applications.

Among those proposed WSD systems, supervised approaches have achieved the best performance (Tratz et al., 2007; Hatori et al., 2009; Zhong and Ng, 2010). And many of them simply extract some lexicon related features from the local context around the target word, and then independently train a classifier on those features for each word (Zhong and Ng, 2010). Therefore, the correlation between the senses of various words and the long distance dependency specified by the syntactic relation are not considered by them.

adfa, p. 1, 2011.

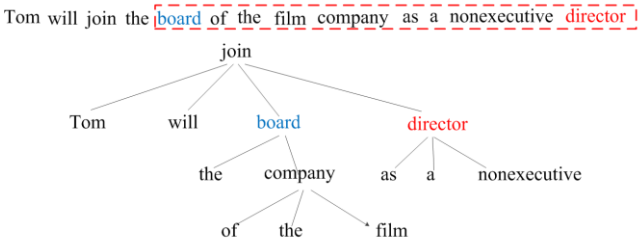

Fig. 1. sample sentence and its dependency tree structure

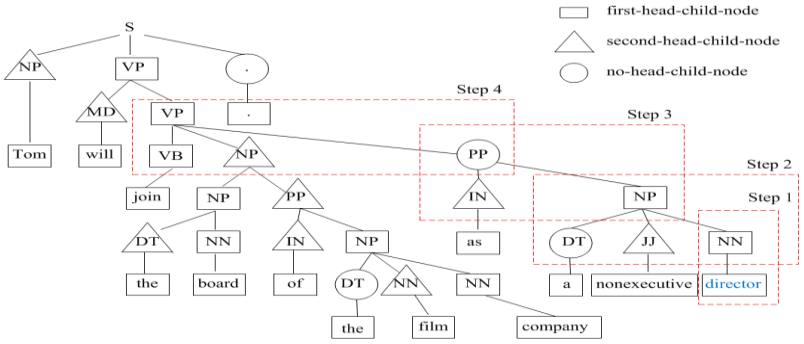

Fig.2. The corresponding phrase structure tree

However, the senses of words do have influence on each other, and the non-local context also affects the sense selection. Furthermore, the dependency should be considered under the syntactic relation between them, as pointed out by Erk and Padó (2008). Also, the long distance dependency should not be covered by simply adopting a large context window, as it would involve a lot of irrelevant words and thus introduce considerable noise. What we really want is to only utilize those closely related non-local words specified by the syntactic structure, not those irrelevant words that are far away from the target word.

Take the following sentence as an example: "*Tom will join the board of the film company as a nonexecutive director*". The desired sense and the sense incorrectly assigned by Zhong and Ng (2010) of the word "*director*" in that sentence are listed as follows.

(a) *member of a board of directors* (desired sense)

(b) *the person who directs the making of a film* (wrongly assigned sense)

Figure 1 shows this sentence and its dependency tree structure, and we want to disambiguate the sense of the word "*director*" (marked in red color), which should be the sense-(a) illustrated above. After having analyzed this sentence, we found that the most important word (or the key-context-word) for deciding the sense of "*director*" is "*board*" (in blue color), and other words in this sentence are either less relevant or irrelevant for deciding its sense. If only the local context is adopted to disambiguate the senses of "director" in this case (Zhong and Ng, 2010), then we need to use a 17-word window (centering on the target word "director") to extract the key word "*board*". In

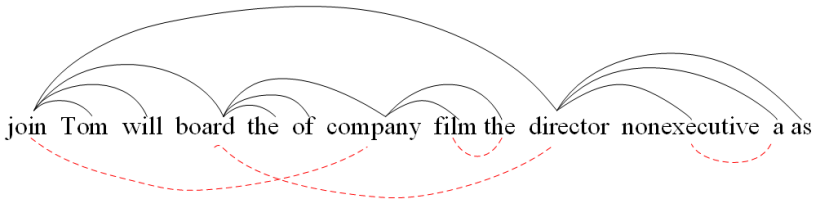

Fig. 3. The *Permuted-Lexical-Sequence* for the sample sentence in Figure 1.

this way, many noisy irrelevant words (e.g. "*film*", "*company*", etc.) would be also involved. As the consequence, it will tag the word "*director*" with the incorrect sense "*the person who directs the making of a film*" because the sense-(b) of "*director*" usually co-occurs with the phrase "*film company*".

To take care of sense dependency, Hatori et al. (2009) had made use of the tree structure in their model. And they achieved a good precision with a small corpus. However, they merely adopted a simple dependency model in which the sense of each target word only depends on the sense of its *reference-head lexicon*. With only one reference sense, as shown in the dependency tree of Figure 1, the sense of "*director*" will only depend on the sense of "*join*" (as it is the head). As the result, the sense of a head lexicon (e.g., "*director*" in Figure 1) will not depend on any other *non-local lexicons* under the same subtree. Therefore, *multi-reference lexicon dependency* (which denotes that the sense of each word would also depend on its sibling lexicons under the same subtree) is not taken consideration (i.e., "*director*" will not be related to "*board*" in Figure 2). In addition, they incorporated a large 60-word window which will import many irrelevant words (and involve considerable noisy features).

To handle the problems mentioned above, a novel approach which integrates *structural context* (which are the *head-lexicons* of those *child-nodes* under each associated syntactic subtree) with *local context* to disambiguate the word sense is proposed in this paper. This approach explicitly expresses the structural dependency between various senses via those associated syntactic subtrees, and let each word have one reference sense and various sibling reference lexicons under each associated subtree. Since a head word (e.g., a verb) might be simultaneously involved in several subtrees, it could have a different set of dependent lexicons under each associated subtree (e.g., a verb will be the head lexicon under both "*VP → VP NP*" and "*S →NP VP*" these two subtrees). Therefore, the sense of a head word would be decided via jointly considering all those associated lexicons under various syntactic relations.

Furthermore, we slightly modified the *head-percolation* rules (which specify a specific child-node under each subtree to percolate its head lexicon to the subtree root-node) to make them fit the WSD task better (e.g., for the PP-phrase "*as a nonexecutive director*" in Figure 2, we regard "*director*", not "*as*", as the head of this *PP* so that "*director*" can be related to "*board*" under an upper level subtree "*VP → VB NP PP*").

In order to investigate the performance of our model, we conduct several experiments on all-words WSD tasks. The results show that our model is significantly better (in statistical sense) than other state-of-the-art approaches. It thus illustrates that the proposed structural context cannot be ignored.

2 Proposed Approach

2.1 Generate Permuted-Lexicon-Sequence

Since the dependent words of a head word might scatter around it at either left side or right side, the search procedure would be quite complicated if we directly proceed the WSD along the original lexicon sequence from left to right. To have a straightforward procedure, we first permute the original lexicon sequence to move all the dependent words of each head word to its right hand side before we conduct a search. After a *Permuted-Lexicon-Sequence* is generated, the search can be proceeded strictly from left to right. The permutation is implemented via two steps described below.

Firstly, we use a parser to process the sentence, and get its phrase structure tree. For each sub-tree, we specify its *first-head-child-node* and *second-head-child-node* according to a few simple pre-specified precedence rules (e.g., *VP* > *NP* > *PP* > *MD*, if they co-occur under the same parent node). The dependence between various *child-nodes* is specified as follows: (1) Let each *non-head-child-node* (if it exists) under the sub-tree depend on both the *first-head-child-node* and the *second-head-child-node* (denoted as *first-reference-lexicon* and *second-reference-lexicon*); (2) Let the *second-head-child-node* depend on the *first-head-child-node*. Afterwards, we generate its corresponding *Permuted-Lexicon-Sequence* by permuting all *child-nodes* into the order "*First-Head* < *Second-Head* < *NonHead*" from the left to the right. Furthermore, we move all those monosemous words under each subtree to the left side of its *first-head-child-node* (because they will not depend on any other *child-nodes* under the subtree, as each of them has only one sense under WordNet (WSD is thus not necessary)). Afterwards, we perform the decoding process on this *Permuted-Lexical-Sequence* from left to right.

To illustrate the permutation procedure, Figure 2 shows the associated phrase structure tree (with the *head-child-nodes* marked) of the sentence given at Figure 1. After having marked the *head-child-nodes* of each sub-tree, we can extract the associated structural context of the lexicon "*director*" via following 4 steps.

Step 1: Regard the terminal-node "director" as the *first-head-child-node* of the sub-tree "*NN* → *director*", and it will be the head-lexicon of *NN*.

Step 2: As "*JJ*" and "*NN*" are the *second-head-child-node* and the *first-head-child-node*, respectively, of the sub-tree "*NP* → *DT JJ NN*", "*director*" will be further percolated to the sub-tree root-node "*NP*". Also, the head-lexicon of "*JJ*" (i.e., the word "nonexecutive") will depend on the word "director" (which is the head-lexicon of "*NN*"). Besides, as "*DT*" in this sub-tree is a *non-head-child-node*, its head-lexicon (i.e., the word "*a*") will depend on both "*nonexecutive*" and "*director*" which are the head-lexicons of "*NN*" and "*JJ*", respectively (called as the *first-reference-lexicon* and the *second-reference-lexicon*). Since *NN* is the *first-head-child-node* under "*NP* → *DT JJ NN*", we will continuously traverse to its parent subtree "*PP* → *IN NP*".

Step 3:. In the subtree "*PP* → *IN NP*", "*IN*" and "*NP*" are the *second-head-child-node* and the *first-head-child-node*, respectively. The head lexicon of "*IN*" (i.e., the word "*as*") will depend on the head lexicon of "NP" (i.e., the word "*director*").

As "*NP*" is the *first-head-child-node* under "*PP* → *IN NP*", we traverse again to its parent subtree "*VP* → *VB NP PP*".

Step 4: In the subtree "*VP* → *VB NP PP*", "*VB*" and "*NP*" are the *first-head-child-node* and the *second-head-child-node*, respectively. As "*PP*" is a *non-head-child-node*, the head lexicon of it (i.e., the target word "*director*") will depend on both the head-lexicons of "*VB*" and "*NP*" (the *first-reference-lexicon* and *second-reference-lexicon* of this subtree). Besides, the *second-reference-lexicon* (i.e., the word "*board*") also depends on the *first-reference-lexicon* (i.e., the word "*join*"). As "*PP*" is not the *first-head-child-node* of current subtree, the traversing procedure stops; otherwise, we will keep going until we reach the root of the whole tree.

Based on the method described above, we can find the dependency relationship between various terminal nodes of the parse tree in Figure 2. Figure 3 shows the associated *Permuted-Lexicon-Sequence* of that sentence, in which the black arc denotes the first reference dependency and the red arc denotes the second reference dependency.

2.2 Proposed Model

The task of WSD is to determine the correct senses of words in the given context. Given a sentence *snt*, let w_1^m denote the sequence of words $(w_1, w_2, \ldots, w_m)$ within the sentence to be assigned their senses, and s_1^m denote the corresponding sense sequence for w_1^m, then the word sense disambiguation problem can be formulated as: $\widehat{s_1^m} = arg\max_{s_1^m} P(s_1^m | w_1^m, snt)$, where m is the number of words to be assigned senses[1]. In the discriminative model adopted by Zhong and Ng (2010), the above $P(s_1^m | w_1^m, snt)$ is derived as follows.

$$P(s_1^m | w_1^m, snt) = \prod_{i=1}^{m} P\left(s_i | s_1^{i-1}, w_1^m, snt\right) \approx \prod_{i=1}^{m} P(s_i | w_i, snt) \quad (1)$$

However, if the associated parse-tree *pt* can be given, then $P(s_1^m | w_1^m, snt)$ will be re-formulated as:

$$P(s_1^m | w_1^m, snt) = \sum_{pt} P(s_1^m, pt \mid w_1^m, snt) \approx max_{Pt} P(s_1^m, pt \mid w_1^m, snt) \quad (2)$$

Where $P(s_1^m, pt \mid w_1^m, snt)$ can be further derived as follows.

$$P(s_1^m, pt \mid w_1^m, snt) = P(s_1^m | w_1^m, snt, pt) \times P(pt \mid snt) \quad (3)$$

We will first permute those m lexicons into its corresponding *Permuted-Lexicon-Sequence* LX_1^m according to the dependency relationship specified by the associated *parse-tree*. With LX_1^m specified above, $P(s_1^m | w_1^m, snt, pt)$ can be replaced with $P(LXS_1^m | LX_1^m, snt, pt)$, where LXS_1^m denotes a specific sense sequence assigned to LX_1^m.

[1] Which words should be assigned senses depends on the given task.

It is reasonable to assume that the sense assignment of each lexicon mainly depends on its local context and its structural context specified by the *parse-tree*. In the above formulation, for each permuted lexicon LX_i in LX_1^m, we will find its original location in the given sentence (call it <*i*>), and then extract its associated local context vector CLX_i which is a window $[w_{<i>-K}^{<i>+K}]$ around $w_{<i>}$ (which is LX_i) with the length "2*K*+1" (including $w_{<i>}$). Take the following sentence as an example:

(i) He works in a bank in the capital of his hometown.

For the word "*bank*" in this sentence, if we set K to 3, the local context will be the phrase "*works in a bank in the capital*". We will extract the position, POS, word form and local collocations (specified at (Zhong and Ng, 2010)) of each word from them (even they are not specified in WordNet). Take the word at the position <-2> (i.e., the word "*in*" at the left side) as an example, it is not defined in WordNet as it is not a content word; however, it still helps our disambiguation task, because it usually co-occurs with the "*bank*" when its sense is "*a building in which the business of banking transacted*".

Besides the local context, we also extract the *structural context sequence* of each lexicon from all its associated syntactic subtrees. Take the target word "*director*" in Figure 2 as an example, the procedure of extracting its structural lexicons is described as follows. In the Step 2 specified in the previous section, it shows that the associated context words under the sub-tree "*NP→DT JJ NN*" are "*a*" and "*nonexecutive*" (the head lexicons of "*DT*" and "*JJ*"). In the Step 3, we get only one associated context word "*as*" (the head lexicons of "*IN*") under the sub-tree "*PP -> IN NP*". Finally, in the Step 4, the associated context words obtained under the sub-tree "*VP-> VB NP PP*" are "*join*" and "*board*" (the head lexicons of "*VB*" and "*NP*"). Those extracted words make up the structural context sequence "*join board as a nonexecutive director*", and we can see that this sequence includes the key-lexicon "*board*" for disambiguating the sense of "*director*" without importing too many irrelevant words (such as "*the*", "*of*", "*the*", "*film*" and "*company*", if a large local context window is adopted). We will pack the lemmas, *POS*es and collocations of those words in this sequence as the *structural-lexicon-dependency* feature (denoted as *SLX*) to improve the performance.

Also, for each permuted lexicon LX_i, and for each subtree that it is involved, the *first-reference-lexicon* and *second-reference-lexicon* under the subtree will also be specified according to the procedure mentioned in Section 2.1. For each associated subtree, use the *Reference-Sense-Tuple <first-reference-lexicon-sense, second-reference-lexicon,* associated *production-rule>* to denote its corresponding structural context. The associated *structure-reference-information* for LX_i (denoted by $RXSI_i$) is then a set of such tuples derived from all its associated subtrees. Take the word "*director*" in Figure 2 as an example, its *structure-reference-information* $RXSI_i$ will involve three subtrees (i.e., "*NP→DT JJ NN*", "*PP -> IN NP*" and "*VP-> VB NP PP*"). And the corresponding tuple for the subtree "*VP-> VB NP PP*" would be *<assigned sense of "join", "board", "VP-> VB NP PP">*.

Assume that the assignment of the lexicon sense LXS_i (for LX_i) only depends on its *local context vector* CLX_i , *structural lexicon information* SLX_i and its associated *structure-reference-information* $RXSI_i$. Let $RLXI_i$ denote the associated set of *Reference-*

Lexicon-Tuple (which is obtained by replacing the first element "*first-reference-lexicon-sense*" of the corresponding *reference-sense-tuple* with "*first-reference-lexicon*"), then $RXSI_i$ can be obtained from $RLXI_i$ after all associated "*first-reference-lexicon-sense*" are given. Let t_1^m denote the corresponding POS-sequence for LX_1^m, then the original probability factor $P(LXS_1^m|LX_1^m, snt, pt)$ can be derived as follows.

$$\begin{aligned} & P(LXS_1^m|LX_1^m, snt, pt) \\ & \approx P(LXS_1^m|LX_1^m, t_1^m, CLX_1^m, SLX_1^m, RLXI_1^m) \\ & \approx \prod_{i=1}^{m} P\left(LXS_i|LX_i, t_i, CLX_i, , SLX_i, LXS_1^{i-1}, RLXI_i\right) \\ & \approx \prod_{i=1}^{m} P(LXS_i|LX_i, t_i, CLX_i, SLX_i, RXSI_i) \end{aligned} \quad (4)$$

To enhance the coverage rate of the test set, we will pool the training samples of various word-types (i.e., different LX_i) together by replacing their *LXS* and *first-reference-lexicon-sense* (in the tuple of *RLXI*) with their corresponding *synsets* defined in WordNet 3.1 (i.e., replacing " $P(LXS_i|LX_i, t_i, CLX_i, SLX_i, RXSI_i)$ " with "$P(synset_i|t_i, CLX_i, SLX_i, RXSI_i)$" in *Eq(4)*, in which LX_i has been dropped).

3 Evaluation

3.1 Data Sets

We train various models on Semcor corpus (Miller et al., 1993), and then conduct word sense disambiguation experiments on the test sets of senseval-2 (Palmer et al., 2001) and senseval-3 (Snyder et al., 2004). We choose these corpora because they are frequently used in evaluating WSD performance in the literature; and the quality of these corpora is good (Navigli, 2009).

Semcor corpus is constructed via annotating a subset of the English Brown Corpus (Kucera and Francis, 1967) with WordNet synsets (Miller et al., 1990; Fellbaum, 1998). It is the largest publicly available sense-tagged corpus. And we select two all-words test sets from Semantic Evaluation (Palmer et al., 2001; Snyder et al., 2004) (i.e., senseval-2 and senseval-3) as the test sets. These two testing sets are from WSJ articles and Brown Corpus.

3.2 Experiments

Experimental Setup.
We first use the *Berkeley parser*[2] to process the sentences extracted from the Semcor corpus (Miller et al., 1993) to get the phrase structure trees. As some of the sub-trees we parsed only have one child node, we will use the approach described in (Su et al., 1995) to normalize the trees. After this step, all the sub-trees (except the leaf nodes) in the phrase structure trees will have at least two child nodes.

[2] http://nlp.cs.berkeley.edu/software.shtml

We then mark the *head-child-node* for each sub-tree in the phrase structure trees. The rules that specify which syntactic label in the sub-tree should be the *head-child-node* are taken from *Penn2Malt*[3]. Afterwards, with the method presented in Section 2.1, we will permute the original sentence into its corresponding *Permuted-Lexicon-Sequence*. During the permutation, we also extract the *structural dependency* features and contextual features described in Section 2.2 from the phrase structure subtrees. We then use those *structural context* and *local contextual* features to train a *Maximum Entropy Classifier*[4]. When a test sentence is encountered, we will first obtain its *Permuted-Lexicon-Sequence*, as mentioned above, and then proceed the decoding on the *Permuted-Lexicon-Sequence*.

Results and Analysis.

	SE2
IMS	68.75%
Our model	69.59%
Rank-1 system (Palmer et al. 2001)	69.0%
Rank-2 system (Palmer et al. 2001)	63.6%
MFS	61.9%

Table 1. SE2 all-words task results. The improvement of our model over the IMS baseline is statistically significant ($p<0.05$).

	SE3
IMS	64.58%
T-CRF (Hatori et al. 2009)	65.40%
Our model	66.04%
IMS + adapted CW (Taghipour et al. 2015)	68.20%
PNNL (Tartz et al. 2007)	67.00%
MFS	62.37%

Table 2. SE3 all-words task results. The improvement of our model over the IMS baseline is statistically significant ($p<0.05$).

Table 1 and Table 2 show the performance of our system on senseval-2 and senseval-3 data-sets, respectively. In order to compare with those state-of-the-art systems, we also add those participants that were ranked within Top-2 in SE2 all-words task into Table 1 and Table 2 (the WordNet Most Frequent Sense "*MFS*" is also added as the lower bound). Those official scores are extracted from (Taghipour et al., 2015) and (Tartz et al., 2007). It should be noted that some systems (except IMS, T-CRF and our model) use additional training corpus, while we just use the Semcor corpus (Miller et al., 1993) as our training set. For example, "IMS + adapted CW" (Taghipour et al. 2015) adopted additional six parallel corpora and DSO corpus (Ng and Lee, 1996) as the training set,

[3] http://stp.lingfil.uu.se/~nivre/research/Penn2Malt.html
[4] http://homepages.inf.ed.ac.uk/lzhang10/maxent.html

Feature type	SE2	SE3
Local-context (*baseline*)	68.75%	64.58%
Structural-lexicon-dependency	63.20%	63.90%
Structural-sense-dependency	57.50%	61.82%
Local-context + structural-lexicon-dependency	69.04%	64.73%
Local-context + structural-sense-dependency	69.03%	65.84%
Local-context + structural-lexicon + structural-sense	69.59%	66.04%

Table 3. The performance for each dependency relation on these two datasets

	SE2			
POS	adj	noun	verb	adv
#Tokens	404	1065	535	265
IMS	72.03%	75.39%	46.54%	81.89%
Our model	72.28%	75.96%	48.60%	82.26%
Diff.	+0.25%	+0.57%	+1.06%	+0.37%

Table 4. The performance for each *POS* on SE2 all-words task.

besides, it used three large corpus to train the required word embedding; and "PNNL" use additional OMWE 1.0 (Chklovski and Mihalcea, 2002) and example sentences in WordNet as the training corpus. Therefore, their performances cannot be directly compared with that of ours (i.e, only IMS and T-CRF can be directly compared).

From these two tables, we can see that the performance has been improved significantly ($p<0.05$) over the baseline on both datasets. This shows that the *structural context* features are very useful for WSD. Table 3 investigates the individual contribution from each set of those adopted feature sets (i.e., *local context* features, *structural sense dependency* features, and *structural lexicon dependency* features), which shows that *local context* feature is the most effective feature set, but other two feature sets are also helpful.

Table 3 shows the effect of each feature on these two datasets. When we adopt just one type of feature, the *local context* feature is the best. This is because the associated *reference-lexicons* are within the *local context* window about 66% of the time (we got the ratio of reference lexicons within the *local context* window are 66.34% and 65.60% on SE2 and SE3, respectively). However, to further improve the performance, the remaining 34% cases with complex *structural dependency* should also be taken care. Besides, when we add the *structural context* features to the model, the improvement on senseval-3 is better than senseval-2 (In Table 3, when we add these two *structural context features*, the improvement on SE3 is *1.46%* while the improvement on SE2 is just *0.84%*). The reason for that is that senseval-3 contains more words whose dependency relation is complex (As we calculated, the ratio of reference lexicons without local context window on SE3 is bigger than the ratio on SE2).

In Table 4, we present the performance of each *POS* on Senseval2 all-words task. From this tables, we find that the influences of structural context on each *POS* category are different. The distribution and size of the samples may have an influence on the results, however, we can still see that it can improve the performance of *noun* and *verb* words significantly. And it also has a little positive influence on *adj* and *adv* words.

This is also true for SE3. This phenomenon matches the observation that the *long distance dependency* and *multi-reference dependency* usually exist between *verb* and *noun* words, while the *adj* and *adv* words frequently only depend on the local context. As the use of *structural lexicon dependency* features, we can see the performance of *adj* and *adv* words also improves. In summary, the *structural dependency* we proposed contributes more to the words with complex dependency relations.

4 Related work

WSD is a well-known topic, and many related papers have been published. Navigli (2009) had given a good survey of this field. Based on the classification method adopted, the task of WSD could be divided into (1) Supervised (Tratz et al., 2007; Hatori et al., 2009; Zhong and Ng, 2010; Chen et al., 2014), (2) Unsupervised (Agirre et al., 2011; Chen et al., 2009), and (3) Semi-supervised (Mihalcea, 2004) approaches. Among them, the supervised approach gives the best performance so far. As our method is a supervised method for all-words WSD (Hatori et al., 2009; Zhong and Ng, 2010; Taghipour and Ng, 2015), we will focus and introduce this kind of approaches in the following.

Zhong and Ng (2010) proposed a WSD system based on supervised learning, and achieved state-of-the-art results on several Senseval and Semeval evaluations. They adopted POS tags, content words and collocations in a 7-word local window as features, and used a SVM to perform classification. In comparison with our approach, they ignored the structural dependency and did not consider the correlation between various senses.

On the other hand, Hatori et al. (2009) considered the structural dependency (via a dependency tree) in addition to the local context mentioned above. They described these dependencies on the tree-structured conditional random fields. Furthermore, they incorporated these sense dependencies in combination with various coarse-grained sense tag sets, which are expected to relieve the data sparseness problem, and enable their model to work even for words that do not appear in the training data. Their approach was shown to be comparable to those state-of-the-art systems on Senseval datasets. In comparison with our approach, they adopted a large 60-word context window, which would involve many irrelevant words and thus introduce additional noisy information. Also, each sense only depends on one reference sense in their model, which is inadequate in many cases.

5 Conclusion

To correctly classify each content word in the sentence, not only *local context* but also *structural context* (which is mainly responsible for handling long distance sense/lexicon dependency) is required. To take the *structural context* into account without introducing too much additional noisy information, we propose a new approach to describe various syntactic dependency relations between different words. In this approach, after parsing a sentence into its phrase structure tree, we mark two *head-child-nodes* under

each sub-tree. Then we can use these *head-child-nodes* and syntactic subtrees to describe the *long distance dependency* and *multi-reference dependency* (which lets each target word be capable of depending on several *non-local* words).

Our contributions include: (1) Proposing a novel model to represent different dependency relations between various senses, which is able to handle the long distance *multi-reference dependency* that has not been touched in those previous WSD tasks. (2) Proposing a way to permute the original lexicon sequence to improve the search efficiency. (3) Showing that the structural dependency relations are useful for distinguish the senses of words with complex dependency relations.

6 Acknowledgements

The research work has been funded by the Natural Science Foundation of China under Grant No. 61333018.

Reference

1. Eneko Agirre, Oier Lopez de Lacalle, and Aitor Soroa. 2014. Random Walks for Knowledge-Based Word Sense Disambiguation. *Computational Linguistics*, 40(1), pages 57-84.
2. Miller, G. A., Beckwith, R., Fellbaum, C. D., & Gross, D. Miller. K. 1990. WordNet: An online lexical database. *Int. J. Lexicograph*, 3(4), 235-244.
3. Bettina Berendt and Roberto Navigli. 2006. Finding your way through blogspace: Using semantics for cross-domain blog analysis. In Proceedings of the American Association for Artifical Intelligence, pages 1-8.
4. Marine Carpuat and Dekai Wu. 2005. Word sense disambiguation vs. Statistical machine translation. In *proceedings of the 43rd Annual Meeting of the ACL*, pages 387-394.
5. Marine Carpuat and Dekai Wu*. 2007. Improving Statistical Machine Translation using Word Sense Disambiguation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 61-72.
6. Yee Seng Chan, Hwee Tou Ng and David Chang. 2007. Word Sense Disambiguation Improves Statistical Machine Translation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 33-40.
7. Ping Chen, Wei Ding, Chris Bowes and David Brown. 2009. A Fully Unsupervised Word Sense Disambiguation Method Using Dependency Knowledge. *Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the ACL, pages 28-36*.
8. Xinxiong Chen, Zhiyuan Liu, and Maosong Sun. 2014. A Unified Model for Word Sense Representation and Disambiguation,. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing* (EMNLP), pages 1025-1035.
9. Chklovski, T. and R. Mihalcea (2002) Building a sense tagged corpus with open mind word expert. In *Proceedings of the ACL-02 workshop on Word sense disambiguation: recent successes and future directions*.
10. Fellbaum, Christiane. 1998. WordNet: An electronic database. *MIT Press, Cambridge, MA.*

11. Katrin Erk and Sebastian Pado. 2008. A Structured Vector Space Model for Word Meaning in Context, *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 897–906.
12. William A. Gale, Kenneth W. Church and David Yarowsky. 1992. A method for disambiguation word senses in a corpus. *Computers and the Humanities*, 26(5-6), pages 415-439.
13. Jun Hatori, Yusuke Miyao and Jun'ichi Tsujii. 2009. On contribution of sense dependencies to word sense disambiguation. *Information and Media Technologies*, 4(4), pages 1129-1155.
14. Mihalcea, Rada, Paul Tarau, and Elizabeth Figa. 2004. PageRank on semantic networks, with application to word sense disambiguation. *In Proceedings of the 20th international conference on Computational Linguistics. Association for Computational Linguistics.*
15. George A. Miller, Claudia Leacock, Randee Tengi and Ross T. Bunker. 1993. A semantic Concordance. In *Proceedings of the workshop on Human Language Technology.*
16. Roberto Navigli. 2009. Word sense disambiguation: A survey. *ACM Computing Surveys*, 41(2), Article 10.
17. Hwee Tou Ng and Hian Beng Lee. 1996. Intergrating multiple knowledge sources to disambiguate word sense: An exemplar-based approach. In *Proceeding of the 34th Annual Meeting of the Association for Computational Linguistic (ACL)*, pages 40-47.
18. Martha Palmer, Christiane Fellbaum, Scott Cotton, Lauren Delfs, and Hoa Trang Dang. 2001. English Tasks: All-Words and Verb Lexical Sample. In *Proceedings of the Second International Workshop on Evaluating Word Sense Disambiguation Systems (SENESEVAL-2)*.
19. Benjamin Snyder and Martha Palmer. 2004. The English All-Words Task. In *Senseval-3: Third International Workshop on the Evaluation of the Systems for the Semantic Analysis of Text.*
20. Christopher Stokoe, Michael P. Oakes and John Tait. 2003. Word sense disambiguation in information retrieval revisited. In *Proceedings of the Twenty-Sixth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 159-166.
21. Keh-Yih Su, Jing-Shin Chang and Yu-Ling Una Hsu. 1995. A Corpus-Based Statistic-Oriented Two-Way Design for Parameterized MT Systems: Rationale, Architecture and Training Issues. In Proceedings of the Sixth International Conference on Theoretical and Methodological Issues in Machine Translation (TMI-95), Vol. 2, pages 334-353, Leuven, Belgium.
22. Kaveh Taghipour and Hwee Tou Ng. 2015 Semi-Supervised Word Sense Disambiguation Using Word Embeddings in General and Specific Domains. *The 2015 Annual Conference of the North American Chapter of the Association for Computational Linguistics.*
23. Stephen Tratz, Antonio Sanfilippo, Michelle Gregory, Alan Chappell, Christian Posse and Paul Whitney. 2007. PNNL:A Supervised Maximum Entropy Approach to Word Sense Disambiguation. In *Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007)*, pages 264-267.
24. M. Turing, 1950. Computing machinery and intelligence. *Mind*, 54, pages 443-460.
25. Evgenia Wasserman-Pritsker, William W.Cohen and Einat Minkov. Learning to Identify the Best Contexts for Knowledge-based WSD.
26. Zhi Zhong and Hwee Tou Ng. 2010. It Makes Sense: A Wide-Coverage Word Sense Disambiguation System for Free Text. In *Proceedings of the ACL 2010 System Demonstrations*.