Word Sense Disambiguation through Sememe Labeling

***Xiangyu Duan, *Jun ZHAO, **BO XU**
Institute of Automation, Chinese Academy of Sciences
National Laboratory of Pattern Recognition
Eastern Road of ZhongGuan Cun, 95#, 100080
*{xyduan, jzhao}@nlpr.ia.ac.cn, **xubohitic.ia.ac.cn

Abstract

Currently most word sense disambiguation (WSD) systems are relatively individual word sense experts. Scarcely do these systems take word sense transitions between senses of linearly consecutive words or syntactically dependent words into consideration. Word sense transitions are very important. They embody the fluency of semantic expression and avoid sparse data problem effectively. In this paper, HowNet knowledge base is used to decompose every word sense into several sememes. Then one transition between two words' senses becomes multiple transitions between sememes. Sememe transitions are much easier to be captured than word sense transitions due to much less sememes. When sememes are labeled, WSD is done. In this paper, multi-layered conditional random fields (MLCRF) is proposed to model sememe transitions. The experiments show that MLCRF performs better than a base-line system and a maximum entropy model. Syntactic and hypernym features can enhance the performance significantly.

1 Introduction

Word sense disambiguation (WSD) is one of the important tasks in natural language processing. Given the context of a word, a WSD system should automatically assign a correct sense to this word. WSD has been studied for many years. Current word sense disambiguation systems are mainly individual word sense experts [Ide *et al.*, 1998]. In these systems, there are mainly two categories of the target word's contexts. One category is the words in some windows surrounding the target word, the other category is the relational information, such as syntactic relation, selectional preferences, etc. But senses of contextual words, which may also need to be disambiguated, attract little attention. In this paper, senses of contextual word and the target word are simultaneously considered. We try to incorporate sense transitions between every contextual word and the target word into one framework, and our target is to disambiguate all words through determining word sense transitions.

Some scholars have done relative researches on globally modeling word sense assignments. Some methods have been proposed, including GAMBL (a cascaded memory-based classifiers) [Decadt *et al.*, 2004], SenseLearner [Mihalcea and Csomai, 2005], Naïve Bayes methods [Yuret, 2004].

Most of these methods still based on individual word experts while our motivation is to model word sense transitions. This is a different view of globally modeling word sense assignments. In a sentence with no syntactic information, word sense transitions take the form of linear chain. That is, transitions are from left to right, word by word. In a sentence with dependency syntactic information, word sense transitions are from all children to their parent. Although linear chain word sense transitions had been studied, by using simulated annealing [Cowie *et al.*, 1992] and unsupervised graph-based algorithm [Mihalcea, 2005], the number of word sense transitions is so tremendous that they are not easy to be captured. In order to circumvent this shortcoming, we adopt HowNet (http://www.keenage.com) knowledge base to decompose every word sense into several sememes (usually no more than 6 sememes). HowNet defines a closed set of sememes that are much less than word senses. But various combinations of these sememes can describe various word senses. It is more practical to model sense transitions through sememes than through pure word senses.

Then one transition between two words' senses becomes multiple transitions between sememes. In this paper, we propose multi-layered conditional random fields (MLCRF) to model sememe transitions. As we know, conditional random fields (CRF) can label nodes of structures like sequences and trees. Consecutive labels in CRF also constitute label transitions. Therefore, the problem of modeling sememe transitions can also be viewed as sememe labeling problem.

There are researches on HowNet-based WSD. Wong and Yang [2002] have done a similar work to ours. Just like POS tagging, they regarded WSD as word sense tagging and adopted a maximum entropy approach to tag senses. In section 4.2, we made some comparisons. The experiments show that MLCRF performs better than a base-line system and Wong and Yang's maximum entropy model. Syntactic and hypernym features can enhance the performance significantly.

2 An Introduction of HowNet

HowNet [Zhendong Dong *et*, 2006] is a bilingual common-sense knowledge base. One important component of HowNet is the knowledge dictionary, which covers over 6,500 words in Chinese and close to 7,500 English equivalents. We use this knowledge dictionary to generate candidate senses of a word. In HowNet, each candidate sense of one word is a combination of several sememes. For example, a sense definition of the Chinese word "research institute" is as follows:

DEF=InstitutePlace, *research, #knowledge

where word sense (DEF) is split into sememes by commas. The sememes in this example are "InstitutePlace", "research", "knowledge". Symbols preceding sememes represent relations between the entry word and the corresponding sememe. In this example, symbols are "*" and "#". Symbol "*" represents agent-event relation, "#" represents co-relation. This word sense (DEF) can be glossed as: "research institute" is an "InstitutePlace", which acts as an agent of a "researching" activity, and it has a co-relation with "knowledge".

Sememes are the most basic semantic units that are non-decomposable. It is feasible to extract a close set of sememes from Chinese characters. This is because each Chinese character is monosyllabic and meaning bearing. Using this closed set of sememes, HowNet can define all words' senses through various combinations of these sememes.

A word sense is defined by sememes according to an order. The first sememe in a word sense definition represents the main feature of the sense of that word, and this main feature is always the category of that sense. In the example mentioned above, the main feature of the word "research institute" is "InstitutePlace". Other sememes in a word sense are organized by an order of importance from the HowNet's point of view. In our current system, symbols representing relations are omitted.

Totally, HowNet contains 1,600 sememes, which are classified into 7 main classes, including "entity", "event", "attribute", "attribute value", "quantity", "quantity value", "secondary feature". These 7 main classes are further classified hierarchically. In the end, all of 1,600 sememes are organized as 7 trees. Each sememe besides the root sememe has a hypernym (A is a hypernym of B if B is a type of A). For example, "InstitutePlace" has a hypernym "organization", "organization" has a hypernym "thing", "thing" has a hypernym "entity". The hypernym feature shows usefulness in our word sense disambiguation system.

3 System Description

3.1 Task Definition

Our task is to disambiguate word senses through sememe labeling. Here is the formal description of our task. In what follows, $\boldsymbol{X}$ denotes a sentence. The *i*th word of $\boldsymbol{X}$ is denoted by x_i, and the sentence's length is *n*. According to HowNet, the sense of each x_i is decomposed into *m* layers of sememes denoted by $y_{i1},...y_{im}$, where *m* is an empirical parameter depending on how much layers of sememes bound together can differentiate word senses from each other. Please note that $y_{i1},...y_{im}$ are ordered by decreasing importance. For example, y_{i1} is the first sememe in the sense definition of x_i. With some simplification of the HowNet knowledge dictionary, word senses can distinguish each other by 2 layers of sememes. If sememes are taken as labels, every word x_i has 2 layers of labels. Word sense disambiguation becomes a 2-layered labeling problem. Sememe labeling can be carried on flat sentences (sequences) or dependency syntactic trees. This is illustrated in Figure 1.

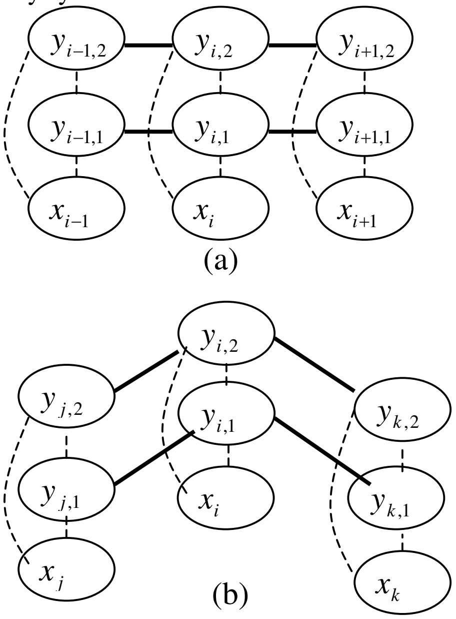

Figure 1. Graphical representation of words and sememe layers. (a) is the sequence representation, and (b) is the tree representation. Solid lines show sememe transitions, dashed lines show related links. Please note that the direction of sememe transition is different for sequences and trees. For sequences, the direction is from left to right. For trees, the direction is from child node to its parent node. Although sememes can depend on any word, higher layer sememes can depend on lower layer sememes, for clarity, we do not show these cross links. For more detailed description, see the feature set of section 3.5.

Sememes per layer constitute a label configuration of the sentence. These *m* layer configurations of the sentence are denoted by $Y_1,..Y_m$. Our aim is to find:

$$\hat{Y}_1...\hat{Y}_m = \underset{Y_1...Y_m}{\arg\max} P(Y_1...Y_m \mid X) \qquad (1)$$

We propose a layer-by-layer strategy "MLCRF" to solve this equation. Current layer sememe labeling can use labeled pre-layer sememes as features. The order of the strategy is bottom up, from 1st layer to *m*th layer. To introduce MLCRF, we introduce conditional random fields in the following section.

3.2 Introduction of Conditional Random Fields

Conditional random fields (CRF) [Lafferty *et al.*, 2001] are undirected graphical models used to calculate the conditional probability of a set of labels given a set of input values. We cite the definitions of CRF in [McCallum, 2003]. It defines the conditional probability proportional to the product of potential functions on cliques of the graph,

$$P(Y \mid X) = \frac{1}{Z_X} \prod_{c \in \boldsymbol{C}(Y,X)} \phi_c(Y_c, X_c) \qquad (2)$$

where $\boldsymbol{X}$ is a set of input random variables and $\boldsymbol{Y}$ is a set of random labels. $\phi_c(Y_c, X_c)$ is the clique potential on clique *c*. $\boldsymbol{C}(\boldsymbol{Y},\boldsymbol{X})$ is the set of all cliques of the graph. In CRF, clique is often an edge with a node on each end. Y_c represents labels of the two end nodes. Clique potential is often defined as:

$$\phi_c(Y_c, X_c) = \exp(\sum_{r=1}^{R} \lambda_r f_r(Y_c, X_c)) \qquad (3)$$

where f_r is an arbitrary feature function over its arguments, λ_r is a learned weight for each feature function, *R* is the total number of feature functions of the clique. Z_X is a normalization factor over all output values,

$$Z_X = \sum_{Y'} \prod_{c \in C(Y',X)} \phi_c(Y_c^{'}, X_c) \qquad (4)$$

Two dynamic programming algorithms are employed: Viterbi for decoding and a variant of the forward-backward algorithm [Sha and Pereira, 2003] for the computing of expectations and normalization.

Two kinds of CRF are used in our system: sequential CRF and tree CRF. Sequential CRF is like traditional linear chain CRF, while tree CRF is slightly different. In linear chain CRF, one clique is composed of current word and its pre-word. In tree CRF, one clique is composed of current word and one child of it. Tree CRF adopts sum-product algorithm [Pearl, 1988], which is an inference algorithm in graphical models. Cohn and Blunsom [2005] have used tree CRF to solve semantic role labeling. But the algorithm was not presented. In this paper, we take sum-product algorithm into a forward-backward frame. Each node in the tree has several sub forward (backward) vectors. The detailed description of the algorithm is not included in this paper for the limit of the paper length.

3.3 Multi-layered Conditional Random Fields

In multi-layered conditional random fields, the conditional probability is as follows:

$$P(Y_1..Y_m \mid X) = P(Y_1 \mid X)P(Y_2 \mid Y_1,X) \cdots\cdots P(Y_m \mid Y_{m-1},\ldots Y_1,X)$$
$$= \frac{[\prod \phi_c]_1}{Z_X} \cdot \frac{[\prod \phi_c]_2}{Z_{X,Y_1}} \cdots\cdots \frac{[\prod \phi_c]_m}{Z_{X,Y_1,\ldots Y_{m-1}}} \qquad (5)$$

where $[\prod \phi_c]_i$ denotes the product of clique potentials of the *i*th layer. Z_X etc denotes normalization factor. For example,

$$[\prod \phi_c]_m = \prod_c \exp(\sum_{r=1}^{R} \lambda_r \cdot f_r(Y_{c,m}, Y_{c,m-1}, \ldots Y_{c,1}, X)) \qquad (6)$$

$$Z_{X,Y_1,\ldots Y_{m-1}} = \sum_{Y_m} [\prod \phi_c]_m \qquad (7)$$

where $Y_{c,m}$ denotes the pair of *m*th layer labels of the words of clique *c*.

$P(Y_1 \mid X)$, $P(Y_2 \mid Y_1, X)$, …, $P(Y_m \mid Y_{m-1},\ldots,Y_1,X)$ represent *1, 2, …, m*th layer probabilities respectively. Each layer CRF model can be trained individually, using sequential CRF or tree CRF.

Sutton *et al.* [2004] use dynamic CRF to perform multiple, cascaded labeling tasks. Rather than perform approximate inference, we perform exact inference. To avoid exponential increase, we propose a multi-layered Viterbi algorithm for decoding.

To illustrate multi-layered Viterbi, let us look back on single layer decoding. For single layer, the most probable *Y* is as follows:

$$\hat{Y} = \arg\max_Y (P(Y \mid X)) = \arg\max_Y (\lambda \cdot F(Y,X)) \qquad (8)$$

where $F(Y,X) = \sum_{c \in C} f(Y_c, X_c)$. The denominator Z_X is omitted because Z_X is the same for all $\boldsymbol{Y}$. But in multi-layered case, the next layer decoding will use pre layer labels as features, so the denominators are no longer the same.

For the ease of explanation, let us consider linear chain CRF of single layer. If the denominator Z_X is not omitted for one layer, it can also be expressed as:

$$Z_X = sum(\alpha_1) \cdot \frac{sum(\alpha_2)}{sum(\alpha_1)} \cdots\cdots \frac{sum(\alpha_n)}{sum(\alpha_{n-1})} \qquad (9)$$

where α_i denotes the forward vector at the *i*th position of the chain. $sum(\alpha)$ denotes the sum of the vector α's elements. In CRF, the sum of the forward vector α_i's elements is the sum of all possible paths up to the *i*th position. So $sum(\alpha_n)$ is equal to Z_X.

Then the linear chain probability is as follows:

$$\log(P_\lambda(Y \mid X)) = \boldsymbol{\lambda} \cdot f(Y_1,X_1) - \log(sum(\alpha_1)) +$$
$$\boldsymbol{\lambda} \cdot f(Y_2,X_2) - \{\log(sum(\alpha_2)) - \log(sum(\alpha_1))\} + \ldots$$
$$\boldsymbol{\lambda} \cdot f(Y_n,X_n) - \{\log(sum(\alpha_n)) - \log(sum(\alpha_{n-1}))\} \qquad (10)$$

Equation 8 shows that the linear chain probability can be computed dynamically. At each time step, we can get probability up to that time. Based on this single layer equation, we can derive the multi-layered Viterbi algorithm.

With some simplification of the HowNet knowledge dictionary, word senses can distinguish each other by 2 layers of sememes. We present a 2-layer Viterbi for sequential CRF in Figure 2. Multi-layered Viterbi can be extended from 2-layer Viterbi. In Figure 2, all forward vectors α and transition matrices *M* are only about second layer. S_i denotes a combined label of 2 layers at *i*th position, $S_{i,2}$

denotes the second layer label at *i*th position. Suppose that the label set of each layer contains *k* elements, then S_i has *k*k* kinds of 2-layer combinations.

In 2-layer Viterbi, we do not need transition matrices and forward vectors of 1st layer because Z_X is the same for all Y_1. But for 2nd layer, the normalizing factor is Z_{X,Y_1}, where Y_1 is the 1st layer sememes that are related to 1st layer decoding. So forward vectors and transition matrices are needed to calculate normalization factor for the 2nd layer only. $\alpha_i(s_{i-1,2}, s_{i,2})$ is the forward vector of 2nd layer at *i*th position. $M_i(s_{i-1,2}, s_{i,2})$ is the transition matrix of 2nd layer at *i*th position. $[\boldsymbol{\lambda} \cdot \boldsymbol{f}]_{i,1}$ and $[\boldsymbol{\lambda} \cdot \boldsymbol{f}]_{i,2}$ are the logarithm value of 1st layer and 2nd layer's *i*th clique potential respectively.

```
Initial value: $Val(s_0) = 0$, $s_0$ is the initial state
$\alpha'_0(s_{0,2}) = [1...0]$
for every position $i$:
    for every candidate 2-layer label $s_i$:
    $\alpha_i(s_{i-1,2}, s_{i,2}) = \alpha'_{i-1}(s_{i-1,2}) M_i(s_{i-1,2}, s_{i,2})$
    $Val(s_i) = \max_{s_{i-1}} \{Val(s_{i-1}) + pair(s_{i-1}, s_i)\}$
    $pair(s_{i-1}, s_i) = [\boldsymbol{\lambda} \cdot \boldsymbol{f}]_{i,1} + [\boldsymbol{\lambda} \cdot \boldsymbol{f}]_{i,2} -$
    $\{\log[sum(\alpha_i(s_{i-1,2}, s_{i,2}))] - \log[sum(\alpha'_{i-1}(s_{i-1,2}))]\}$
    $s'_{i-1} = \arg\max_{s_{i-1}} \{Val(s_{i-1}) + pair(s_{i-1}, s_i)\}$
    $\alpha'_i(s_{i,2}) = \alpha_i(s'_{i-1,2}, s_{i,2})$
till the end node
```

Figure 2. 2-Layer Viterbi algorithm for Sequential CRF. s'_{i-1} record the optimal path.

The time complexity of 2-layer Viterbi for one sentence is $\boldsymbol{O}(J^2 K^4 T)$, where *J* is the number of 1st layer labels, *K* is the number of 2nd layer labels, *T* is the length of the sentence. Figure 2 mainly deals with sequence. When applying it to trees, sub α s for every tree node should be introduced. In our problem for WSD, S_i is the candidate sense at position *i*, which is less than 2-layer combinations of entire labels.

3.4 Reducing the Number of Candidate Labels

In common CRF, every input variable x_i has several candidate labels. There are 1,600 sememes in HowNet. For our WSD problem, it is impractical to treat all these sememes as candidate labels for the training of single layer CRF.

Using the HowNet knowledge dictionary we can generate candidate sememes much less. We propose two methods to reduce the number of candidate sememes. One is DefSpace and the other is ExDefSpace. In the following, we will introduce how to apply these two methods to first layer CRF in detail.

DefSpace

For DefSpace, candidate first layer sememes of a word are the first layer sememes appeared in the definitions of this word in HowNet. DefSpace generates a real space that only includes the labels (sememes) listed in the definitions of the entry word. For example, Chinese word ***fazhan*** has 3 senses in the HowNet dictionary: "CauseToGrow", "grow" and "include". Since all these senses are single sememes, candidate sememes of the first layer are these 3 sememes for DefSpace.

ExDefSpace

ExDefSpace is the extended version of DefSpace. Suppose there are two words *a* and *b*. According to the HowNet dictionary, *a* has 2 candidate first layer sememes sem_1 and sem_2, *b* has 2 candidate first layer sememes sem_2 and sem_3. Suppose *a* appears in the training corpus while *b* does not appears. Considering training examples of *a* with sem_1 or sem_2, for DefSpace, the training procedure only discriminates sem_1 and sem_2, while sem_2 and sem_3 are not discriminated in this discriminative training. Then when *b* appears in test data, it can't be disambiguated. To overcome this shortcoming of DefSpace, candidate sememes of a word are extended. For every sememe in HowNet, we build a candidate sememe list. Concretely speaking, for every sememe *s*, we search the whole dictionary entries to find sememes of first layer that together with *s* define a common polysemous word. All such sememes are put into the candidate sememe list of *s*. Then during training, the candidate sememes of a word are generated according to the hand labeled sememe's candidate list. For above example, sem_3 is added to the candidate lists of sem_1 and sem_2. Then word *a* provides a positive example of sem_1 or sem_2, but a negative example of sem_3. Also, if one sem_3 is tagged in the training corpus, it provides a negative example of sem_1 or sem_2.

Finally, in a sememe's candidate sememe list, there are average 11.76 candidate sememes.

Candidate second layer sememes of a word are generated according to this word's entries in HowNet whose first layer sememes are the same as the labeled first layer sememe. This is like DefSpace used in first layer CRF.

In the phase of decoding (multi-layered Viterbi), only multiple senses of a word are regarded as candidate senses of this word. No special measure like ExDefSpace is taken.

3.5 Feature Set

In MLCRF, training is taken layer by layer. For first layer, we use the following factored representation for features.

$$f(Y_c, X_c) = p(X, c) q(y'_c, y_c) \quad (11)$$

where $p(X, c)$ is a binary predicate on the input $\boldsymbol{X}$ and current clique *c*, $q(y'_c, y_c)$ is a binary predicate on pairs of labels of current clique *c*. For instance, $p(X,c)$ might be " whether word at position *i* is ***de***".

Table 1 presents the categories of $p(X,c)$ and $q(y_c^{'},y_c)$.

$q(y_c^{'},y_c)_{seq}$	$p(X,c)_{seq}$	$q(y_c^{'},y_c)_{tree}$	$p(X,c)_{tree}$
s_i_0 *s_i-1_0* *s_i_0,s_i-1_0* ^ *s_i_0* ^ *s_i-1_0*	true	*s_n_0* *s_c_0* *s_c_0,s_n_0* ^*s_n_0* ^*s_c_0*	true
s_i_0	uni_w bi_w uni_p bi_p tri_p (from i-2 to i+2)	*s_n_0*	uni_w bi_w uni_p bi_p tri_p (from *ff* to *cl* and *cr*)

Table 1. feature set of sequential CRF and tree CRF

For first layer, a label of the clique *c* is the first layer sememe. In table 1, *s_i_0* denotes 1st layer sememe of *i*th word in sequential CRF and *s_n_0* denotes 1st layer sememe of current node *n* in tree CRF. "^" denotes hypernym, *uni*, *bi* and *tri* denote unigrams, bigrams and trigrams respectively, *w* denotes word, *p* denotes POS. The subscript *seq* and *tree* represent sequential CRF and tree CRF respectively. For tree CRF, *c* denotes the child of *n* in current clique. *cl* and *cr* are the left and right sibling of node *c*. *f* denotes the parent node of *n*, *ff* denotes grandfather of *n*.

For second layer, s_i_0 (s_n_0) is replaced by s_i_1 (s_n_1). *p*(***X***,*c*) excluding *true* are appended by s_i_0 (s_n_0). For *true* value, when *q* is about s_i_1 (s_n_1) or s_i-1_1 (s_c_1), additional feature s_i_0 (s_n_0) is added to *p*(***X***,*c*).

4 Experiments and Results

4.1 Experimental Settings

Sememe (HowNet) based disambiguation is widely used in Chinese WSD such as Senseval 3 Chinese lexical sample task. But full text disambiguation at sememe level is scarcely considered. Furthermore, the full text corpus tagged with WordNet like thesaurus is not publicly available. The sentences used in our experiments are from Chinese Linguistic Data Consortium (http://www.chineseldc.org) that contains phrase structure syntactic trees. We can generate gold standard dependency trees using head rules. We selected some dependency trees and hand tagged senses on them with over 7,000 words for training and over 1,600 words for testing. There are two taggers to tag the corpus and an adjudicator to decide the tagging results. The agreement figure is the ratio of matches to the total number of annotations, and the figure is 0.88. This corpus is included in ChineseLDC.

For comparison, we build a base-line system and duplicate Wong and Yang's method [Wong *et al*., 2002]. The base-line system predicts word senses according to the most frequent sense of the word in the training corpus. For words that do not appear in the training corpus, we choose the first sense entry in the HowNet dictionary. We also imitate Wong's method. Their method is at a word level. The sense of a word is the first sememe optionally combined with the second sememe. They defined this kind of word sense as categorical attribute (and also a semantic tag). Then an off-the-shelf POS tagger (MXPOST) is used to tag these senses.

In our system, named entities have been mapped to predefined senses. For example, all *person* named entities have the sense "human|人". For those words that do not have entries in dictionary, we define their sense by referring to their synonyms that have entries. In the end, polysemous words take up about 1/3 of the corpus, and there are average 3.7 senses per polysemous word.

In the following sections, precisions of polysemous words are used to measure the performance. Since all words have entries in dictionary or can be mapped to some senses, recall is the same with precision. Note that for single layer sememe labeling, precision is the rate of correct sememes of the corresponding layer. Sense precision is the rate of correct senses, not sememes.

4.2 Results of MLCRF in WSD

	1st layer		2nd layer		Sense
	c	+h	c	+h	
Seq_{Def}	81.0	78.8	87.2	85.1	79.9
Seq_{ExDef}	80.6	78.5	-	-	79.3
$Tree_{Def}$	82.6	84.0	90.0	95.9	85.9
$Tree_{ExDef}$	82.0	83.5	-	-	85.1

Table 2. Performance of MLCRF in WSD. Numbers are precisions of polysemous words with respect to first layer, second layer and whole word sense.

Table 2 shows the performance of MLCRF in WSD. "*c*" represents common features that presented in table 1 except hypernym features. "+*h*" means common features plus hypernym features. Seq_{Def} (Seq_{ExDef}) denotes sequential CRF using DefSpace (ExDefSpace). $Tree_{Def}$ ($Tree_{ExDef}$) denotes tree CRF using DefSpace (ExDefSpace). From table 2, we can see that tree CRF performs better than sequential CRF, which shows that sense transitions in trees are easier to be captured than in sequences. Hypernym features enhance performance of tree CRF partly because it avoids sparse data problem, while it adds noise to sequential CRF. ExDefSpace performs similarly as DefSpace, which maybe due to our relatively small corpus.

In the test set, there are 203 different types of 502 polysemous words. Our best result correctly label 57 word senses that are labeled wrongly by baseline system. Of these 57 polysemous words, there are 36 word types, of which 16 word types never occur in the training corpus. Although baseline system also wins our method a few words, totally our method improves 6.9% over baseline.

Our dependency trees are golden standard. We insist that parsing and WSD should not be processed in a pipe line mode. WSD is not the ultimate object of NLP. It has interplays with parsing, semantic role labeling etc. They can be integrated together if WSD can be fulfilled by globally modeling word senses. There are some methods like re-ranking and approximate parameter estimation that have the potential to solve that integrating.

Table 3 shows the comparisons between our system and (1) base-line system, (2) Wong and Yang's method. Both sequential CRF and tree CRF perform better than base-line system and Wong and Yang's method. The syntactic feature enhances performance significantly. Sequential CRF enhances little over base line partly because sense transitions in sequences are not easy to be captured. The performance of Wong and Yang's method is below base-line system, which maybe due to its simple features and sparse representation of all possible paths. That is, even POS features are not included in MXPOST and no syntactic features are used. Moreover, in MXPOST, only features that appear more than 4 times in the training corpus are included in the feature set. But candidate labels are all sememes. No special measures like DefSpace and ExDefSpace are taken. Then the feature set is very sparse to represent all possible paths even we duplicate training corpus 3 times.

	our		Wong	Base-line
	seq	tree		
Sense Acc.	79.9	85.9	77.1	79.0

Table 3. Comparisons between our system and Wong and Yang's method and base-line system.

5 Conclusion

In this paper, we probe into sense transition by decomposing sense into sememes. Then sense transition can be replaced by sememe transitions.

We model sememe transitions by MLCRF. For each layer sememe, sequential CRF and tree CRF are used. Multi-layered Viterbi algorithm is proposed at the predicting phase. Experiments show that MLCRF performs better than a base-line system and a max entropy model. Syntactic and hypernym features can enhance the performance significantly.

Acknowledgments

This work was supported by the Natural Sciences Foundation of China under grant No. 60372016, 60673042, and the Natural Science Foundation of Beijing under grant No. 4052027. Thanks for the helpful advices of Professor Zhendong Dong, the founder of HowNet.

References

[Cohn and Blunsom, 2005] T. Cohn and P. Blunsom. Semantic Role Labeling with Tree Conditional Random Fields. In *Proceedings of Ninth Computational Natural Language Learning.*

[Cowie *et al.*, 1992] J Cowie, J. Guthrie and L. Guthrie. Lexical Disambiguation using Simulated Annealing. In *Proceedings of 8th International Conference on Computational Linguistics.*

[Decadt *et al.*, 2004] B. Decadt, V. Hoste, W. Daelemans and A. Bosch. GAMBL, Genetic Algorithm Optimization of Memory-Based WSD. In *Proceedings of ACL/SIGLEX Senseval-3*, Barcelona, Spain, July.

[Ide *et al.*, 1998] N. Ide, Jean V. Word Sense Disambiguation: The state of the art. *Computational Linguistics.*

[Lafferty *et al.*, 2001] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of 18th ICML.*

[McCallum, 2003] A. McCallum. Efficiently inducing features of conditional ramdom fields. In *Proceedings of Conference on Uncertainty in Artificial Intelligence (UAI).*

[Mihalcea, 2005] R. Mihalcea. Unsupervised large-vocabulary word sense disambiguation with graph-based algorithms for sequence data labeling. In *Proceedings of the Joint Conference on Human Language Technology / Empirical Methods in Natural Language Processing* (EMNLP), Vancouver, October.

[Mihalcea and Csomai, 2005] R. Mihalcea, A. Csomai. Sense learner: word sense disambiguation for all words in unrestricted text. In *Proceedings of the 43th Annual Meeting of the Association for Computational Linguistics* (ACL)*, companion volume*, Ann Arbor, MI, June.

[Pearl, 1988] J. Pearl. Probabilistic reasoning in intelligent systems: Networks of plausible inference. *Morgan Kaufmann.*

[Sha and Pereira, 2003] F. Sha, F. Pereira, Shallow parsing with conditional random fields. In *Proceedings of the Human Language Technology, NAACL Conference.*

[Sutton *et al.*, 2004] C. Sutton, K. Rohanimanesh, A. McCallum. Dynamic conditional random fields: factorized probabilistic models for labeling and segmenting sequence data. In *Proceedings of 21st ICML.*

[Wong *et al.*, 2002] Ping-Wai Wong and Yongsheng Yang. A Maximum Entropy Approach to HowNet-Based Chinese Word Sense Disambiguation. In *Proceedings of SemaNet'02.*

[Yuret, 2004] D. Yuret. Some experiments with a naive bayes wsd system. In *Proceedings of ACL/SIGLEX Senseval-3*, Barcelona, Spain, July 2004.

[Zhendong Dong *et*, 2006] Zhendong Dong, Qiang Dong. HowNet and the Computation of Meaning. *World Scientific.*