Identifying Patterns in Texts

Minhua Huang  
Computer Science, Graduate Center  
City University of New York  
New York, NY 10016  
mhuang@gc.cuny.edu

Robert M. Haralick  
Computer Science, Graduate Center  
City University of New York  
New York, NY 10016  
haralick@aim.com

Abstract

We discuss a probabilistic graphical model for recognizing patterns in texts. It is derived from the probability function for a sequence of categories given a sequence of symbols under two reasonable conditional independence assumptions and represented by a product of combinations of conditional and marginal probability functions. The novelty of our model is that it has a mathematical representation which is completely different from existing graphical models such as CRFs, HMMs, and MEMMs. Moreover, it can be used for identifying various patterns in texts. Up to now, we have used this model for recognizing NP chunks and senses of a polysemous word in sentences. This model has achieved very promising results on standard data sets. In the future, we will use this model for extracting semantic roles in a sentence.

1 Introduction

NLP researchers put their efforts on developing methods for extracting patterns in texts. These patterns can be viewed as syntactic patterns or semantic patterns. For example, NP chunks (noun phrases) are syntactic patterns because they are defined by grammatical rules while senses of a polysemous word are semantic patterns because they can be identified by the contexts of the word. Here, we discuss a probabilistic graphical model for recognizing these patterns. The mathematical representation of our model is: $p(c_1, ..., c_N | s_1, ..., s_N) = \prod_{n=1}^{N} p(s_n | c_n) p(c_n)$.

For identifying these patterns in texts, a sentence is partitioned into several phrases. In the case of the NP chunking problem, these phrases can be NP chunks and OTHERs. In the case of the word sense disambiguation (WSD) problem, each phrase is represented by its last word, called the head word. The context of a polysemous word is represented by a sequence of words. These words may be the last words of phrases or words within $\pm N$ words around the polysemous words. By the model, we automatically assign the category to each word of the sequence having highest probability. We determine NP chunks by grouping consecutive words with the same particular category. We determine the category of the ambiguous word by selecting the most frequent category assigned to that word in the sequence. We may determine a semantic role by first, grouping consecutive words with the same particular category, then categorizing these roles into different classes by designing a set of rules based on the Levin’s verb classes.

We test our model for identifying NP chunks with two data sets: the WSJ data set from the Penn Treebank and the CoNLL-2000 shared task data set. Our method achieves an average precision 91.38% and an average recall 91.08% for identifying the project sense of the word line; an average of precision 91.36% and an average of recall 90.07% for identifying the supply_with_food sense of the word serve; an average of precision 86.50% and an average of recall 91.43% for distinguishing the difficult sense of the word hard; an average of precision 89.50% and an average of recall 91.78% for identifying the money_paid_for_the_use_of_money sense of the word interest. We are going to test our model for semantic role labeling (SRL) on the CoNLL-2005 shared task data set.

The rest of our discussion is structured in the following way. The second section presents the method. The third section demonstrates the empirical results. The fourth sec-
tion reviews related researches and discussions. The fifth section gives a conclusion.

2 The Proposed Method

2.1 An Example

Table 1 shows the input sentence “He had deposited his paycheck to the local PNC bank last Saturday morning.” with its POS tags. By the method, each word of the sentence is assigned to one of three different categories $C_1, C_2, \text{and } C_3$. $C_1$ represents a word inside a block (a block can be a NP chunk or a semantic role), $C_2$ represents a word outside a block, $C_3$ represents a word starting a new block. NP chunks or semantic roles are formed by grouping successive words with the same category $C_1$ or starting with the category $C_3$ and followed by zero, one, or more consecutive $C_3$s. The context of the polysemous word bank is found by grouping the words corresponding to the last $C_1$ of consecutive $C_3$s or last $C_2$ of consecutive $C_2$s. Moreover, different semantic roles $A_0, A_1, A_2, A_3$ are needed to be separated from the semantic roles obtained from the previous step.

Table 1. An example of recognizing text patterns

<table>
<thead>
<tr>
<th>Lexicon</th>
<th>POS tag</th>
<th>NP</th>
<th>Class</th>
<th>WSD</th>
<th>SRL</th>
<th>NP chunks</th>
<th>WSD chunks</th>
</tr>
</thead>
<tbody>
<tr>
<td>He</td>
<td>NNP</td>
<td>$C_1$</td>
<td>$C_1$</td>
<td>$c_1$</td>
<td>$c_1$</td>
<td>$c_1$</td>
<td>$A_1$</td>
</tr>
<tr>
<td>had</td>
<td>VBD</td>
<td>$C_2$</td>
<td>$C_2$</td>
<td>$c_2$</td>
<td>$c_2$</td>
<td>$c_2$</td>
<td>$A_2$</td>
</tr>
<tr>
<td>deposited</td>
<td>VBN</td>
<td>$C_2$</td>
<td>$C_2$</td>
<td>$c_2$</td>
<td>$c_2$</td>
<td>$c_2$</td>
<td>$A_3$</td>
</tr>
<tr>
<td>his</td>
<td>PRPS</td>
<td>$C_1$</td>
<td>$C_1$</td>
<td>$c_1$</td>
<td>$c_1$</td>
<td>$c_1$</td>
<td>$A_1$</td>
</tr>
<tr>
<td>paycheck</td>
<td>NNS</td>
<td>$C_1$</td>
<td>$C_1$</td>
<td>$c_1$</td>
<td>$c_1$</td>
<td>$c_1$</td>
<td>$A_2$</td>
</tr>
<tr>
<td>the</td>
<td>DT</td>
<td>$C_1$</td>
<td>$C_1$</td>
<td>$c_1$</td>
<td>$c_1$</td>
<td>$c_1$</td>
<td>$A_3$</td>
</tr>
<tr>
<td>local</td>
<td>JJ</td>
<td>$C_1$</td>
<td>$C_1$</td>
<td>$c_1$</td>
<td>$c_1$</td>
<td>$c_1$</td>
<td>$A_3$</td>
</tr>
<tr>
<td>PNC</td>
<td>NN</td>
<td>$C_1$</td>
<td>$C_1$</td>
<td>$c_1$</td>
<td>$c_1$</td>
<td>$c_1$</td>
<td>$A_3$</td>
</tr>
<tr>
<td>bank</td>
<td>NNS</td>
<td>$C_1$</td>
<td>$C_1$</td>
<td>$c_1$</td>
<td>$c_1$</td>
<td>$c_1$</td>
<td>$A_3$</td>
</tr>
<tr>
<td>last</td>
<td>JJ</td>
<td>$C_1$</td>
<td>$C_1$</td>
<td>$c_1$</td>
<td>$c_1$</td>
<td>$c_1$</td>
<td>$A_3$</td>
</tr>
<tr>
<td>morning</td>
<td>NN</td>
<td>$C_1$</td>
<td>$C_1$</td>
<td>$c_1$</td>
<td>$c_1$</td>
<td>$c_1$</td>
<td>$A_3$</td>
</tr>
</tbody>
</table>

In the case of NP chunking, $C_1$ represents a symbol inside a NP chunk, $C_2$ represents a symbol outside of a NP chunk, and $C_3$ represents a symbol starting a new NP chunk. In the case of word sense disambiguation, $C_1, C_2, \text{and } C_3$ are followed the conventions of NP chunking. In the case of semantic role labeling, $C_1$ represents a symbol inside a semantic role, $V$ represents the main verb, $C_2$ represents a symbol outside of a semantic role, and $C_3$ starts a new semantic role.

2.2 Describing the Task

Let $L$ be a language, $V$ be a vocabulary of $L$, and $T$ be $POS$ tags of $V$. Let $S$ be a sequence of symbols associated with a sentence, $S = (s_1, ..., s_N)$, where $s_n = \langle w_n, t_n \rangle$, $w_n \in V$, $t_n \in T$. Let $C$ be a set of categories, $C = \{C_1, C_2, C_3\}$. Let $B$ be a block. The definition of $B$ can be found in the section 2.5. $C_1$ indicates the current symbol is in $B$, $C_2$ indicates the current symbol is not in $B$, and $C_3$ starts a new $B$. The tasks can be stated as, given $S$, we need to find:

1. a sequence of categories, $(c_1, ..., c_N), c_i \in C$, with the best description of $S$;
2. all the $B$s based on $(c_1, ..., c_N)$, s.t. $B = \{B_1, \ldots, B_M\}$, s.t. $B_i \cap B_j = \phi$ and $B_i \subset S$

2.3 Building Graphical Models

Given $S = (s_1, s_2, ..., s_N)$, $C = \{C_1, C_2, C_3\}$, for $s_i \in S$, we want to find $c_i \in C$, s.t.

$$
(c_1, c_2, ..., c_N) = \arg\max_{c_1, c_2, ..., c_N} p(c_1, c_2, ..., c_N|S)
$$

Suppose $c_i$ is independent of $c_{j \neq i}$ given $(s_1, s_2, ..., s_N)$. This means that the symbol sequence contains all the information with respect to the category class associated with any word. Moreover, assume $c_i$ is independent of $(s_1, ..., s_{i-2}, s_{i+2}, ..., s_N)$ given $(s_{i-1}, s_i, s_{i+1})$. This means that all the information pertaining to the category class of the word $i$ is in entities contained by the symbol associated with the word $i$, its predecessor word $i - 1$, and its successor word $i + 1$. The probability graphical model using these assumptions is shown in Fig 1. From this model, a set of
$2N - 2$ cliques\footnote{A clique is a maximal complete set of nodes.} is obtained:

$$CIL = \{\{s_1, s_2, c_1\}, \{s_1, s_2, c_2\}, \{s_2, s_3, c_2\}, \ldots, \{s_{N-1}, s_N, c_{N-1}\}, \{s_{N-1}, s_N, c_N\}\}$$

Moreover, there is a corresponding set of $2N - 3$ separators $^2$:

$$SEP = \{\{s_1, s_2\}, \ldots, \{s_{N-1}, s_N\}, \{s_2, s_3\}, \ldots, \{s_{N-1}, c_{N-1}\}\}$$

The junction tree\footnote{A junction tree is a maximum spanning tree w.r.t separator size. A maximum spanning tree is a tree over $V$ whose sum of edge weights has a maximum value. Here the edge weights are the sizes of the separators.} is formed as shown in Fig 2. The cliques are represented as nodes and separators are represented as edges. From this model, according to [2],

\begin{equation}
p(c_1, \ldots, c_N | s_1, \ldots, s_N) = M_S \prod_{n=1}^{N} p(s_{n-1} | s_n, c_n)p(s_{n+1} | s_n, c_n)p(s_n | c_n)p(c_n)\end{equation} \tag{2}

In (2), we define $p(s_0 | s_1, c_1) = p(s_{N+1} | s_N, c_N) = 1$ and $M_{s_1, \ldots, s_M}$ is a constant depending only on $s_1, \ldots, s_M$ and not depending on any $c_n$. It can be obtained by the equation (3).

\begin{equation}M_S = \frac{1}{p(s_1, s_2)p(s_2, s_3)\ldots p(s_{N-1}, s_N)}\end{equation} \tag{3}

2.4 Making Decisions

Because each $c_n$ is independent of each other in (2), its value can be determined individually. We can find $c_n$ by

$$c_n = \arg\max_{c_n} p(s_{n-1} | s_n, c_n)p(s_{n+1} | s_n, c_n)p(s_n | c_n)p(c_n)$$

Then, (1) can be rewritten as:

\begin{equation}\begin{aligned}
(c_1, c_2, \ldots, c_N) = (\arg\max_{c_1} p(s_2 | s_1, c_1)p(s_1 | c_1)p(c_1)), \\
\arg\max_{c_2} (p(s_1 | s_2, c_2)p(s_3 | s_2, c_2)p(s_2 | c_2)), \\
\ldots, \\
\arg\max_{c_{N-1}} (p(s_{N-2} | s_{N-1}, c_{N-1})p(s_{N-1} | s_{N-1}, c_{N-1})) \\
\arg\max_{c_N} (p(s_{N-1} | s_N, c_N)p(s_N | c_N)p(c_N)) \end{aligned}\end{equation} \tag{4}

2.5 Determining NP Chunks or Semantic Roles

Given $(<s_1, c_1>, \ldots, <s_N, c_N>)$, $s_i \in S$, $c_i \in C$, $S, C$ are defined in the section 2.2. Let $B$ be a block if and only if:

1. for some $i < j$, $B = (<s_i, c_i>, <s_{i+1}, c_{i+1}>$ $\ldots, <s_j, c_j>)$
2. $c_i \in \{C_1, C_3\}$
3. $c_n = C_1$, $n = i + 1, \ldots, j$
4. $B' \subseteq B$ and $B'$ satisfies (1), (2), and (3) $\Rightarrow B' = B$

2.6 Determining the Sense of a Polysemous Word

Given $(<s_1, c_1>, \ldots, <s_N, c_N>)$, $s_i \in S$, $c_i \in C$, $S, C$ are defined in the section 2.2. We assign the class $C_k$ for the polysemous word $w_i$ if and only if:

\begin{equation}\begin{aligned}
\# \{n \in \{1, \ldots, N\} | c_n = C_k\} \\
> \# \{m \in \{1, \ldots, N\} | c_m = C_j\}, i \neq k
\end{aligned}\end{equation} \tag{5}

2.7 Estimating Probabilities of the Model

We use a training set to estimate the probabilities for the equation (2). In our model, the probability of a current symbol being assigned to the class $c$ in a sequence associated with a sentence is partially dependent on the probability of the previous symbol given the current symbol and the class $c$, and the probability of the success symbol given the current symbol and the class $c$. In this way, the adjacency between two neighboring symbols of an incoming sentence are preserved by the overlapping of our model. Therefore, we can consider a group of $k$ sentences for estimating the
probabilities. Our training set has K sentences. Each sentence \( k \) consists of \( N_k \) words, \( < x_{k,1}, \ldots, x_{k,N_k} > \), and the corresponding class labels \( < y_{k,1}, \ldots, y_{k,N_k} > \). Hence the training set is \( \Psi = \{ \psi_1, \psi_2, \ldots, \psi_K \} \), where \( \psi_k = ( < x_{k,1}, y_{k,1} >, \ldots, < x_{k,N_k}, y_{k,N_k} > ) \). Let \( l, w, z, c \) be random variables, where \( w \) designates a word, \( t \) designates a word before \( w \), \( z \) designates a word after \( w \), and \( e \) designates a class. Let \( p_{w|e}(c|\gamma) \) designate the conditional probability of a word being \( \alpha \) given that its class is \( \gamma \). Let \( p_{t|w,e}(c|\beta, \gamma) \) designate the conditional probability of the word previous to the current word being \( \beta \) and its class is \( \gamma \). Let \( p_{z|w,e}(c|\beta, \gamma) \) designate the conditional probability of the word after the current word being \( \alpha \) given that the current word is \( \beta \) and its class is \( \gamma \).

Let:

\[
I = \{(k, n)|k = 1, \ldots, K, n = 1, \ldots, N_k\}
\]

\[
p_{w|e}(c|\gamma) \text{ can be estimated by:}
\]

\[
\hat{p}_{w|e}(c|\gamma) = \frac{\#\{(k, n) \in I|\alpha = x_{k,n}, \gamma = y_{k,n}\}}{\#\{(k, n) \in I|\gamma = y_{k,n}\}} (6)
\]

\[
p_{t|w,e}(c|\beta, \gamma) \text{ can be estimated by:}
\]

\[
\hat{p}_{t|w,e}(c|\beta, \gamma) = \frac{\#\{(k, n) \in I|\alpha = x_{k,n-1}, \beta = x_{k,n}, \gamma = y_{k,n}\}}{\#\{(k, n) \in I|\beta = x_{k,n}, \gamma = y_{k,n}\}} (7)
\]

\[
p_{z|w,e}(c|\beta, \gamma) \text{ can be estimated by:}
\]

\[
\hat{p}_{z|w,e}(c|\beta, \gamma) = \frac{\#\{(k, n) \in I|\alpha = x_{k,n+1}, \beta = x_{k,n}, \gamma = y_{k,n}\}}{\#\{(k, n) \in I|\beta = x_{k,n}, \gamma = y_{k,n}\}} (8)
\]

3 Empirical Results

We define features based on the equation (2) as follows:

\[
f(s_i, c_i) = p(c_i)q(s_i|c_i)r(s_{i-1}|s_i, c_i)I(s_{i+1}|s_i, c_i) (9)
\]

We form the training set by including 90% instances and the testing set by including 10% instances of the whole data set. In this way, we do it iteratively 10 times to select the different training sets and testing sets. The evaluation metric we have used are precision \( P_{re} \), recall \( R_{ec} \), and f-measure \( F_{me} \).

\[
F_{me} = \frac{2P_{re}R_{ec}}{P_{re}+R_{ec}}
\]

3.1 Identifying NP Chunks Using CoNLL-2000 Shared Task Data Set

We have conducted three different tests on the CoNLL-2000 shared task data set by choosing different values of features. From these selections, we examine the probabilities in order to find the one which contributes the best performance. In the first test, we include all lexicon and POS tags from the data set. We have tested our model according to descriptions in the section 3. We averaged the results that we received. The average precision is 95.15%, the average recall is 96.05%, and the average f-measure is 95.59%. In the second test, the lexicon is excluded. We only include the POS tags. In the third test, all POS tags are excluded and only the lexicon is included. The results are shown in the table 2. By comparing the three results on the CoNLL-2000 shared task data, we have noticed that if the model is built only on the lexical information, it has the lowest performance of f-measure 89.75%. The model’s performance improved 3% in f-measure if it is constructed by POS tags. The model achieves the best performance of 95.59% in f-measure if we are considering both lexicons and POS tags.

<table>
<thead>
<tr>
<th>Measurement</th>
<th>Lexicon + POS tags</th>
<th>POS tags</th>
<th>Lexicon</th>
</tr>
</thead>
<tbody>
<tr>
<td>( P_{re} )</td>
<td>95.15</td>
<td>92.27</td>
<td>86.42</td>
</tr>
<tr>
<td>( R_{ec} )</td>
<td>96.05</td>
<td>93.76</td>
<td>93.35</td>
</tr>
<tr>
<td>( F_{me} )</td>
<td>95.59</td>
<td>92.76</td>
<td>89.75</td>
</tr>
</tbody>
</table>

The best performances have achieved on the feature values containing lexicon + POS tags. In this case, the average f-measure is 95.59%.

3.2 Identifying NP Chunks Using WSJ Data Set from Penn Treebank

The second data set on which we have experimented is the WSJ data of Penn Treebank. The main reason for us to use this data set is that we want to see whether the performance of our model can be improved when it is built on more data. We build our model on a training set which is seven times larger than the CoNLL-2000 shared task training data set (Section 3.3). The performance of our method for the data is listed in the table 3. The average precision is increased 2.7% from 95.15% to 97.73%. The average recall is increased 2.8% from 96.05% to 98.65%. The average f-measure is increased 2.7% from 95.59% to 98.2%.

3.3 Identifying Sense of Polysemous Words Using line_interest_hard_serve Data Sets

We test our model for WSD on the data sets line, hard, serve, and interest. The senses’ descriptions and in-
Table 3. The test results on the WSJ data from the Penn Treebank

<table>
<thead>
<tr>
<th>Training</th>
<th>Testing</th>
<th>( P_{re} )</th>
<th>( R_{re} )</th>
<th>( F_{me} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>800 files</td>
<td>100 files</td>
<td>0.9806</td>
<td>0.9838</td>
<td>0.9822</td>
</tr>
<tr>
<td>200-999</td>
<td>1100-1199</td>
<td>0.9759</td>
<td>0.9868</td>
<td>0.9814</td>
</tr>
<tr>
<td>1200-1299</td>
<td>1300-1399</td>
<td>0.9794</td>
<td>0.9863</td>
<td>0.9828</td>
</tr>
<tr>
<td>1400-1499</td>
<td>1500-1599</td>
<td>0.9771</td>
<td>0.9868</td>
<td>0.9871</td>
</tr>
<tr>
<td>1600-1699</td>
<td>1700-1799</td>
<td>0.9771</td>
<td>0.9848</td>
<td>0.9809</td>
</tr>
<tr>
<td>1800-1899</td>
<td>1900-1999</td>
<td>0.9774</td>
<td>0.9863</td>
<td>0.9819</td>
</tr>
<tr>
<td>2000-2099</td>
<td>( \mu )</td>
<td>0.9773</td>
<td>0.9865</td>
<td>0.9818</td>
</tr>
<tr>
<td></td>
<td>( \sigma )</td>
<td>0.0019</td>
<td>0.0014</td>
<td>0.0008</td>
</tr>
</tbody>
</table>

The recall, precision, and f-measure obtained for each test of 100 files. The average recall, precision, and f-measure and their standard deviations are obtained from 1000 testing files.

stances’ distributions can be found in [11] and [3]. Because of the limitations of number of instances (a sentence having the polysemous word in it) for each sense in the corpora, we select the first three senses for each polysemous word in our test. Again, the values of features are made by lexicon + POS tags. We test our model based on descriptions of the section 3. The test results are shown in the table 4. We have noticed that, with the same instances for a polysemous noun, adjective, or verb, our model achieves the best f measure for polysemous nouns and the worst result for polysemous adjectives. For example, the average \( f_{me} > 91 \% \) if the number of instances > 1270 for the polysemous nouns line and interest. However, in order to keep the same f-measure value, the polysemous word serve needs to have more instance: 1841 instances. The polysemous adjective needs to have about 3350 instances to reach the average \( f_{me} = 88.76\% \). Moreover, in any case, our model achieves an average \( f_{ma} = 80\% \) if the number of instances is reduced to about 400. We conclude that the performance of our model on WSD is dependent on the number of instances in the training set: the larger the better.

4 Related Research and Discussion

Currently existing graphical models for NLP are HMMs [13] [16], MEMMs [13], and CRFs [9] [18]. These models are built under different conditional independence assumptions for obtaining the sequence \( < c_1, \ldots, c_N > \) that maximizes \( p(c_1, \ldots, c_N, s_1, \ldots, s_N) \) or \( p(c_1, \ldots, c_N | s_1, \ldots, s_N) \). Among these models, HMMs and MEMMs are directed graphic models while CRFs and our model are undirected graphical models. Comparing with these two undirected graphic models, each \( c_i \) links to \( c_{i-1} \) and \( s_i \) in CRFs. Therefore, \( c_i \) is dependent on \( c_{i-1} \) and \( s_i \). In contrast, in our model, each \( c_i \) links to \( s_i \), \( s_{i-1} \), and \( s_{i+1} \), not the previous category \( c_{i-1} \). Therefore, \( c_i \) is not dependent on \( c_{i-1} \). This makes it possible for the sequence \( < c_1, \ldots, c_N > \) with the maximum value of \( p(c_1, \ldots, c_N | s_1, \ldots, s_N) \) to be determined by finding each \( c_i \) that satisfies the equation (4). In this way, the time complexity for recognizing a new incoming sequence with \( N \) symbols at the worst case is \( M * N \), where \( M \) is the number of categories. For example, if \( C = \{C_1, C_2, C_3\} \), then \( M = 3 \). Therefore, the time complexity is \( O(N) \). The memory space also will be reduced compared the other graphic models because we don’t need to store all the previous category chains into the memory. Moreover, our model is more reliable and stable due to the global maximum probability being obtained by the local maximal probabilities. There is no a chance to change the previous category chain because of an accidently higher probability at the current state.

Table 4. The results on line, hard, serve, interest data

<table>
<thead>
<tr>
<th>Word</th>
<th>Description</th>
<th># of Instance</th>
<th>( f_{me} ) %</th>
</tr>
</thead>
<tbody>
<tr>
<td>line</td>
<td>project</td>
<td>2218</td>
<td>92.24</td>
</tr>
<tr>
<td>noun</td>
<td>phone</td>
<td>429</td>
<td>85.22</td>
</tr>
<tr>
<td>hard</td>
<td>difficult</td>
<td>3345</td>
<td>88.76</td>
</tr>
<tr>
<td>adj</td>
<td>not soft</td>
<td>502</td>
<td>83.75</td>
</tr>
<tr>
<td></td>
<td>physical not soft</td>
<td>376</td>
<td>80.05</td>
</tr>
<tr>
<td>serve</td>
<td>supply with food</td>
<td>1841</td>
<td>91.04</td>
</tr>
<tr>
<td>verb</td>
<td>hold an office</td>
<td>1272</td>
<td>87.48</td>
</tr>
<tr>
<td>function as something</td>
<td>853</td>
<td>82.52</td>
<td></td>
</tr>
<tr>
<td>interest</td>
<td>money paid for the use of money</td>
<td>1272</td>
<td>91.45</td>
</tr>
<tr>
<td>noun</td>
<td>a company</td>
<td>500</td>
<td>88.85</td>
</tr>
<tr>
<td></td>
<td>readiness to give attention</td>
<td>361</td>
<td>79.95</td>
</tr>
</tbody>
</table>
contrast with other methods, the polysemous word is represented by a sequences of ordered words with POS tags. Our model assigns a category for a word of the input sentence based on the information of the word, the previous word, and the next word we have met before, which a human often does this in the same way. The experiments in the section 3.1 show our model achieves better performance than HMMs and CRFs [18]. For WSD, our method can achieve precision and recall > 90 if number of instances of a sense \( \geq 1000 \).

5 Conclusions

Recognizing patterns in a sentence is the first step toward understanding the meaning of the sentence. This paper presents a new probabilistic graphical model for doing such tasks. Experiments show that our model is effective. We have achieved an average of precision 97.7\% and an average of recall 98.7\% on WSJ data from the Penn Treebank and an average precision 95.15\% and an average recall 96.05\% on CoNLL-2000 shared task data set for recognizing NP chunks in a sentence. Moreover, we have achieved an average precision 90.57\% and an average of recall 92.35\% on recognizing a particular sense of polysemous nouns, an average precision 90.86\% and an average recall 91.22\% on recognizing a particular sense of a polysemous verb, and an average precision 86.50\% and an average recall 91.01\% on recognizing a particular sense of a polysemous adjective. From the empirical results, in order to improve the performance of WSD, we need to increase the size of the training set. In the future, we will expend number of instances for line_interest_hard_ser data sets and test our model for other senses of these polysemous words. Moreover, we will test our model on recognizing the semantic roles in a sentence by using CoNLL-2005 shared task data set.

References