# The Improving Techniques for Disambiguating Non-alphabet Sense Categories

Feng-Long Hwang, Ming-Shing Yu, Min-Jer Wu

Department of Applied Mathematics, National Chung-Hsing University

Page 67 ~ 86

Proceedings of Research on Computational Linguistics

Conference XIII (ROCLING XIII)

Taipei, Taiwan

2000-08-24/2000-08-25

# The Improving Techniques
# for Disambiguating Non-alphabet Sense Categories

**Feng-Long Hwang**[Ψ], **Ming-Shing Yu, Min-Jer Wu**
Department of Applied Mathematics, National Chung-Hsing University,
Taichung 40227, Taiwan,
flhwang@mail.lctc.edu.tw, msyu@dragon.nchu.edu.tw

**ABSTRACT**

Usually, there are various non-alphabet symbols ("/", "**:**", "**-**", etc.) occurring in Mandarin texts. Such symbols may be pronounced more than one oral expression with respect to its sense category. In our previous works, we proposed the multi-layer decision classifier to disambiguate the sense category of non-alphabet symbols; the elementary feature is the statistical probability of token adopting the Bayesian rule. This paper adopts more features of tokens in sentences. Three techniques are further proposed to improve the performance. Experiments show that the proposed techniques can disambiguate the sense category of target symbols quite well, even with small size of data. The precision rates for inside and outside tests are upgraded to 99.6% and 96.5% by using more features of token and techniques.

***Key Words*****:** *Multi-layer decision classifier*, *Bayesian rule*, *word* sense *disambiguation, voting scheme, pattern table.*

## 1. Introduction

Various homographs or non-alphabet symbols in the Mandarin (but not limited to) occur frequently. The patterns containing these symbols may be pronounced with respect to its semantic sense. The **non-alphabet symbols** are defined: the symbols which are not the Mandarin characters (字) and may be pronounced different oral expressions. We call such phenomenon *oral ambiguity*.

The purpose of word sense disambiguation (WSD) is to identify the most possible category among candidate's sense category. It is important to disambiguate the word sense automatically for the natural language processing (NLP). Many works [Brown etc., 1991], [Fujii and Inue,1998] and [Ide and Veronis,1998], addressed WSD problems in the past.

In our previous works [Hwang, etc., 1999a; Hwang, etc., 1999b], we proposed the

[Ψ] Correspondence author.

multi-layer decision classifier (MLDC) to predict the sense category, in which the voting scheme is used to predict the final category. Even though the domains of sense in the paper just focus on three non-alphabet symbols, the proposed approach can be extended into other symbols in Mandarin and related ambiguity problems. The features of token and improving techniques described in this paper will be employed in the $2^{nd}$ layer classifier. The main domain will focus on the improvements for the $2^{nd}$ layer decision classifier. The model of our previous works is regarded as the baseline system. Comparing with the *baseline* model, the proposed features of token and techniques in this paper improve the performance of inside test from 97.8 to 99.6% and outside test from 93.0 to 96.6%.

The paper is organized as follows: related information and previous works will be described first. Section 3 elaborates the principal techniques for $2^{nd}$ layer classifier in MLDC. Section 4 focuses on the evaluation for empirical features. Some improving techniques are proposed in section 5. The conclusions are presented in last Section.

## 2. Description of Related Works

In this Section, we first describe the applications of word sense disambiguation. The precious literatures on WSD and several methods, which are used to disambiguate the sense categories and classification problems of ambiguity, will be introduced next. Finally we will illustrate our previous approach.

### 2.1 Applications of Word Sense Disambiguation

The applications of WSD in natural language processing include the following domains:

- **Content and thematic analysis**
  Analyzing the distribution of pre-defined categories of words in text.
- **Information retrieval and extraction**
  When querying information, in a standalone system or Internet environment, the system should identify the real meaning for the query; excluding unnecessary data then correctly return desirable information among heterogeneous data.
- **Machine translation**
  We can first disambiguate the word sense categories, and then translate the word into correct semantic meanings associated with the target word.
- **Speech processing**
  Within the text analysis phase of TTS synthesis, the sense ambiguity of non-alphabet symbols or homographs should be resolved. The patterns containing such symbols can be translated into their oral expressions. The problem dealt with in our paper is very important for the precise speech output of TTS system.

## 2.2 Related Works

A lot of literatures have been published on word sense disambiguation in the past. They range from dictionary-based to corpus-based approaches. The former is dependent on the definitions of machine readable dictionary (MRD) [Veronis, etc., 1990] while the later usually rely only on the frequency of word extracted from the text corpus to construct the feature database [Schutze, etc.,1995]. Corpus-based approach adopts the co-occurrence of words which are extracted from the large text corpora to construct the feature database [Leacock, 1993] and provides the advantage of being generally applicable to new text, domains and corpus without the costly, error-prone parsing and semantic analysis. However, corpus-based approach also has some weakness: the corpus is always hard to collect and is time-consuming. The situation is so called “knowledge acquisition bottleneck” [Gale etc., 1992].

Based on the type of context in examples, the classifiers for word sense category use two contextual information: *local* and *topical context*. Hearst, etc. [1999] use local context with a narrow syntactic parse, in which the context is segmented into noun phrases, verb groups and other groups. Gale etc.[1992] developed a topical classifier, in which the Bayesian rule is used and the only information adopted is the co-occurrence of unordered word.

With respect to the contextual information, lexical information is formalized form of information involved in each surrounding word. Lee etc. [1997] adopt the discrimination score, based on maximum entropy of surrounding words in a sentence, to discriminate the word sense. Its precision rate is 80 % average.

Yarowsky [1994 and 1997] build a classifier using the local context cues within ± k windows for target word. A log-likelihood ratio is generated, which stands for the strength of each clue of local context. The decision will be made for matching sorted ratio sequence to decide the sense category of target word. The average performance ranges from 96% to 97% while the domain size of sense is only 2 for all ambiguous questions.

## 2.3 Our Previous Works

In contrast to *2*-gram, *3*-gram and *n*-gram language models, our previous paper [Hwang, etc., 1999a, 1999b] proposed an approach of multi-layer decision classifiers, which can resolve the category ambiguity of oral expression for non-alphabet symbols. A two-layer classifier has been developed. The first layer decision classifier can be viewed as decision tree based on the linguistic knowledge. Some impossible categories will be excluded while the remaining categories are all the possible categories. The second classifier employs a voting scheme to predict the final category with maximum probability score. The precision rates for inside and outside testing are 97.8% and 93.0% average.

# 3. The Principal Techniques

At first, the data set and sense categories for three target symbols are described. In 2nd decision classifier, a voting scheme, derived from Bayesian rule, is used to predict the portable sense category with maximum score.

## 3.1 Elementary Information of Data Set

The original data set is collected through different source, including: Academic Sinica Balance Corpus (ASBC), text files downloaded from Internet. ASBC is composed of 316 text files which contain 5.22M characters in Mandarin, English and other symbols totally [Huang, 1995; CKIP, 1995]. Only the sentence with such non-alphabet symbols will be extracted and appended into the empirical data set. Examples of three non-alphabet symbols *slash* (/), *colon* (**:**) and *dash* (**-**) are extracted and appended into our empirical data set. The sentence size of three non-alphabet symbols is 1115,1282 and 1685 respectively. The ratio of training and testing set is 4:1 appropriately. These sentences will be classified into different sense category with respect to target symbols. The sense categories and their oral expressions are listed in Tables 1-3. Less frequent (less than 1%) sense categories will be neglected.

Word segmentation paradigm is based on the Academia Sinica Chinese Electronic Dictionary (ASCED), which contains about 78,000 words. The words in ASCED are composed of one to 10 characters. Our principal rule of segmentation is first subject to maximal length of words and then to least number of words in a segmented pattern sequence. The priority scheme is that the segmented word sequence, which contains a word of maximal length, will be chosen. If two sequences have same maximum length of words, we compare further the total number of words in such sequences; then the sequence that is composed of least number of words will be chosen. The same segmentation's priority will be adopted within the training phase and testing phase.

There are several categories which speech for non-alphabet symbol "/" are silence; the duration for silence in prosodic parameter is still different to other senses. During the synthesis processing in TTS system, the duration with respective to its category will be varied and decided with respect to prosody needed. The numbers of token and sentence for three target symbols in our feature database are listed in Table 4.

Table 1: Seven sense categories and their related oral expressions of the target symbol "/".

| category | lexical patterns with non-alphabet symbol "／" | oral expression in Mandarin | data dis. (%) |
|---|---|---|---|
| 1. date | ３／４(March 4th) | 三月四日 | 15.96 |
| 2. fraction | ３／４(three fourth) | 四分之三 | 8.88 |
| 3. time(music) | ３／４(three four time) | 四分之三拍 | 17.52 |
| 4. path, directory | ／ｄｅｖ／ｎｕｌｌ | 斜線ｄｅｖ斜線ｎｕｌｌ | 25.69 |
| 5, computer words | Ｉ／Ｏ | Silence or 斜線 | 2.04 |
| 6. production version | ＶＡＸ／ＶＭＳ | Silence (longer pause) or 斜線 | 5.52 |
| 7. others | 中／日／韓文(China/Japan/Korea) | Silence (longer pause) | 25.45 |

Table 2: Five sense categories and its related oral expressions of target symbol ":".

| Sense category | lexical patterns with non-alphabet symbol "：" | oral expression in Mandarin | data dis. (%) |
|---|---|---|---|
| 1. punctuation | 優點：經濟省時 | 優點(silence)經濟省時 | 32.64 |
| 2. time | ３：２０PM | 下午三點二十分(three twenty PM) | 11.63 |
| 3. versus | ３：２０ | 三比二十(three versus twenty) | 13.39 |
| 4. telephone | TEL：４２６４８５６ | 電話(silence)４２６４８５６ | 8.50 |
| 5. expression | 教練表示：照常進行 | 教練表示(silence)照常進行 | 33.43 |

Table 3: Seven sense categories and its related oral expressions of target symbol "-".

| Category | lexical patterns with non-alphabet symbol "－" | oral expression in Mandarin | data dis. (%) |
|---|---|---|---|
| 1. figure, address | 圖２－１(Figure 2-1) | 圖２之１ | 7.64 |
| 2. interval | ６－９月份營業收入 | ６至９月份營業收入 | 21.05 |
| 3. production | ｐｃ－ｃｉｌｌｉｏｎ | ｐｃ(silence)ｃｉｌｌｉｏｎ | 17.01 |
| 4. computer term | Ｅ－Ｍａｉｌ | Ｅ(silence)Ｍａｉｌ | 5.91 |
| 5. tel. fax | 電話：４２６－４８５６ | 電話：４２６(silence)４８５６ | 21.91 |
| 6. hyphen | 登記地點－圖書館前 | 登記地點(silence)圖書館前 | 24.22 |
| 7. minus | 公式：Ｘ－２＝２０ | 公式：Ｘ減２等於２０ | 2.23 |

Table 4: numbers of token and sentence for three target symbols.

| $S_n$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | token no. | sentences no. |
|---|---|---|---|---|---|---|---|---|---|
| slash(/) | 2906 | 1325 | 2471 | 3051 | 232 | 821 | 3772 | 14578 | 1115 |
| colon(:) | 4198 | 2564 | 2801 | 1464 | 3963 | 0 | 0 | 15028 | 1282 |
| dash(-) | 1568 | 5083 | 3481 | 1199 | 6004 | 4328 | 445 | 22103 | 1685 |

Table 5 displays several entries of token word "公車" in the feature database. Twelve entries of token "公車" are listed in Table 5, in which each entry is composed of 5 tuples (*w*, *l*, *count*, *s*, *pos*). Tag "*Na*" represents the *common noun*. The number in field *l* represents that the location of token *w* is preceding (negative number) or following (positive number) the target symbol respectively. It is possible that one token maybe occurs in more than two categories. Table 6 represents the tokens occurrence only considering the two location types: $CH_L$ and $CH_R$. Field *l* represents the token's location preceding ($CH_L$) or following ($CH_R$) the non-alphabet symbols neglecting the token order. *p* and *f* in field *l* denote the location preceding and following the non-alphabet symbols. In our experiments, two location schemes will be evaluated in Section 4 and 5.

**Table 5**: the token word “公車” in feature database occurs in sense category 1,3,6

| *w* | *l* | *count* | *s* | *pos* | *w* | *l* | *count* | *s* | *pos* |
|---|---|---|---|---|---|---|---|---|---|
| 公車 | -6 | 1 | 1 | Na | 公車 | +5 | 4 | 3 | Na |
| 公車 | -5 | 2 | 1 | Na | 公車 | -7 | 1 | 6 | Na |
| 公車 | -2 | 1 | 1 | Na | 公車 | -5 | 2 | 6 | Na |
| 公車 | -1 | 3 | 1 | Na | 公車 | -2 | 1 | 6 | Na |
| 公車 | -4 | 2 | 3 | Na | 公車 | -1 | 2 | 6 | Na |
| 公車 | -1 | 1 | 3 | Na | 公車 | +5 | 8 | 6 | Na |

.

**Table 6:** The token “公車” occurs in feature database; without regarding the individual location.

| *W* | *l* | *count* | *s* | *pos* |
|---|---|---|---|---|
| 公車 | p | 7 | 1 | Na |
| 公車 | p | 3 | 3 | Na |
| 公車 | f | 4 | 3 | Na |
| 公車 | f | 5 | 6 | Na |
| 公車 | f | 8 | 6 | Na |

### 3.2 The Structure of MLDC

The function of multiple decision classifiers (MLDC) can be described as follow:

Suppose that *E* denotes the example with non-alphabet symbols, $\Phi_1$and $\Phi_2$ denote the 1st and 2nd classifier respectively. And *possi_set* is the set containing all possible categories induced by 1st classifier. *TScore*( · ) will compute the total score for a given category based on the voting criterion and statistical parameters schemes.

$$\Phi_1(E) = possi\_set, \tag{1}$$

$$\Phi_2(possi\_set) = \arg\max_{s_j \in possi\_set} TScore(s_j) \tag{2}$$

where $s_j$ denotes the sense category for target symbols. *possi_set* contains all the possible sense categories. *TScore*( · ) denotes the function of computing the total score for sense category.

### 3.3 The Statistical Decision Classifier with voting schemes

The segmentation task of testing phase adopts same criterions as that in training phase. A sentence will be divided into $CH_L$ and $CH_R$, which are segmented into one to several basic tokens (Mandarin word or character). For each token in example, the probability of each category can be calculated and summed up based on the evidence (parameters found in feature database) respectively. It is called the ***voting scheme***.

Based on the *voting schemes,* each token in $CH_L$ and $CH_R$ have a statistical probability value, which looks like the voting suffrage, assigned to each category of the non-alphabet symbol. Like the political voting mechanism, the only candidate who gets the tickets in majority (maximum score in our approach) will become to be the predicted one. First the token unit we use is word with the location feature in $CH_L$ or $CH_R$, in which the count of token occurred in same chunk ($CH_L$ or $CH_R$) will be summed up with respect to the sense category. The scheme with character token will be analyzed in Section 4.

The prediction processing is based on the occurrence of each token inside training corpus for each category. The example *E* is composed of word sequence *W* and contains three parts: chunk-L($CH_L$), non-alphabet symbol *TS* (target symbol) and chunk-R($CH_R$). *E,* $CH_L$ and $CH_R$ can be expressed as:

$$E = CH_L + TS + CH_R \tag{3}$$

$$CH_L = w_{-m}w_{-(m-1)}\cdot\ \cdot\ \cdot w_{-j}\cdot\ \cdot\ \cdot \mathrm{w}_{-1} \tag{4}$$

$$CH_R = w_{+1}w_{+2}\cdot\ \cdot\ \cdot w_{+j}\cdot\ \cdot\ \cdot \mathrm{w}_{+n} \tag{5}$$

where *m* and *n* are the total number of tokens in $CH_L$ and $CH_R$.

Let the category $s_{\max}$ be the sense category with maximum conditional probability of sense category *s*, given the word sequence *W*. By the definition of the Bayesian rule, *P(s|W)* can be written as:

$$P(s|W)=\frac{P(s)\bullet P(W|s)}{P(W)} \tag{6}$$

MLDC needs to find the sense category $s_{\max}$ with maximum conditional probability *P*(*s*|*W*). Thus:

$$s_{\max} = \max_s \frac{P(s)\bullet P(W \mid s)}{P(W)} = \frac{p(s)\bullet p(w_1, w_2, \cdots, w_M \mid s)}{p(w_1, w_2, \cdots, w_M)}, \tag{7}$$

where *N* and *M denote* the number of sense category of target symbol and token (word) in word sequence *W*.

Two problems should be considered for the Eq. (7). One is the fact that the probability of p($w_1$,$w_2$,…,$w_n$|*s*) needs large memory and computation for the word sequence *W*. The other is the data sparseness because of the small amount of data set; which usually cause the situation of zero frequency. Each token *w* in word sequence *W*, under our voting scheme of preference scoring, can be regarded independent to other token. For the probability of sense category *s* given a token *w,* the Eq. (7) can be modified as:

$$Score(s \mid W) = \sum_i P(s \mid w_i) \tag{8}$$

where P(*s*|$w_i$) is the probability of sense category s given a token $w_i$. Such probability can be considered further as the score for token *w* to vote for sense category *s*. Eq. (8) can be expressed as:

$$P(s \mid w) = Score(s \mid w) = \frac{C(s,w)}{TC(w)} \tag{9}$$

where C*(s,w)* denotes the count of token *w* occurred in feature database for certain sense category *s*. *TC(w)* is the total count of token *w* in feature database for target symbol.

*Score(s|w)* is the relative frequency, which can be regarded as the score of token *w* voting to sense category *s* in our voting approaches. Eq. (9) satisfies the Bayesian rule and easy to understand intuitively. When computing the probability score of each word *w* for sense category *s*, we just need to use token count *C(s,w)* and total count *TC(w)* with respect to the sense category *s* and target symbol. So, the *Score*(*s*|*w*) can be computed easily for all tokens in the word sequence *W* of sentence. The probability can be regarded further as a score for each token in $CH_L$ and $CH_R$ to vote for each category of non-text symbol.

Referring to the Eq. (10), the score[1] $Score_L$ and $Score_R$ of each token in $CH_L$ and $CH_R$ voting for sense category $s_j$ of non-text symbol can be computed as:

$$Score_L(s_j, w_{-i}) = \frac{C_L(s_j, w_{-i})}{TC_L(w_{-i})} \ , \quad Score_R(s_j, w_{+i}) = \frac{C_R(s_j, w_{+i})}{TC_R(w_{+i})} \tag{10}$$

where $-1 \le -i \le -m$ and $+1 \le +i \le +n$ , $w_{-i}$ and $w_{+i}$ are labeled as the token *w* in $CH_L$ and $CH_R$ . $C_L(s_j, w_{-i})$ and $\mathrm{C}_R(s_j, w_{+i})$ are the count of token $w_{-i}$ and $w_{+i}$ occurred in $CH_L$ and $CH_R$ for the category $s_j$ in feature database. $TC_L(w_{-i})$ and $TC_R(w_{+i})$ stand for the total count of $w_{-i}$ and $w_{+i}$ occurred in $CH_L$ and $CH_R$ , which can be computed as:

$$TC_L(w_{-i}) = \sum_{j=1}^{J} C_L(s_j, w_{-i}) \ , \qquad TC_R(w_{+i}) = \sum_{j=1}^{J} C_R(s_j, w_{+i}) \tag{11}$$

$$\sum_{j=1}^{J} Score_L(s_j, w_L) = 1 \ , \qquad \sum_{j=1}^{J} Score_R(s_j, w_R) = 1 \tag{12}$$

where J denotes the number of sense category for target symbol.

By definition of *score( )* above, $Score_L(s_j, w_{-i})$ and $Score_R(s_j, w_{+i})$ can be regarded as the relative frequency which the $w_{-i}$ and $w_{+i}$ will occur in the sense category $s_j$. As the result, our voting schemes are based on such probability value.

For the 2nd decision classifier in MLDC, the total score $TScore_L$ ( • )and $TScore_R$ ( • ) for all the tokens in substring $CH_L$ and $CH_R$ of example *E* to vote for sense category $s_j$ can be computed as:

[1] The resulting score of each token fall between 0 and 1, while it is possible that the accumulated scores of all tokens in sentence for certain sense category will be greater than 1.

$$TScore_L(s_j) = \sum_{-i=-1}^{-m} Score_L(s_j, w_{-i}) \ , \quad TScore_R(s_j) = \sum_{+i=+1}^{+n} Score_R(s_j, w_{+i}) \qquad (13)$$

In 2nd decision classifier, total score *TScore*( • ) of all tokens in example E for each sense category are displayed as:

$$\underset{s_j \in possi\_set}{TScore(s_j)} = TScore_L(s_j) + TScore_R(s_j) \qquad (14)$$

*TScore*( • ) will be used in Eq. (2) by the multi-layer decision classifiers to predict the final sense category $s_j$.

**3.4 The Probability of Unknown token**

Several well-known methods for probability of unknown words are described in [Su etc.,1996; Daniel et al.,2000]: *additive discounting*, *Good-Turing* and *Back-Off* . The principle reason is that there are a lot of tokens in natural language, usually more several ten thousands. New lexicons or tokens will be occurred in near future. Within natural language processing, it is so hard to collect all the words.

In our paper, the so-called unknown tokens can be considered that do not occur in our feature database, which have been generated in the training phase. It is so apparent that the distribution and total number of collected data set will affect the statistical parameters seriously, especially on the statistical models. Another situation is the data sparseness. The smoothing techniques can alleviate the problems. In this paper we use additive discounting and assign 0.5 to the count of unknown tokens.

# 4. Evaluations

The experiments with elementary approach and schemes are evaluated first. Two different scoring scheme adopted by our classifier are tested to decide which is better for WSD problems in this paper. We will compare the 2nd classifier in MLDC with the well-known language model. The location effectiveness with respect to different token unit (Mandarin word or character) is also evaluated in final subsection.

**4.1 Evaluation for Two Scoring Schemes**

At first, we will describe the voting scheme with winner-take-all scoring then compare such two scoring schemes. In contrast to the so-called preference-scoring scheme described in Section 4.3, the voting scheme with *winner-take-all* scoring adopts a different scoring rule. Ho Lee etc [1997]. Lee employed the *winner-take-all* scoring scheme to word sense disambiguation, without comparison between these two schemes in his paper. Lee's precision rate was 80% average.

For each token in sentence, $Score_L(s_{j^*}, w_{-i})$ and $Score_R(s_{j^*}, w_{+i})$ will be assigned the score 1 to sense category $s_j^*$ for token $w_{-i}$ and $w_{+i}$ and 0 to all the other sense categories. Eq. (10) should be rewritten as:

$$Score_L(s_{j^*},w_{-i})=\begin{cases}1 & \text{if } s_{j^*}\in possi\_set \text{ and } Score_L(\mathrm{s}_{\mathrm{j}^*},w_{-i_1})=\arg\max\limits_{j=1,2,\ldots J}(Score_L(\mathrm{s}_{\mathrm{j}},w_{-i}))\\ 0 & \text{otherwise}\end{cases} \quad (15)$$

$$Score_R(s_{j^*},w_{+i})=\begin{cases}1 & \text{if } s_{j^*}\in possi\_set \text{ and } Score_R(s_{j^*},w_{+i})=\arg\max\limits_{j=1,2,\ldots J}(Score_R(s_j,w_{+i}))\\ 0 & \text{otherwise}\end{cases} \quad (16)$$

where sense category $(s_{j^*})$ is with respect to the category of which the $Score_L(s_{j^*},w_{-i})$ and $Score_R(s_{j^*},w_{+i})$ have the maximum score among all categories for $w_{-i}$ and $w_{+i}$. Based on the voting scheme with winner-take-all scoring, Eqs. (10) – (14) should not be modified.

In case that several sense categories have the maximum score for token *w*, Eqs (15) and (16) should be revised. The total probability score 1 for token *w* will be shared by these sense categories. It means that the total score 1 will be divided by the number of sense categories with same maximum score.

The first parameter to be evaluated is the scoring scheme for each token. Figure 6 displays an example of the accumulated score for 5 categories using two different scoring methods: preference and winner-take-all scoring on the Eqs (15) and (16). The example (E1) contains 15 individual tokens (including symbol ":"). Sense category *time* ($s_2$) gets the maximum score 6.92 in Figure 1. Similarly, it still gets maximum score 7.0 by using the winner-take-all scoring.

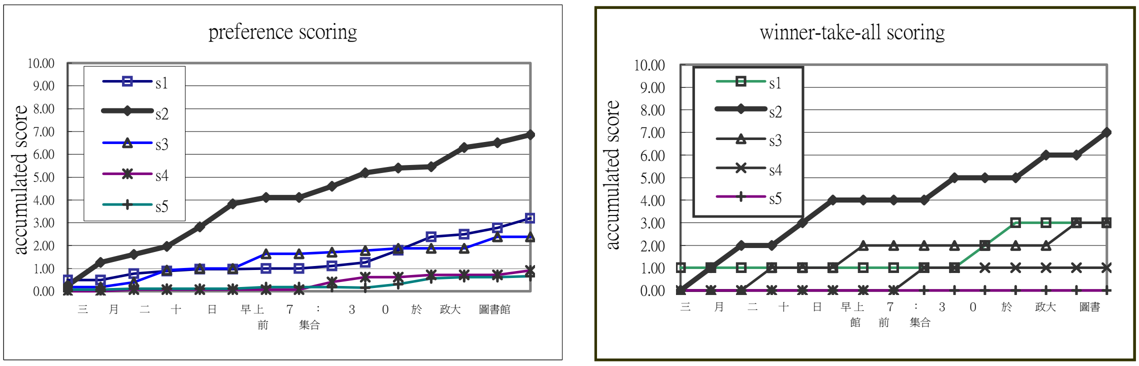

**Figure 1:** (left) accumulated score of categories for non text symbol “:” based on the preference scoring ; category *time* ($s_2$) gets maximum score 6.92. (right) based on winner-take-all scoring; the category *time* ($s_2$)

(E1) 三月二十日早上 $\boxed{7 : 30}$ 於政 大圖書館前集合。

| $S_n$ / scoring scheme | 1<br>punctuation | 2<br>time | 3<br>versus | 4<br>telephone | 5<br>expression | prediction |
|---|---|---|---|---|---|---|
| winner-take-all | 3.0 | **7.0** | 3.0 | 1.0 | 0.0 | correct |
| preference | 3.1 | **6.9** | 2.3 | 0.9 | 0.8 | correct |

The sense category *time* ($s_2$) in (E1) gets maximum score in two scoring schemes, however, some other examples may not hold yet. Especially, while top 2 scores are so close it is possible that the sense category with second maximum score will precede the first category with maximum score by employing different scoring scheme. For instance, as shown in example (E2), the sense category *date* ($s_1$) got the maximum score and is predicted as the final category by using the winner-take-all scoring scheme. Instead of such scoring scheme, we use the preference scoring to predict the category and the result is correct. In fact, the substring “1/3” means “one third”. This is an example that winner-take-all scoring makes a wrong prediction while preference scoring can finds the correct sense category. The scores for each sense category[2] are listed below example (E2).

(E2) 價格 比 台灣 便宜 約 1／3 左右 ，

| $s_n$ <br> scoring | 1<br>date | 2<br>fraction | 3<br>time | 4<br>directory | 5<br>computer term | 6<br>version | 7<br>others | prediction |
|---|---|---|---|---|---|---|---|---|
| winner-take-all | **4.0** | 0.0 | 3.0 | 0.0 | 1.0 | 1.0 | 0.0 | incorrect |
| preference | 3.1 | 0.4 | **3.2** | 0.4 | 0.9 | 0.7 | 0.4 | correct |

Table 7 lists the performances with two voting schemes: preference and winner-take-all scoring. Obviously, the former is superior uniformly to the later on both inside and outside testing for three symbols. So we adopt the voting scheme of preference scoring, excluding winner-take-all scoring, for all following experiments. Note that the 2nd decision classifier in MLDC, based on the voting scheme of preference scoring with Mandarin word’s token, is regarded as the *baseline* model in this paper. As shown in Table 7, the net results are enhanced up 5.5% and 5.9% for inside and outside testing respectively.

**Table 7:** The performance of the 2nd decision classifier (baseline) in MLDC; employing two scoring scheme.

| scoring scheme <br> precision rate(%) | preference | | winner-take-all | |
|---|---|---|---|---|
| | inside test | outside test | inside test | outside test |
| “／” | 99.2 | 94.6 | 92.9 | 84.8 |
| “：” | 95.7 | 91.1 | 91.5 | 84.1 |
| “＿“ | 96.8 | 85.7 | 90.8 | 83.5 |
| average (net) | **97.2(+5.5)** | **90.0(+5.9)** | 91.7 | 84.1 |

### 4.2 Comparing the 2nd Classifier with *n*-gram Models

In this Section, we will compare baseline defined in previous subsection with the *n*-gram

[2] All the sense categories for three target symbols discussed in our paper are displayed in Tables 1-3.

(*n*=1, 2 in this experiments), widely used in various domains of natural language processing. The base line model displays attractive empirical results.

Table 8 indicates the performance of three models: baseline with voting scheme, *uni*-gram and *2*-gram, on the same testing data set without employing the 1st layer decision classifier or other techniques. Comparing the *2*-gram with *uni*-gram, it is so apparent that the former is superior to the latter. The average net results for inside and outside test are 1.3% and 4.3% respectively.

We observe further the performance between baseline and *n*-gram. The minimum difference between is +1.4% for outside testing of target symbol ":". The baseline is superior to *2*-gram model for all target symbols. The average net results for inside and outside test are 0.5% and 4.7%.

Because of the data sparseness and small size of data set on our WSD problem, there are more unknown tokens for *n*-gram model than that for baseline. The performance for outside testing of *n*-gram is upgraded by baseline model for three target symbols. The ratio of unknown tokens (words) for three target symbols: 11.8%, 15.3% and 19.3%. The more the unknown tokens appear, the lower the performance is. The size of unknown tokens will affect seriously the performance of *n*-gram model. The zero count of token leads to the degradation for *n*-gram.

**Table 8:** Comparisons between our *base line* and *n*-gram (n=1,2).
The numbers in parenthesis denote the net performance comparing *base line* with *2*-gram.

| scheme | inside test | | | outside test | | |
|---|---|---|---|---|---|---|
| symbols | *base line* | *uni*-gram | *2*-gram | *baseline* | *uni*-gram | *2*-gram |
| “ ／” | 99.2(+0.3) | 97.6 | 98.9 | 94.6(+2.0) | 90.5 | 92.6 |
| “ : ” | 95.7(+0.5) | 92.2 | 95.2 | 91.1(+1.4) | 79.9 | 89.7 |
| “—” | 96.8(+0.7) | 95.9 | 96.1 | 85.7(+9.2) | 74.3 | 76.5 |
| average (net) | **97.2(+0.5)** | 95.4 | 96.7 | **90.0(+4.7)** | 81.0 | 85.3 |

### 4.3 Merging Two Layer Classifiers Together

In addition to our baseline model, we will analyze further the effectiveness of the 1st classifier in MLDC. Two classifiers in MLDC could be merged together to improve the prediction rate.

For instance, example (E3) shows the effectiveness of merging the 1st layer classifier into baseline (the 2nd layer classifier). Exploiting the 1st classifier to exclude some impossible categories first. As shown in example (E3), the sense category with maximum score (2.4), predicted by using the 2nd layer classifier with voting scheme only, is *date* ($s_1$) and it is apparent that the prediction is incorrect. The number of $w_{+1}$ token (32) in pattern “3/32” is larger than 31, which is the maximum number of date. Therefore sense category *date*[3] was excluded for target symbol “/” by the 1st layer classifier. However, the category *music time* ($s_3$) with second maximum score (1.8) was predicted as the final one among all remained categories correctly by the 2nd layer classifier with voting scheme.

(E3) 演奏 的 曲子 是 ３／３２ 拍 且 爲 Ｄ 大調 。

| $s_n$ <br> merging | 1<br>date | 2<br>fraction | 3<br>time | 4<br>directory | 5<br>computer term | 6<br>version | 7<br>others | prediction |
|---|---|---|---|---|---|---|---|---|
| 2nd classifier only | **2.4** | 1.3 | 1.8 | 1.1 | 0.3 | 0.7 | 0.4 | incorrect |
| merging two classifier | 2.4* | 1.3 | **1.8** | 1.1 | 0.3* | 0.7 | 0.4* | correct |

ps. * denotes the sense category was excluded by the 1st layer decision classifier.

The performances are attractive and listed in Table 9. As shown, the final results for outside testing is 97.8, 95.6 and 92.1 for three symbols respectively by combining the 1st and 2nd classifier with voting scheme of preference scoring in 2nd classifier. The numbers in parenthesis are the net results. The average net results by merging two classifiers are upgraded 0.5% and 4.5% (referring to Table 8 and Table 10).

**Table 9:** The effectiveness of merging the 1st and 2nd decision classifiers

| | merging 1st classifier ? | inside testing | outside testing |
|---|---|---|---|
| “／” | without merging | 99.2 | 94.6 |
| | merging | **99.5(+0.3)** | **97.9(+3.3)** |
| “：” | without merging | 95.7 | 91.1 |
| | merging | **98.3(+2.6)** | **95.6(+4.5)** |
| “＿“ | without merging | 96.8 | 85.7 |
| | merging | **98.4(+1.6)** | **92.1(+5.4)** |
| average | merging | **97.7** | **94.5** |

### 4.4 Evaluation for the Effect of Word’s Location

In previous Section, the location of each token is just labeled two types: preceding (*p*) and following (*f*) the target symbol. While the count for each token was statistically accumulated, we just consider whether the token is located within the chunk-L ($CH_L$) or chunk-R ($CH_R$) of

[3] In fact, the decision tree excludes three sense categories: *date*, *computer term* and *version*.

sentence. Will the performance be improved by considering further the individual location of each token in $CH_L$ ($w_{-i}$) and $CH_R$ ($w_{+i}$)? In this Section, the effect of individual location for each token (word) will be evaluated further.

In this Section Token unit is still Mandarin word. Instead of the two chunk types described previously, each token is labeled with the individual location in $CH_L$ and $CH_R$, in which the count of each token occurred in same location will be summed up with respect to the sense category. So the technique is the *word-based* scheme with individual location.

The Eqs. (10)-(12) can be changed as follow:

$$Score_L(s_j, w_{-i}) = \frac{C_{-i}(s_j, w_{-i})}{TC_{-i}(w_{-i})} \qquad Score_R(s_j, w_{+i}) = \frac{C_{+i}(s_j, w_{+i})}{TC_{+i}(w_{+i})} \tag{17}$$

$$TC_{-i}(w_{-i}) = \sum_{j=1}^{J} C_{-i}(s_j, w_{-i}) \qquad TC_{+i}(w_{+i}) = \sum_{j=1}^{J} C_{+i}(s_j, w_{+i}) \tag{18}$$

$$\sum_{j=1}^{J} Score_L(s_j, w_{-i}) = 1 \ , \qquad \sum_{j=1}^{J} Score_R(s_j, w_{+i}) = 1 \tag{19}$$

where $i$ is the location of word with respect to the non-text symbol , -*m*<= -*i* <=-1 and 1<=+i<=n. $C_{-i}(s_j, w_{-i})$ and $\mathrm{C}_{+\mathrm{i}}(s_j, w_{+i})$ are the count of word $w_{-i}$ and $w_{+i}$ with the location –*i* and +*i* occurred in feature corpus for sense category $s_j$ respectively.
$TC_{-i}(w_{-i})$ and $TC_{+i}(w_{+i})$ are the total count of word $w_{-i}$ and $w_{+i}$ occurred in the location –*i* and +*i* in feature database respectively

Let's take a look at the example (E4), the sense category (*date*) is incorrectly predicted based on the chunk scheme whereas correctly predicted on individual location of each token.

(E4) 曹 錦輝還 有機會 在 １１／２０ 代 表 台灣大聯 盟 與統一 隊比 賽。

| $s_n$ token location | 1<br>date | 2<br>fraction | 3<br>time | 4<br>directory | 5<br>computer term | 6<br>version | 7<br>others | prediction |
|---|---|---|---|---|---|---|---|---|
| chunk | 2.9 | 1.7 | **5.2** | 1.2 | 0.2* | 1.8* | 4.0 | incorrect |
| individual | **2.4** | 0.5 | 2.2 | 0..5 | 0.2* | 0.4* | 0.2 | correct |

Comparing two schemes of token (word) with individual and two chunks' location, the net precision rates of outside testing are 0.6%, 1.5% and –0.3% for three target symbols respectively. As Shown the Table 10, the former is average superior to the later, in which the sentence is divided into two chunks ($CH_L$ *or* $CH_R$). Referring to the accumulated score for correct predicted sense category, although the rate of unknown words token in data set reaches about 45%, the former still make the prediction efficiently. However, it is easier for the techniques with voting scheme, which identify a half of total tokens in sentence, to make the correct prediction. The net precision rates for inside and outside testing are 0.2 and 0.6.

**Table 10:** The comparison of two location schemes for each token.

| | inside testing | | outside testing | |
|---|---|---|---|---|
| | individual | chunk | individual | chunk |
| “／” | **99.3(-0.2)** | 99.5 | **98.6(+0.7)** | 97.9 |
| “：” | **99.2(+0.9)** | 98.3 | **97.1(+1.5)** | 95.6 |
| “—” | **98.2(-0.2)** | 98.4 | **91.8(-0.3)** | 92.1 |
| average | **98.9(+0.2)** | **98.7** | **95.1(+0.6)** | **94.5** |

### 4.5 Evaluation for Effect of Token Unit

Until now, the sentence will be divided into two chunks: chunk-L($CH_L$) and chunk-R($CH_R$), which are in the left and right side of target symbol *TS* in sentence. Such chunks will be segmented into one to several words based the ASCED and segmentation scheme. In Mandarin Vocabulary, there are about 70000 frequent Mandarin words, which are composed of one to ten characters. For example, the number for 1-character token (Mandarin word) is 7522 and 48315 for 2-character token (Mandarin word) in ASCED while just 13053 for frequent Mandarin characters. It is apparent that segmented sentence will generate more unknown tokens for the same data set. The more unknown tokens are in sentence, the less precision rate will be. The process of word segmentation may generate possible mistake, which will also degrade the performance of prediction. Usually the situation becomes serious if the data set is sparse or volume of sentence is small.

In this section, the sentence will not be segmented so each character in sentence is the voting token. The location of each character will be considered same as described in previous section. The token unit is character with the individual location in $CH_L$ or $CH_R$, in which the count of each character occurred in same chunk ($CH_L$ or $CH_R$) will be summed up with respect to the sense category. So the technique is the *character-based* scheme with individual location. Example *E* is still composed of three parts: $CH_L$, *TS* and $CH_R$. Each chunk may comprise one to several characters. Note that the foreign words (such as: *IBM*, *DR.*, *Windows*, etc.) within chunk will be regarded as a token.

$$\begin{aligned} E &= CH_L + TS + CH_R \\ CH_L &= c_{-m}c_{-(m-1)}\cdot\cdot\cdot c_{-j}\cdot\cdot\cdot c_{-1} \\ CH_R &= c_{+1}c_{+2}\cdot\cdot\cdot c_{+j}\cdot\cdot\cdot c_{+n} \end{aligned} \tag{20}$$

where *c* denotes the individual character in $CH_L$ and $CH_R$ and *m*, *n* the number of characters in $CH_L$ and $CH_R$ respectively. The Eqs. (10)-(12) of probability scoring can be rewritten as:

$$Score_L(s_j, w_{-i}) = \frac{C_{-i}(s_j, c_{-i})}{TC_{-i}(c_{-i})}, \quad Score_R(s_j, w_{+i}) = \frac{C_{+i}(s_j, c_{+i})}{TC_{+i}(c_{+i})} \tag{21}$$

$$TC_{-i}(c_{-i}) = \sum_{j=1}^{J} C_{-i}(s_j, c_{-i}), \quad TC_{+i}(c_{+i}) = \sum_{j=1}^{J} C_{+i}(s_j, c_{+i}) \tag{22}$$

$$\sum_{j=1}^{J} Score_L(s_j, c_{-i}) = 1 \ , \quad \sum_{j=1}^{J} Score_R(s_j, c_{+i}) = 1 \tag{23}$$

where $i$ is the location of character with respect to the non-text symbol , $-m$<= $-i$ <=-1 and 1<=$+i$<=$n$. $C_{-i}(s_j, c_{-i})$ and $C_{+i}(s_j, c_{+i})$ are the count of character $c_{-i}$ and $c_{+i}$ occurred in feature corpus with the location $-i$ and $+i$ for sense category $s_j$ respectively.

$TC_{-i}(c_{-i})$ and $TC_{+i}(c_{+i})$ are the total count of character $c_{-i}$ and $c_{+i}$ occurred with the location $-i$ and $+i$ in feature corpus respectively

The total score $TScore_L$ ( • )and $TScore_R$ ( • ) for all individual characters of $CH_L$ and $CH_R$ in example $E$ to vote for sense category can be computed like Eqs. (13) and (14). The method will be regarded as the character-based approach with location scheme.

Until now, the adopted token unit of sentence is Mandarin *word*. There are some possible errors occurred during the segmentation process for generating the token (word). Based on the character[4] token unit with location scheme, there are fewer unknown token. The example (E5) in our data set is divided into two chunks, in which the individual token is the character without needing the word segmentation. The characters in $CH_L$ will be labeled with location $-m$~-1 and the characters in $CH_R$ labeled with +1~+$n$. (E5) is an example in which the correct sense category can't be predicted by using the scheme with word token, while it can be correctly predicted by using character as token.

(E5) 結 果 １０／１０ 那 天 桃 園 縣 建 築 師 再 來 認 定 時 ，[5]

| $s_n$ <br> location/tioken | 1<br>date | 2<br>fraction | 3<br>time | 4<br>directory | 5<br>computer term | 6<br>version | 7<br>others | prediction |
|---|---|---|---|---|---|---|---|---|
| individual/word | 2.4 | 0.5 | **3.5** | 0.2 | 0.2* | 0.4* | 0.2 | incorrect |
| individual/character | **1.3** | 0.8 | 0.7 | 0.2 | 0.1* | 0.1* | 0.3 | correct |

Intuitively, in natural language processing of Mandarin, the token unit used is usually word, which is the basic unit containing complete and useful semantic information. Instead,

[4] The Mandarin characters we use is 13053, which are collected in the BIG-5 character set.

[5] In contract to our previous example, each Mandarin character here is regarded as a token, without word

why the performance for *character* tokens is superior to that for *word* tokens both with individual location?

Depending on our observations, there are three following reasons with respect to such phenomenon. First, it is not easy for the process of word segmentation to generate the most portable word sequence *W*. The second reason is the data sparseness; the situation exists in our WSD problem and more unknown tokens will happen. The third, related to the unknown token, is the token unit. The number for Mandarin character is approximately 13,000 whereas 70,000 for Mandarin word. It is obvious that adopting word's token will lead to more unknown tokens than that of character's token. Such situation will affect the performance. As described below, suppose that a two-character word “昨天(yesterday)” occurred with specific location in our feature database. Now a token “今天(today)” in a testing example occurs, labeled by same location of token “昨天”, and will be still regarded as a unknown token based on the token with scheme of individual location. However, the token “昨天” can further be divided into two characters: “昨” and “天”. The second character of word “昨天” and “今天” is both “天”. So character “天” is a known token and can provide the statistical information based on the character token with individual location. Referring to Table 10, the average precision rates in Table 11 are upgraded 0.5% and 0.4% for inside and outside testing obtained from the individual location for each token (character).

**Table 11:** Two token units: *word* and *character*. Each token is labeled by individual location.

| | inside testing | | outside testing | |
|---|---|---|---|---|
| | character | word | character | word |
| “ ／” | **99.6(+0.3)** | 99.3 | **98.3(-0.3)** | 98.6 |
| “ ： ” | **99.6(+0.4)** | 99.2 | **98.1(+1.0)** | 97.1 |
| “－” | **99.2(+1.0)** | 98.2 | **92.4(+0.6)** | 91.8 |
| average | **99.4(+0.5)** | 98.9 | **95.5(+0.4)** | 95.1 |

Currently, the elementary experiments have been implemented and several schemes in our proposed approach were evaluated. The best performance for WSD problem based on such empirical parameters can be achieved. In summary, that are the following empirical features: preference scoring, merging the 1st and 2nd decision classifier together, individual location (*-m*~*+n*) of token, character token. The precision rates, obtained by using the techniques above, of outside testing are 98.3%, 98.1% and 92.4% (95.5% average) for the three target symbols respectively.

# 5. Further Improvements

In this Section, we will discuss several features of token in example to improve the performance. At first, the weighting of token in different location with respect to the target symbol will be analyzed. We hope to find the effectiveness of weighting value for each individual token. Another technique is subject to the specific patterns contained in example. Such patterns represent a special semantic meaning. In the next subsection, we will discuss the difference of top 2 score for each example. A threshold value will be used to decide when the alternative technique can be used to improve the performance.

## 5.1 Weights for Individual Token

It is our intuition that the nearer a token is to target symbol, the higher prediction capability to token is. So in this Section we will try to find the effect of the tokens in different locations. And possibly, we can assign different weights to tokens with respect to its location in sentence.

The function *weight(i)* denotes the weighting value for token unit with location *i ,* which can be derived from experiments for three symbols. Therefore, the related Equations, Eqs. (13) and (14), will be revised as:

$$TScore_L(s_j) = \sum_{i=-1}^{-m} (Score_L(s_j, w_i) * weight(i)) \qquad (24)$$

$$TScore_R(s_j) = \sum_{\mathrm{i}=1}^{-n} (Score_R(s_j, w_i) * weight(i)) \qquad (25)$$

## 5.2 Pattern Table

In this subsection, we will discuss the patterns in text, which belong to the specific sense category and can be assigned directly. For instance, example (E6) contains the pattern “42/7”, which is incorrectly predicted as category *others* ($s_7$) with maximum score 4.6 generated by MLDC.

In fact, the pattern “42/7” stands for a name of network company. The target symbol “/” in “24/7” will be a silence. Therefore the pattern should be pronounced directly in Mandarin “四十二 (shi si er), a silence and 七 (chi)”. All such specific patterns, which are ambiguous and represent the specific term, such as a company name, specific date “9/21” etc., will be collected into the pattern table. Such table should be searched in front of adopting the MLDC. If the specific patterns of examples are found, its associated sense category will be assigned immediately without the prediction of MLDC. Currently, there are 12 entries collected in our pattern table. The use of pattern table can resolve several special cases and improve the performance by the amounts 0.6% ~ 1.0% for the three target symbols.

(E6) 42／7 可 協助 網 站 解決 網路 廣告 存貨 問題。

| $s_n$<br>method | 1<br>date | 2<br>fraction | 3<br>time | 4<br>directory | 5<br>computer term | 6<br>version | 7<br>others | prediction |
|---|---|---|---|---|---|---|---|---|
| our approach | 1.6* | 0.9 | 1.6* | 0.6 | 0.1* | 1.5 | **4.6** | incorrect |

### 5.3 Adopting the Alternative

In the previous section, we introduced the token schemes of word and character, which are based on the different token unit in sentence. Finally the best average precision rate of outside test are 97.83%, 98.46 and 92.37% for symbols "/", ":" and "-" respectively using the character token scheme with location. One consideration is that whether the performance can be improved further by merging different token schemes or not? Although the token scheme of characters can obtain highest precision rate currently, what is the condition to adopt the alternative schemes to improve the performance further?

The normalized difference is defined as: $(score_1\text{-}score_2)/NT$. $score_1$ and $score_2$ are the top 2 score computed by proposed approach for target symbols. *TN* denotes the token number of sentence and will be changed with different token schemes. *TN* will normalize the difference of top 2 scores.

Note that the Elementary approach here was described at the end of Section 4.5. The final empirical performances of inside and outside testing are 99.6% and 96.5% average, employing the improving techniques proposed in this Section.

## 6. Conclusions

We have developed an approach, which contains the multi-layer decision classifiers and can disambiguate the sense ambiguity of non-alphabet symbols in Mandarin effectively. In contract to the *n*-gram language models, the new approach just needs smaller size of corpus and still hold the linguistic knowledge for statistical parameters. The model with voting scheme (baseline) is superior to *n*-gram (*n*=1,2) model. Several techniques are proposed and evaluated in our elementary experiment. Some examples are displayed to illustrate for each technique. The precision rates are 99.4% and 95.5% for inside and outside testing.

Three techniques are proposed to improve the performance further: weights for token with individual location, pattern table and the alternative. The final precision rates of further improvements are 99.6% and 96.5% for inside and outside test respectively.

In addition to the target symbols " /", ":" and "-" analyzed in the paper, there are some other symbols, such as *, %, [] and so on, in which the oral ambiguity problems will be incurred and should be resolved. Our approaches can be extended into related symbols.

# References

P. Brown, S. Della Pietra, V. Della Pietra and R. Mercer. *Word Sense Disambiguation Using Statistical Methods*. In Proceeding of the 29th Annual Meeting of the Association for Computational Linguistics, Berkeley, pp. 264- 270, 1991.

Atsushi Fujii, Kentaro Inui, *Selective Sampling for Example-base Word Sense Disambiguation*, Computational Linguistics, vol. 24, number 4, 1998,pp 573-597.

William Gale, Kenneth W., Church and David Yarowsky, *A Method for Disambiguating Word Sense in a Large corpus*, Computer and the Humanities, 1992, Vol. 26.

A.R.Golding, *A Bayesian hybrid method for Context-Sensitive Spelling Correction*, In Proceedings of the third workshop on Very Large Corpora, pp. 39-53, Boston, USA, 1995.

Chu Ren Huang, Introduction to the Academic Sinica Balance Corpus, Proceeding of ROCLLING VII*,* pp. 81-99, 1995.

Feng-Long Hwang, Ming-Shing Yu, Min-Jer Wu and Shyh-Yang Hwang**,** *Semantic Classification for Patterns Containing Non-alphabet Symbols in Mandarin Text,* ROCLING XII, NCTU, 1999a, pp. 55-66.

Feng-Long Hwang, Ming-Shing Yu, Min-Jer Wu and Shyh-Yang Hwang**,** *Sense Disambiguation of Non-alphabet Symbols in Mandarin Text Using Multiple Layer Decision Classifiers*, Proceedings of 5$^{th}$ Natural Language Processing Pacific Rim Symposium (NLPRS), Beijing China, 1999b, pp. 334-339.

Nancy Ide and Jean Veronis, *Introduction to the Special issue on Word Sense Disambiguation: The State of the Art*, Computational Linguistics, vol. 24, number 1,1998,pp 1-40.

Daniel Jurafsky, James H. Martin, Speech and Language Processing, Printice Hall, 2000.

Ho Lee, Dae-Ho Baek, Hae-Chang Rim, *Word Sense Disambiguation Based on the Information Theory*, Proceedings of ROCLING X International Conference, Research on Computational Linguistics, Taiwan, pp. 49-58,1997

Claudia Leacock, Geoffery Towell, and Ellen M. Voorhees, *Corpus-based Statistical sense Resolution*, In proceedings of ARPA Workshop on Human Language Technology, San Francisco, CA, Morgan Kaufman, 1993.

Hinrich Schutze, *Ambiguity and Language Learning: Computational and Cognitive Models*, Ph. D Thesis, and Standard University, 1995.

K. Y. Su, T. H. Chiang, J. S. Chang, *A Overview of Corpus-Based Statistical-Oriented (CBSO) Techniques for Natural Language Processing*, Computational Linguistics and Chinese Language Processing, vol. 1, no. 1, pp.101-157, August 1996.

Jean Verious and Nancy Ide, *Word sense Disambiguation with very large neural extracted from Machine Readable Dictionaries*, in proceeding of COLING-90, 1990.

David Yarowsky, *Homograph Disambiguation in Text-to Speech Synthesis*, pp.157-172, 1997.

Chinese Knowledge Information Processing (CKIP) Group, *Technical Report: The Content of Academia Sinica Balanced Corpus(ASBC)* , Sinica Academia, R.O.C., 1995.