Computational Linguistics and Chinese Language Processing

Vol. 4, No. 2, August 1999, pp. 1-20

A Model for Word Sense Disambiguation

Li Juanzi*, Huang Changning*

Abstract

Word sense disambiguation is one of the most difficult problems in natural language processing. This paper puts forward a model for mapping a structural semantic space from a thesaurus into a multi-dimensional, real-valued vector space and gives a word sense disambiguation method based on this mapping. The model, which uses an unsupervised learning method to acquire the disambiguation knowledge, not only saves extensive manual work, but also realizes the sense tagging of a large number of content words. Firstly, a Chinese thesaurus *Cilin* and a very large-scale corpus are used to construct the structure of the semantic space. Then, a dynamic disambiguation model is developed to disambiguate an ambiguous word according to the vectors of monosemous words in each of its possible categories. In order to resolve the problem of data sparseness, a method is proposed to make the model more robust. Testing results show that the model has relatively good performance and can also be used for other languages.

Key Words: natural language processing, word sense disambiguation, unsupervised learning, vector space, language modeling

1. Introduction

Word sense disambiguation, that is, identifying the correct sense of a word from all its senses as defined in a dictionary or a thesaurus, has long been one of the most difficult problems in natural language processing. In the 1990s, the research on this topic has entered a new phase with the availability of machine-readable dictionaries and very large corpora. Such research mainly falls into two classes: dictionary-based and corpus-based methods. The dictionary-based disambiguation methods, such as the models put forward by Lesk [1986] and Wilks [1990], do not perform well when the context of a word has little overlap with the text of its dictionary definition. Corpus-based methods, such as the

* The State Key Laboratory for Intelligent Technology and Systems, The Department of Computer Science and Technology, Tshinghua University, Beijing 100084.
e-mail: ljz@s1000e.cs.tsinghua.edu.cn

disambiguation model proposed by Yarowsky [1992] using a thesaurus (Roget's) and a corpus, suffer from a large amount of noise for polysemous words with high frequency in the statistical data, which greatly degrades the results. Data sparseness is another problem of corpus-based methods. Furhtermore, most of the models based on corpus-based methods have been tested on only ten to twenty polysemous words each with two or three selected senses.

Therefore, we think that a faithful word sense disambiguation model must accomplish two critical tasks. One is to acquire disambiguation knowledge from a very large raw corpus instead of from a manually tagged corpus. The other is to develop a disambiguation model suitable for dealing with the great majority of polysemous words.

In this paper, we first provide evidence that there is some degree of consistency between the distribution of words in classes provided by the thesaurus *Cilin* and the distribution of their context vector representations in vector space. We then convert *Cilin's* structural semantic space into a vector space and propose a word sense disambiguation model based on it. In order to solve the "bottleneck" problem encountered in acquiring word sense disambiguation knowledge, the model makes full use of the distribution of monosemous words in the same class in *Cilin* to disambiguate the polysemous words. Therefore, a large amount of the labor involved in manually tagging word senses is eliminated. Section 2 gives a definition of *Cilin's* structural semantic space and its projection into vector space. Then, some evidence from clustering experiments is presented to support the reasonableness of the above projection. Section 3 puts forward a disambiguation criterion based on the semantic space and gives details of its implementation. Section 4 presents our experiments and analysis of the results. Section 5 is a conclusion, which explains the differences between the method proposed in this paper and other word sense disambiguation methods.

2. *Cilin's* structural semantic space

Semantic space can be established by automatically clustering words using a very large corpus or by simply using an existing thesaurus compiled by linguists. The weaknesses of the former are that the meaning of the clustered class needs to be judged by humans, and that the objects to be clustered are in word forms other than word senses. On the other hand, the presently available thesauri are not perfect resources of information about word relations because they have been compiled for human. Therefore, these thesauri, such as *Cilin*, have not been used extensively for word sense disambiguation. However, such thesauri, especially the information in the lower levels, do provide a rich network of word associations and a set of semantic categories potentially valuable for natural language processing. Based on this idea, this paper will try to construct a semantic space using an existing thesaurus, *Cilin*, and a very large corpus.

2.1 *Tongyici Cilin*

The *Cilin* thesaurus, which was originally developed for people to use to select appropriate words in translation and writing, is the only machine-readable Chinese thesaurus presently available. It is organized in a hierarchical structure, with all of the word entries classified into 12 major classes, 94 medium classes and 1428 minor classes. The major classes, middle classes and minor classes are presented by one-character, two-character and four-character semantic codes, respectively. To show even finer differences in meaning, the compilers have further divided the minor classes under several headings. For example, the semantic code "Hc04" stands for "PaiQian(dispatch), ZhiShi(order about)." Under "Hc04," there are two sub-classes with titles the" 派遣 (PaiQian, dispatch)" and" 支使 (ZhiShi, order about)," respectively :

派遣 指派 派 特派 打 ……
支使 支派 使 指名 ……

Thus, we give them two more characters. For example, the semantic code of " 派遣 (PaiQian)" is "Hc0401," and that of " 支使 (ZhiShi)"is "Hc0402".

2.1.1 Words in *Cilin*

Because a word sense can be presented by a semantic code in *Cilin*, a polysemous word corresponds to several different codes. For example, " 材料 (CaiLiao)" is a polysemous word with such senses as 1) material which can be used to produce a product, 2) a document which can be referred to, and 3) people who have the ability to do something. Their class codes in *Cilin* are "Ba06", "Dk14" and "Al03," respectively.

The thesaurus *Cilin* contains 52,716 Chinese word entries. The statistical data obtained shows that the polysemous words account for 14.8% of all the words in *Cilin*. Moreover, the shorter the length of the word, the greater the number of word senses; for example, the percentage of one-character polysemous is 48%, of two-character polysemous is 16% and of others is 9%.

2.1.2 The objective of disambiguation

In the paper, the objective of word sense disambiguation is to select the correct semantic code in *Cilin* for a polysemous word in a particular context. Because the major classes in *Cilin* have a coarse correspondence with the parts of speech of Chinese words, we will constrain ourselves to disambiguating word senses under the same part of speech. The table 1 lists noun, verb and adjective and their corresponding major classes in *Cilin*, respectively.

Table 1. *Parts of speech vs. Major classes.*

Noun	A, B, C, D
Verb	F, G, H, I, J, K
Adjective	E

2.1.3 The processing of unknown words

Unknown words are defined as those that are not included in *Cilin*. We list all of these words in a very large corpus and then assign their semantic codes in *Cili*n by making use of their definition texts in the Chinese dictionary *Xiandai Hanyu Cidian* and tools which have been developed for semi-automatic word sense tagging.

2.2 The vector space representation of *Cilin*

"You shall know a word by the company it keeps" [Firth, 1957]. The neighboring words provide strong and consistent clues to the correct sense of a target word in a given context. This implies that word sense can be generally judged by its context; therefore, we choose to model word sense based on context. We also choose to represent the context as a vector.

2.2.1 Word sense vector

Definition 1: word sense vector

Suppose W is a word and YL(W) is the set of all its senses, $YL(W)=\{s_1, s_2, .., s_{YN}\}$, where YN is the number of senses of W (where W is a monosemous word when YN=1). The vector of sense s_i of word W can be calculated using a corpus.

Suppose sense s_i of word W occurs k times in the corpus, i.e. $W_{si,1}, W_{si,2}, \dots, W_{si,k}$. Their neighboring words within a distance of d words are listed as follows, respectively:

$$a_{1,-d}, a_{1,(-d-1})), \dots, a_{1,-1}\ W_{si,1}, a_{1,1}, a_{1,2}, \dots, a_{1,d}$$

$$a_{2,-d}, a_{2,(-d-1)}, \dots, a_{2,-1}\ W_{si,2}, a_{2,1}, a_{2,2}, \dots, a_{2,d}$$

$$\vdots$$

$$a_{k,-d}, a_{k,-(d-1)}, \dots, a_{k,-1}\ W_{si,k}, a_{k,1}, a_{k,2}, \dots, a_{k,d}.$$

Letting VW(W_{si}) stand for the word sense vector of sense s_i of word W , it can be calculated by using the following formulae:

$$VW(W_{si}) = \left\{ p(W_{si}, X_j, \middle| 1 \le j \le n, x_j \in A \right\} \tag{1}$$

$$A = \left\{ x_1, x_2, \ldots x_n \middle| x_i \, is \; a \; content \; word \; co - occurring \; with \; W \right\}, \tag{2}$$

$$p(W_{si}, x_j) = \frac{c(W_{si}, x_j)}{c(x_j)} \tag{3}$$

Here, x_j is a co-occurring context word of W_{si} within a distance of d words; x_j occurrs c(x_j) in the corpus. c(W_{si}, x_j) and p(W_{si}, x_j) are the number of co-occurrences and the probability of W_{si} and x_j, respectively. d (the value of which was set to 7 in our experiment) is the length of the context window. As can be seen from the above definitions, the word sense vectors of monosemous words can be calculated straightforwardly using a very large corpus. All word sense vectors, which have their own different feature sets in the initial stage, eventually have the same fixed dimensional vector after standardization. Table 2 shows the standardization results of the feature sets of words w_1 and w_2.

Table 2. *Standardization results of the feature sets*

Word	The feature set before standardization	The feature set after standardization
W_1	{阿，阿哥，阿拉伯，阿拉伯人，阿婆, ... }	{阿, 阿哥, 阿拉伯, 阿拉伯人, 阿婆, 阿姨, ...}
W_2	{阿, 阿拉伯, 阿拉伯人, 阿姨, ...}	{阿, 阿哥, 阿拉伯, 阿拉伯人, 阿婆, 阿姨, ...}

2.2.2 Two hypotheses regarding the semantic space

So far, each word sense vector is represented as a multidimensional real-valued vector, and the semantic space is thus mapped into a vector space, where each word sense corresponds to a point in it. Word sense vectors in the space can not be distributed in a uniform way. Miller and Chales [1991] in their study found evidence that human subjects determine the semantic similarity of words from the similarity of the context in which they are used. Therefore, we get Hypothesis 1:

Hypothesis 1

If the meanings of two words are similar, then their contexts are similar.

Thus, if a word meaning is represented by a word sense vector, then the more similar the meaning of the two words, the smaller the distance between the two word sense vectors. Here, we use a strong hypothesis, i.e. Hypothesis 2:

Hypothesis 2:

Word sense vectors formed by the same or similar meanings construct a coherent clustering in the vector space.

The reliability of Hypothesis 2 can be verified by comparing the words in a semantic class provided by a thesaurus with the words clustered by word sense vectors.

1) Problem description

Let ClassA and ClassB be two semantic classes codes in *Cilin*, and let G(ClassA) and G(ClassB) be two sets composed by all the monosemous words in the these classes. e.g.,

$G(ClassA)=\{WA_1,WA_2,...,WA_{Am}\}$ and $G(ClassB)=\{WB_1,WB_2,...,WB_{Bn}\}$.

Let $C = G(ClassA) \cup G(ClassB)$.

Then, the word sense vectors $VW(WA_1),VW(WA_2),..., VW(WA_{Am})$ and $VW(WB_1), VW(WB_2),..., VW(VW_{Bn})$ can be calculated based on the definition of a word sense vector using a very large corpus. Now, cluster these words with these vectors into two classes, G_1 and G_2, satisfying the conditions

$C=G_1 \cup G_2$ and $G_1 \cap G_2= \Phi$.

The conclusion that Hypothesis 2 is reasonable can be obtained if G_1 and G_2 are similar to G(ClassA) and G(ClassB).

2) Clustering algorithm

A bottom-up clustering algorithm is introduced by using monosemous word sense vectors in set C.

As can be seen from the definition of a word sense vector, if a word in set C occurs few times in the corpus, then its vector does not preciously indicate its correct position in the semantic space. Therefore, the words which occur frequently in the corpus are clustered first according to the distance between their word sense vectors and the centroids of the two classes when sets G_1 and G_2 are changed. The results of experiments show that the clustering result is better when the words which occur more than 100 times are clustered first. In the end, the remaining words are classified according to their distances from the centroids of the two different classes. The distance between two word sense vectors is defined by a cosine metric as follows:

$$dist(VW(w_1), VW(w_2)) = 1 - \cos(VW(w_1), VW(w_2)) \quad , \tag{4}$$

$$\cos(VW(w_1), VW(w_2)) = \sum_{1 \le i \le k} x_i y_i \Big/ \sqrt{\sum_{1 \le i \le k} x_i^2 * \sum_{1 \le i \le k} y_i^2} \quad , \tag{5}$$

where x_i and y_i are the components of $VW(w_1)$ and $VW(w_2)$, respectively, k is the component number in vector $VW(w_1)$ and vector $VW(w_2)$.

***Clustering algorithm*:**

1. Initialization: Find two centroids of G_1 and G_2:

$$dist(VW(c_1), VW(c_2)) = \max_{i,j} dist(VW(w_i), VW(w_j)) \quad , \tag{6}$$

where w_i ,w_j C and $w_i \neq w_j$.

2. G_1={c_1}, G_2={c2}.

3. Repeat:

- calculate the two centroid vectors cent(G_1) and cent(G_2) of two classes:

$$cent(G_1) = \{\frac{1}{N(G_1)} \sum_{w \in G_1} VW(w)\} \quad , \tag{7}$$

where, $N(G_1)$ is the number of words in set G_1

and the formula of cent(G_2) is the same as that of cent(G_1) 。

```
• find $w_1$ and $w_2$ in the remaining words satisfying

        $w1, w2 \in C\ and\ w_1, w_2 \notin G_1 \cup G_2$,

        $w_1 = \underset{w_i}{\arg}\,\underset{i}{\min}\, dist(VW(w_i), VW(G_1))$,

        $w_2 = \underset{w_i}{\arg}\,\underset{i}{\min}\, dist(VW(w_i), VW(G_2))$·

• if $w_1 \neq w_2$ , then:

        $G_1=G_1 \cup \{w_1\}, G_2=G_2 \cup \{w_2\}$,

  else:

        do not cluster w1 and w2

  until all the words occurring more than 100 times have been clustered.
```

```
4. Classify the low frequency words
   for each word $w \in C$ but $w \notin G1 \cup G2$:

       if $dist(VW(w),VW(G_1))<dist(VW(w),VW(G_2))$, then:

           $G1=G1 \cup \{w\}$,

       else:

           $G2=G2 \cup \{w\}$.
```

5. End of the procedure.

In step 3, if there are two words which have the same minimal distance to $G_1(G_2)$, then the word whose distance is calculated first is selected.

3) Clustering results

We selected many class pairs from the *Cilin* thesaurus and ran the algorithm on a 72 MB corpus. The consistency between the *Cilin* categories and the derived clusters is measured based on c, which is defined as the ratio of the number of correctly classified words to the total number of words in the two classes.

Some clustering results are listed in table 3.

Table 3. *Results of Clustering*

CP	SSN	CSN	C1	C2	C3
Hc11/Hc03	17/18	6005/5438	100.00%	91.66%	77.14%
Ba06/Da19	16/16	3415/3954	100.00%	100.00%	100.00%
Hc11/Hi03	17/18	6006/6165	90.91%	82.00%	69.28%
Aa03/Ae07	15/20	6800/6735	90.00%	90.00%	82.48%
Di10/Di08	27/28	12017/11531	87.50%	85.00%	75.47%
Ed29/Ed11	15/17	3543/4054	100.00%	100.00%	78.57%
Ed16/Ef08	14/17	2599/2656	100.00%	88.89%	81.25%
Gb15/Hj20	7/6	2303/2003	100.00%	85.79%	84.61%
Average C			96.05%	90.42%	81.10%

Here, CP represents the class pairs selected from the thesaurus *Cilin*. Column SSN indicates the number of monosemous words in each class, respectively, and column CSN gives the total number of occurrences of these monosemous words in the corpus. The last three columns, C1, C2 and C3, each show the value of c when the algorithm is limited to only cluster words occurring more than 100 times, 50 times and 0 times, respectively.

From the above table, we can draw the following conclusions:

1. When the word frequency is more than 100 times, the average value of *c* is 96.05%. Even when the frequency requirement is lowered to 50 times, the average value of *c* is 90.42%. This supports the rationality of Hypothesis 2.

2. The more often a word appears in the corpus, the better the clustering result will be.

3. The clustering results for those pairs that have different major classes are better than those for those pairs that have the same major class.

2.2.3 Semantic class vector

There always exist some monosemous words in a *Cilin* category, whose context vectors can be straightforwardly calculated using a very large corpus. A census shows that the average proportion of monosemous words in a semantic class is greater than 60%. Therefore, we can use the centroid of monosemous word sense vectors to represent the co-occurrence probability of the class. This centroid vector is defined as the semantic class vector.

Definition 2: Semantic class vector

Suppose Synset is a semantic class code in the thesaurus *CiLin*, and that G(Synset) is the set of words contained in the class Synset. Let G(Synset)= A ∪ B, where A is the set of all the monosemous words while B is the set of all the polysemous words in the given class Synset. We define the class semantic vector of Synset, SV(Synset), as a function of set A:

$$SV(Synset) = \{\frac{1}{N(A)} \sum_{w \in A} VW(w)\} , \qquad (8)$$

Here, N(A) is the number of words in set A. In fact, the semantic class vector is an average vector over the word sense vectors of all the monosemous words included in the class.

2.2.4 Constructing *Cilin's* semantic space using a very large corpus

So far, the semantic space has been converted into a vector space, in which the distribution of vectors displays a coherent pattern of clustering; that is, the more similar the meaning of two words, the closer their word sense vectors in the space. All the similar words form a coherent class which can be represented by the centroid of its constituent word sense vectors. Above this level, similar class vectors can also form larger classes, which have less agreement than do lower-level classes. Thus, the semantic space is viewed as having a hierarchical structure.

In the following, a word sense disambiguation method based on this semantic space is presented. It can be divided into two main steps. First, semantic class vectors are calculated using the distribution of monosemous words in a large-scale corpus. Second, a given instance of a polysemous word is classified by comparing the vector representation of its actual context with the semantic class vectors of the classes corresponding to each of its senses. According to the metrics defined as formula (4) and formula (5), the sense category, whose corresponding semantic class vector is nearest to the actual context, is determined as the correct category of the given polysemous word.

3. Word sense disambiguation based on structural semantic space

3.1 Main ideas

1) Dynamic word sense disambiguation procedure

As defined above, the semantic class vectors corresponding to a given target word are simply calculated as the centroids, of these vectors, without regard to the relations between them. In order to obtain a more powerful model, we add procedures for feature selection and feature weighting, which are sensitive to the relations between the semantic classes to which each specific target word can belong. The features collected are only the words that are helpful for determining the meaning of the target word. The weight values of a feature in different semantic classes indicate how much they support these semantic classes.

2) Robust word sense disambiguation model

Now that the distributions of the words in the lower level classes have more agreement, we can start the procedure of word sense disambiguation in the lowest level classes (i.e. the fourth level class). If the target word can not be disambiguated in this level, the procedure can be restarted in a higher level. In this way, the model becomes more robust. In order to avoid disagreement between words in a high level class, we constrain the third level to be the highest level.

3.2 Word sense disambiguation based on semantic space

3.2.1 Word sense disambiguation method

The *Cilin* thesaurus and a 72MB news corpus are used to construct the semantic space. The input of word sense disambiguation is a polysemous target word, along with its context within a distance of *d* words, and the output is a semantic class code for the target word determined by the model .

Based on the above ideas, figure 1 shows the word sense disambiguation model. The sub-models included in this figure will be explained in the following sections.

3.2.2 Feature collection

The purpose of this step is to collect from the corpus those context words that are particularly helpful for word sense disambiguation. Some functional words, such as prepositions and conjunctions, are unlikely to have significant semantic relations with the target word and, therefore, contribute little to the disambiguation task. Therefore, we pre-constrain the candidate feature words to be content words.

The procedure forms a co-occurrence probability matrix $A_{m^* n}$, where the rows represent the lowest level classes, the columns represent the words co-occurring with monosemous words in the classes, and a_{ij} is the co-occurring probability of C_i and W_j in the corpus.

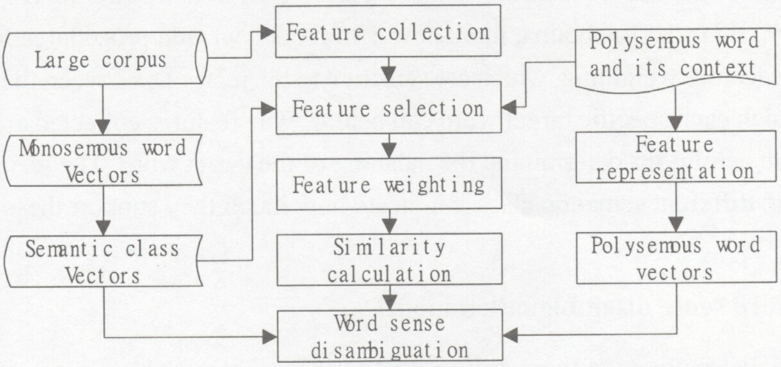

Figure 1 *The word sense disambiguation model based on semantic space*

3.2.3 Feature selection

Using feature selection has two advantages. One is that it can select words that are helpful for disambiguating the current polysemous word; the other is that it can reduce the number of words contained in the semantic class vectors of the target word, so as to promote the efficiency of word sense disambiguation. After this procedure is conducted, the words that co-occur with the semantic class vectors in a uniform way and that rarely co-occur with these vectors are discarded.

Entropy is used as a metric to select feature words. Suppose W is the target word and YL(W) is the set of its possible class codes, $YL(W)=\{C_1, C_2, ..., C_{YN}\}$. YN is the number of class codes to which the instances of the polysemous word can belong. The feature words are selected according to the algorithm as follows.

Feature selection Algorithm

```
Begin
    FV={}, FN=0;
          where FV is the set of feature words and FN is the number of features in FV.
    For each candidate feature word $W_i$, do this:
        calculate its entropy value and store it in H, where
```

$$H(W_i) = -\sum_{j=1}^{YN} a_{ij} \log a_{ij} \,, \tag{9}$$

```
If $H(W_i) < yz_1$ and $c(c_j, W_i) > yz_2$ ,
    then
        $W_i$ is a feature
        $FV=FV \cup \{W_i\}$, $FN=FN+1$.
End
```

yz_1 is the threshold of entropy. A candidate feature can be selected as a feature if its entropy value is less than this threshold. In general, the value of yz_1 is set to 1.2. yz_2 is the threshold of the number of occurrences. The number of occurrence of the collected feature should be greater than yz_2, generally, yz_2=4.

3.2.4 Feature weighting

Feature weighting calculates the degree of support for different semantic classes of the target word provided by each feature word. For a given feature word, if its weighting value for one class of the target word is high, then this feature word is an indication word of this class of the target word.

If f $\in$FV, then its weighted value $WM_i(f)$ for class C_i can be calculated using the following formula.

$$WM_i(f) = \frac{p(f, C_i)}{\sqrt{\sum_{j=1}^{YN} p^2(f, C_j)}} \,. \tag{10}$$

Here, $p(f, C_j)$ is the co-occurrence probability of feature word f and the class C_j in the corpus. Thus, for each class belonging to YL(W), a weighted feature vector can be calculated.

3.2.5 Word sense disambiguation procedure

The similarities between the vector extracted from the actual context of the given target word and the weighted feature vectors provided by the classes containing the target word are calculated. The similarity is defined as follows:

$$similarity(V(W),V(C)) = \cos(V(W),V(C)) \, . \qquad (11)$$

Here, V(W) is the vector formed by the current context of the target word, and V(C) is the weighted feature vector. The calculation of $\cos(V(W),V(C))$ is the same as formula (5).

Word sense disambiguation algorithm

```
1. Form the feature vector $V(W)=\{b(f_1),b(f_2), \cdots , b(f_{FN})\}$ of the polysemous word
   according to its context, where

      $b(f_i) = \begin{cases} 1 \;\; if \;\; f_i \;\; occurring \;\; in \; the \; context \\ 0 \;\; otherwise. \end{cases}$      (12)

2. For $C_j \in YL(W)$  $(j=1, 2, ..., yn)$ ,
      calculate $similarity(V(W), V(C_j))$

3. Let $R = \arg \max_{c_i \in YL(W)} similarity(V(W),V(C_i))$

4. If $similarity(V(W), V(R)) \neq 0$,
      then determine the word sense of W is R
      else
      raise the class level to a higher level $YL(W)=\{C1', C2', ... , Cn'\}$ ;
      get a new YL(W) for the polysemous word ;
      go to step 2;
      until the sense code of W is determined.

5. End
```

4. Experiments and results

The corpus used in the experiments was derived from the news texts *in The People's Daily*. The corpus consists of twenty-million word tokens. The corpus was used to construct a semantic space and to perform the tests. The objectives of the tests were to determine whether the word sense disambiguation algorithm provided here is effective and whether the semantic space outlined by monosemous words in *Cilin* is helpful for polysemous word disambiguation. To accomplish the first objective, we ran a disambiguation test using pseudo-words, which are defined in 4.1.1. To reach the second goal, we ran a disambiguation test using real polysemous words.

4.1 Pseudo-word sense disambiguation

4.1.1 Definition

Pseudo-word: A pseudo-word is defined as the conflation of two or more different monosemous words into a single artificial word, which can thus take on any of the senses of the original words. For example, " 收 (buy)/ 修改 (revise)"is a pseudo-word whose semantic code set is "He03/Hg18."

The pseudo-word sense disambiguation experiment [Schutz, 1992, Gale, Church, Yarowsky, 1992] is a simple approach to evaluating the various methods used for word sense disambiguation. This experiment was used here to verify the validity of our disambiguation method.

4.1.2 Testing method

The test was composed of a closed test and an open test. The data for the closed test came from the training corpus while the data for the open test were selected from other corpora of the same genre as the training corpus. Suppose a is the number of tested samples and *b* is the number of samples that have been disambiguated correctly by the model. Then we define the precision p as $p=b/a \times 100\%$.

4.1.3 Results

The results for five pseudo-words are shown in table 4. In order to give an indication of the generality of the method, the table also gives the number of occurrences in the corpus for the corresponding lowest level classes. The precision P was used to test the performance of the disambiguation model. It is the proportion of the number of words correctly tagged by the model to the total number of testing words. P1 and P2 are the results of the closed test and open test, respectively.

Table 4. *The results of pseudo-word sense disambiguation.*

PW	CT	CN	P1	P2
權利 / 事故	Di21/Da01	5088/2187	97.5%	94%
草案 / 責任	Dk17/Di22	4177/4010	89.5%	85%%
預算 / 預賽	Hj29/Hh07	7914/4450	95%	89%
收購 / 修改	He03/H818	2383/1135	93%	93%
頒布 / 參与	Hc11/Hi23	3062/1472	92.5%	87%
Average P			93.5%	89.6%

In the table, column PW shows the pseudo-words, and column CT gives their corresponding semantic codes. Column CN presents the number of occurrences for words in each lowest level class.

Based on the above table, we can make the following claims:

1. Because there is high precision of pseudo-word sense disambiguation, the feature selection and feature weighting methods presented in this paper are reasonable.

2. Though some pseudo-words, such as " 頒布 / 參与 ," occur very few times in the corpus, their classes appear many times. Therefore, the precision is still satisfactory. This evidence supports the claim that statistical data calculated using monosemous words in a corpus can reflect the general distribution of the space.

4.2 Real polysemous word disambiguation

Since the data selected from monosemous words is exclusive to real polysemaous words, there is only an open test was conducted here. The calculation of the precision P is the same as that for pseudo-sense words.

Table 5. *Results of polysemous word sense disambiguation.*

PW	CT	TN	CN	P
材料	Dk17/Ba06/Al03	971	1913/1021/422	81.7%
改	Ih02/Hg18/Hj66	2847	1315/1135/309	70.6%
表現	Jd06/Di20/Hj59	754	1323/1500/20	68.9%
發表	Hc11/Hi14/Jd03	2973	5761/2943/214	73.4%
健康	Ed43/Eb37	902	1056/101	70.1%
Average P				72.94%

Table 5 shows some testing results for five polysemous words. It confirms that the idea of constructing a semantic space using the distribution of monosemous word in a corpus is reasonable. This method not only has high precision for word sense disambiguation, but also can deal with a large number of polysemous content words.

Based on analysis of the experimental results, we can make the following claims:

1. When the topic and usage of a polysemous word are consistent with the monosemous words in the same class, the precision is high; when the topic and usage are inconsistent, the precision is low. The reason is that the distribution of a class in the corpus reflects the common distribution of the class, and not the usage specific to individual words.

2. When the words in the context vector are widely used, the result is often incorrect.

3. When the sample has a number of feature words whose weighting values are inconsistent, we often get wrong results.

5. How other methods differ

1) How Schutze's method [1998] differs: The differences lie in that: 1) Schutze constructs word vectors using the context of all the appearances of a target word in the corpus while we only use the monosemous words to construct word sense vectors in a very large corpus. Thus, each word sense vector corresponds to a word sense point in semantic space. 2) Schutze uses the EM algorithm for clustering while we use an existing semantic system provided by a thesaurus, the rationality for which has been verified by a clustering method. 3) Schutze's method does not conducted automatically map the sense representation derived from the system onto the more conventional word sense found in the dictionaries.

2) How Yarowsky's method [1992] differs: Yarowsky uses a thesaurus to collect training materials. He points out that noise will be a problem when a class contains one very frequent polysemous word which dominates the training set. In this paper, we have alleviated this problem by only using the distribution of the monosemous words in a corpus. Furthermore, our method can be trained using different types of corpora to adapt to word sense disambiguation for different domains.

3) How Millar's method [1998] differs: Millar's idea, in which the monosemous words are used in word sense disambiguation, is similar to the method proposed in this paper. The greatest difference between the methods is that we view a semantic system as a structural vector space, where each word sense corresponds to a point in the space, and the vectors in the space cluster coherently. Furthermore, we also can collect statistical data for the lowest-level sense class vectors from a very large-scale corpus only once, instead of sampling and retrieving example sentences from a corpus for each different

polysemous word. This change can greatly ease the implementation of our method.

6. Conclusions and future research

The method we have presented in this paper has the following characteristics.

(1) We use a multidimensional real-valued vector space to represent the semantic space of a thesaurus, based on verification of the plausibility of this approach.

(2) Based on the above characteristic, we convert the word sense disambiguation problem into a problem of locating the position of a polysemous word in a particular context in the semantic space formed by the all monosemous word in the corpus.

(3) Because the average percentage of monosemous words occurring in each semantic class is above 60%, the method proposed here can disambiguate a large number of content words.

(4) In principle, the method can also be applied to other languages.

Our future research work will include the following three points.

(1) How can new examples be automatically added when monosemous words occur only a few times in the corpus?

(2) In the process of using the thesaurus *Cilin*, some errors in the classification of some words occur. Therefore it will be a part of our further work to correct the misclassification of words and to add new words to improve the applicability of the *Cilin* thesaurus.

(3) The kinds of words that are classified well by the model and those which still are not need to be investigated. Based on this research, some improved methods will be presented for better processing of these words.

References:

1. Kesk, Michael. 1986. Automated sense disambiguation using machine-readable dictionaries: How to tell a pine cone from an ice cream cone. In *Proceedings of the 1986 SIGDOC conference*, Page 24-26.

2. Wilks, Yorick A. and Dan Fass. 1990. Preference semantics: A family histroy. *Report MCCS-90-194*, Computing Research Laboratory, New Mexico State University, Las Cruces, NM.

3. Yarowsky David. 1992. Word sense disambiguation using statistical model of Roget's cat-

egories trained on large corpora. *Proceedings of the 14th International Conference on Computational Linguistics*, COLING'92, page 454-460, Nantes, France, August.

4. Firth, J. R. 1951. Modes of meaning. *Papers in Linguistics 1934-51*, pages 190-215, Oxford University Press, Oxford, UK.

5. Miller, George A. and Walter G. Charles. 1991. Contextual correlates of semantic similarity. *Language and Cognitive Processes*. 6(1):1-28.

6. Nancy Ide, Jean Veronis. Computational Linguistics Special Issue on Word Sense Disambiguation. *Computational Linguistics*, 1998, 24(1), 1-42.

7. Hinrich Schutze, Automatic Word Sense Discrimination. *Computational Linguistics, 1998, 24(1)*, 97-123.

8. Hinrich Schutze, Context Space. In Robert Goldman, Peter Norvig, Eugene Charniak, and Bill Gall, Editors, W*orking Notes of AAAI Fall Symposium on Probabilistic Approaches to Natural Language*, p113-120, AAAI Press, Menlo Park, CA, 1992.

9. Yarowsky, David, Word-sense disambiguation using statistical models of Roget's categories trained on large corpora. In *Proceedings of COLING-92*, Nantes, France.

10. 梅家駒，竺一鳴，高蘊琦等《同義詞詞林》，上海：上海辭海出版社， 1983.

11. 賈彥德，《漢語語義學》，北京：北京大學出版社， 1992.

12. 朱曼殊，《心理語言學》，上海：華東師范大學出版社， 1990.

13. 中國社會科學院語言研究所詞典編輯室，《漢語漢語詞典》，北京：商解印書館，1994.