Word Sense Disambiguation Based on Structured Semantic Space*

Ji Donghong Huang Changning

Department of Computer Science
Tsinghua University
Beijing, 100084, P. R. China
Email: jdh@s1000e.cs.tsinghua.edu.cn
hcn@mail.tsinghua.edu.cn

Abstract

In this paper, we propose a framework, *structured semantic space*, as a foundation for word sense disambiguation tasks, and present a strategy to identify the correct sense of a word in some context based on the space. The semantic space is a set of multidimensional real-valued vectors, which formally describe the contexts of words. Instead of *locating* all word senses in the space, we only make use of mono-sense words to *outline* it. We design a merging procedure to establish the *dendrogram structure* of the space and give an heuristic algorithm to find the nodes (*sense clusters*) corresponding with sets of similar senses in the dendrogram. Given a word in a particular context, the context would *activate* some clusters in the dendrogram, based on its similarity with the contexts of the words in the clusters, then the correct sense of the word could be determined by comparing its definitions with those of the words in the clusters.

1. Introduction

Word sense disambiguation has long been one of the major concerns in natural language processing area (e.g., Bruce et al., 1994; Choueka et al., 1985; Gale et al., 1993; McRoy, 1992; Yarowsky 1992, 1994, 1995), whose aim is to identify the correct sense of a word in a particular context, among all of its senses defined in a dictionary or a thesaurus. Undoubtedly, effective disambiguation techniques are of great use in many natural language processing tasks, e.g., machine translation and information retrieving (Allen, 1995; Ng and Lee, 1996; Resnik, 1995), etc.

Previous strategies for word sense disambiguation mainly fall into two categories: statistics-based method and exemplar-based method. Statistics-based method often requires large-scale corpora (e.g., Hirst, 1987; Luk, 1995), sense-tagging or not, monolingual or aligned bilingual, as training data to specify significant clues for each word sense. The method generally suffers from the problem of *data sparseness*. Moreover, huge corpora, especially sense-tagged or aligned ones, are not generally available in all domains for all languages.

Exemplar-based method makes use of typical contexts (exemplars) of a word sense, e.g., verb-noun collocations or adjective-noun collocations, and identifies the correct sense of a word in a particular context by comparing the context with the exemplars (Ng and Lee, 1996). Recently, some kinds of learning techniques have been applied to cumulatively acquire exemplars form large corpora (Yarowsky, 1994, 1995). But ideal resources from which to learn exemplars are not generally available for any languages. Moreover, the effectiveness of this method on disambiguating words in large-scale corpora into fine-grained sense distinctions needs to be further investigated (Ng and Lee, 1996).

* The work is supported by National Science Foundation of China.

A common assumption held by both approaches is that neighboring words provide strong and consistent clues for the correct sense of a target word in some context. In this paper, we also hold the same assumption, but start from a different point. We see the senses of all words in a particular language as forming a space, which we call *semantic space*, for any word of the language, each of its senses is regarded as a *point* in the space. So the task of disambiguating a word in a particular context is to locate an appropriate point in the space based on the context.

Now that word senses can be generally suggested by their distributional contexts, we model senses with their contexts. In this paper, we formalize the contexts as a kind of multidimensional real-valued vectors, so the semantic space can be seen as a vector space. The similar idea about representing contexts with vectors has been proposed by Schuetze (1993), but what his work focuses on is the contexts of words, while what we concern is the contexts of word senses. Furthermore, his formulation of contexts is based on word frequencies, while we formalize them with semantic codes given in a thesaurus and their *salience* with respect to senses.

It seems that we should first have a large-scale sense-tagged corpus in order to build semantic space, but establishing such a corpus is obviously too time-consuming. To simplify it, we only try to *outline* the semantic space by locating the mono-sense words in the space, rather than build it completely by *spotting* all word senses in the space.

Now that we don't try to specify all word senses in the semantic space, for a word in a particular context, it may be the case that we cannot directly spot its correct sense in the space, because the space may not contain the sense at all. But we could locate some senses in the space which are similar with it according to their contexts, and based on their definitions given in a dictionary, we could make out the correct sense of the word in the context.

In our implementation, we first build the semantic space based on the contexts of the mono-sense words, and structure the senses in the space as a dendrogram, which we call *structured semantic space*. Then we make use of an heuristic method to determine some nodes in the dendrogram which correspond with sets of similar senses, which we call *sense clusters*. Finally, given a target word in a particular context, some *clusters* in the dendrogram can be *activated* by the context, then we can make use of the definitions of the target word and the words[1] in the clusters to determine its correct sense in the context.

The remainder of the paper is organized as follows: Section 2 defines the notion of semantic space, and discuss how to outline it by establishing the context vectors for mono-sense words. Section 3 examines the structure of the semantic space, and introduces algorithms to merge the senses into a dendrogram and specify the nodes in it which correspond with sets of similar senses. Section 4 discusses the disambiguation procedure based on the contexts. Section 5 describes some experiments and their results. Section 6 presents some conclusions and discusses the future work

2 Semantic Space

In general, a word may have several senses and may appear in several different kinds of contexts. From a point of empirical view, we suppose that each sense of a word is corresponded with a particular kind of context it appears, and the similarity between word senses can be measured by their corresponding contexts. For a particular kind of language, we regard its *semantic space* as the set of all word senses of the language, with *similarity* relation between them.

Now that word senses are in accordance with their contexts, we use the contexts to model word senses. Due to the unavailability of large-scale

[1] Because the senses in the semantic space are of mono-sense words, we don't distinguish "words" from "senses" strictly here.

sense-tagged corpus, we try to *outline* the semantic space by only taking into consideration the mono-sense words, instead of *locating* all word senses in the space.

In order to formally represent word senses, we formalize the notion of *context* as multidimensional real-valued vectors. For any word, we first annotate its neighboring words within certain distances in the corpus with all of their semantic codes in a thesaurus respectively, then make use of such codes and their *salience* with respect to the word to formalize its contexts. Suppose w is a mono-sense word, and there are n occurrences of the word in a corpus, i.e., $w_1, w_2, \ldots, w_n$, (1) lists their neighbor words within d word distances respectively.

(1)

$$\begin{array}{llll} a_{1,-d}, a_{1,-(d-1)}, \ldots, a_{1,-1} & w_1 & a_{1,1}, a_{1,2}, \ldots, a_{1,d} \\ a_{2,-d}, a_{2,-(d-1)}, \ldots, a_{2,-1} & w_2 & a_{2,1}, a_{2,2}, \ldots, a_{2,d} \\ & \vdots & \\ & \vdots & \\ a_{n,-d}, a_{n,-(d-1)}, \ldots, a_{n,-1} & w_n & a_{n,1}, a_{n,2}, \ldots, a_{n,d} \end{array}$$

Suppose C_T is the set of all the semantic codes defined in a thesaurus, for any occurrence w_i, $1 \le i \le n$, let NC_i be the set of all the semantic codes of its neighboring words which are given in the thesaurus, for any $c \in C_T$, we define its *salience* with respect to w, denoted as $Sal(c, w)$, as (2).

$$(2)\quad Sal(c, w)=\frac{\left|\{w_i | c \in NC_i\}\right|}{n}$$

So we can build a *context vector* for w as (3), denoted as cv_w, whose dimension is $|C_T|$.

$$(3)\ cv_w=<Sal(c_1, w), Sal(c_2, w), \ldots, Sal(c_k, w)>$$

where $k=|C_T|$.

When building the semantic space for Chinese language, we make use of the following resources, i) Xiandai Hanyu Cidian(1978), a Modern Chinese Dictionary, ii) Tongyici Cilin(Mei et al, 1983), a Chinese thesaurus, iii) a Chinese corpus consisting of 80 million Chinese characters. In the Chinese dictionary, 37,824 words have only one sense, among which only 27,034 words occur in the corpus, we select 15,000 most frequent mono-sense words in the corpus to build the semantic space for Chinese. In the Chinese thesaurus, the words are divided into 12 major classes, 94 medium classes and 1428 minor classes respectively, and each class is given a semantic code, we select the semantic codes for the minor classes to formalize the contexts of the words. So $k=|C_T|=1428$.

3. Structure of Semantic Space

Due to the similarity/dissimilarity relation between word senses, those in the semantic space cannot be distributed in an uniform way. We suppose that the senses form some *clusters*, and the senses in each cluster are similar with each other. In order to make out the clusters, we first construct a dendrogram of the senses based on their similarity, then make use of an heuristic strategy to select some appropriate nodes in the dendrogram which most likely correspond with the clusters.

Now that word senses occur in accordance with their contexts, we measure their similarity /dissimilarity by their contexts. For any two senses $s_1, s_2 \in S$, let $cv_1=(x_1\ x_2\ \ldots\ x_k)$, $cv_2=(y_1\ y_2\ \ldots\ y_k)$ be their context vectors respectively, we define the *distance* between s_1 and s_2, denoted as $dis(s_1, s_2)$, based on the cosine of the angle between the two vectors.

$$(4)\quad dis(s_1, s_2)=1-cos(cv_1, cv_2)^2$$

Obviously, $dis(s_1, s_2)$ is a normalized coefficient: its value ranges from 0 to 1.

Suppose S is the set of the mono-senses in the

---

[2] $$cos(cv_1, cv_2)=\sum_{1\le i\le k} x_i y_i \Big/ \sqrt{\sum_{1\le i\le k} x_i^2 \bullet \sum_{1\le i\le k} y_i^2}$$

semantic space, for any sense $s_i \in S$, we create a *preliminary node* d_i, and let $|d_i|=1$, which denotes the number of senses related with d_i. Suppose D be the set of all *preliminary nodes*, the following is the algorithm to construct the dendrogram.

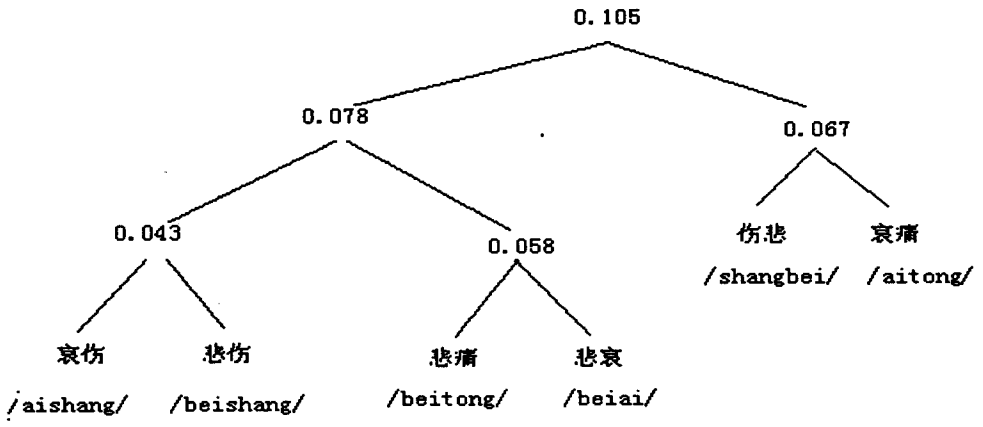

Fig. 1 A subtree of the dendrogram for Chinese mono-sense words

Algorithm 1.

Procedure *Den-construct*(D)

begin

i) select d_1 and d_2 among all in D, whose *distance* is the smallest;

ii) merge d_1 and d_2 into a new node d, and let $|d|=|d_1|+|d_2|$;

iii) remove d_1 and d_2 from D, and put d into D;

iv) compute the context vector of d based on the vectors of d_1 and d_2[3];

v) go to i) until there is only one node;

end;

Obviously, the algorithm is a bottom-up merging procedure. In each step, two closest nodes are selected and merged into a new one. In (n-1)*th* step, where n is the number of word senses in S, a final node is produced. The complexity of the algorithm is $O(n^3)$ when implementing it directly, but can be reduced to $O(n^2)$ by sorting the distances between all nodes in each previous step.

Fig.1 is a sub-tree of the dendrogram we build for Chinese. It contains six mono-sense words, whose English correspondences are *sad*, *sorrowful*, etc. In the sub-tree, we mark each non-preliminary node with the distance between the two merged sub-nodes, which we also refer to as the *weight* of the node.

It can be proved that the distances between the merged nodes in earlier merging steps are smaller than those in later merging steps[4]. According to the similarity/dissimilarity relation between the senses, there should exist a *level* across the dendrogram such that the weights of the nodes above it are bigger, while the weights of the nodes below it smaller, in other words, the ratio between the *mean weight* of the nodes above the level and that of the nodes below the level is the biggest. Furthermore we suppose that the nodes immediately below the level correspond with the clusters of similar senses. So, in

[3] We call $(z_1 z_2 \ldots z_k)$ the context vector of d, where for all i, $1 \le i \le k$, $z_i = (|d_1| \bullet x_i + |d_2| \bullet y_i)/|d|$.

[4] This can be seen from the fact that the context vector of d is a linear composition of the vectors of d_1 and d_2.

order to make out the sense clusters, we only need to determine the level.

Unfortunately, the complexity of determining such a level is exponential to the edges in the dendrogram, which demonstrates that the problem is hard. So we adopt an heuristic strategy to determine an optimal level.

Suppose T is the dendrogram, sub_T is the sub-tree of T, which takes the same root as T, we also use T and sub_T to denote the sets of non-preliminary nodes in T and in sub_T respectively, for any $d \in T$, let $Wei(d)$ be the weight of the node d, we define an *object function*, as (5):

$$(5)\quad Obj(sub\_T) = \frac{\sum_{d \in sub\_T} Wei(d) \Big/ |sub\_T|}{\sum_{d \in (T - sub\_T)} Wei(d) \Big/ |T - sub\_T|}$$

where the numerator is the *mean weight* of the nodes in sub_T, while the denominator is the *mean weight* of the nodes in *T-sub_T*.

In order to specify the sense clusters, we only need to determine a sub-tree of T which makes (5) get its biggest value. We adopt depth-first search strategy to determine the sub-tree. Suppose v_0 is the root of T, for any $v \in T$, we use *v.L and v.R* to denote its two sub-nodes, let T_c be the set of all the nodes corresponding with the sense clusters, we can get T_c by *Clustering*(v_0, *NIL*[5]) calling the following procedure.

```
Algorithm 2
  Clustering(v, sub_T)
    begin
      sub_T← sub_T+{v};
            /* add node v to the subtree*/
      if Obj(sub_T+v.L)>Obj(sub_T)
          then Clustering(v.L, sub_T)
            /* v.L is not a sense cluster */
          else T_c← T_c ∪ {v.L};
            /* v.L is a sense cluster */
      if Obj(sub_T+v.R)>Obj(sub_T)
          then Clustering(v.R, sub_T)
            /* v.R is not a sense cluster */
          else T_c← T_c ∪ {v.R};
            /* v.R is a sense cluster */
  end;
```

The algorithm is a depth-first search procedure. Its complexity is O(n), where n is the number of the leaf nodes in the dendrogram, i.e., the number of the mono-sense words in the semantic space.

When building the dendrogram for the Chinese semantic space, we found 726 sense clusters in the space. The distribution of the senses in the clusters is demonstrated in Table 1.

Number of senses	Number of clusters
[1, 10)	92
[10, 20)	157
[20, 30)	297
[30, 40)	176
[40, 58)	4
	All: 726

Table 1. The distribution of senses in the clusters

4. Disambiguation Procedure

Given a word in some context, we suppose that some clusters in the space can be *activated* by the context, which reflects the fact that the contexts of the clusters are similar with the given context. But the given context may contain much noise, so there may be some activated clusters in which the senses are not similar with the correct sense of the word in the given context. But due to the fact that the given context can suggest the correct sense of the word, there should be clusters, among all activated ones, in which the senses are similar with the correct sense.

[5] NIL is a preliminary value for *sub_T*, which demonstrates the tree includes no nodes.

To make out these clusters, we make use of the definitions of the words in the Modern Chinese Dictionary, and determine the correct sense of the word in the context by measuring the similarity between their definitions.

4.1 Activation

Given a word w in some context, we consider the context as consisting of n words to the left of the word, i.e., w_{-n}, $w_{-(n-1)}$, ..., w_{-1} and n words to the right of the word, i.e., w_1, w_2, w_3,..., w_n. We make use of the semantic codes given in the Chinese thesaurus to

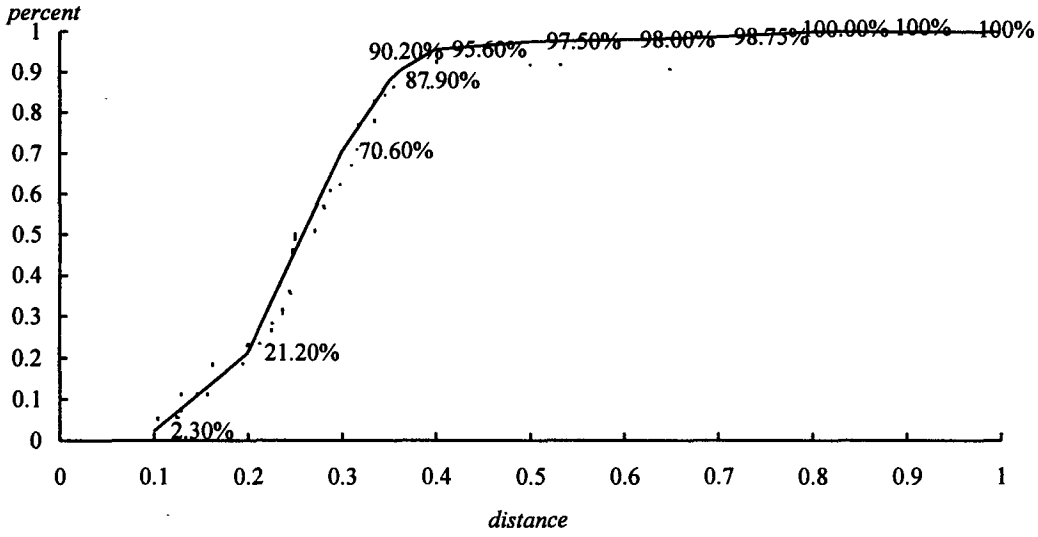

Fig.2 The distribution of $dis_1(clu_w, w)$.

create a context vector to formally model the context,. Suppose NC_w be the set of all semantic codes of the words in the context, then $cv_w=<x_1, x_2, ..., x_k>$, where if $c_i \in NC_w$, then $x_i=1$; otherwise $x_i=0$.

For any cluster *clu* in the space, let cv_{clu} be its context vector, we also define its *distance* from w based on the cosine of the angle between their context vectors as (6).

(6) $dis_1(clu, w)=1-cos(cv_{clu}, cv_w)$

We say *clu* is *activated*, if $dis_1(clu, w) \le d_1$, where d_1 is a threshold. Here we don't define the *activated* cluster as the one which makes $dis_1(clu, w)$ smallest, this is because that the context may contain much noise, and the senses in the cluster which makes $dis_1(clu, w)$ smallest may not be similar with the very sense of the word in the context.

To estimate a reasonable value for d_1, we can compute the distance between the context vector of each mono-sense word occurrence in the corpus and the context vector of the cluster containing the word, then select a reasonable value for d_1 based on these distances as the threshold. Suppose *CLU* is the set of all sense clusters in the space, O is the set of all occurrences of the mono-sense word in the corpus, for any $w \in O$, let clu_w be the sense cluster containing the sense in the space, we compute all distances $dis_1(clu_w, w)$, for all $w \in O$. It should be the case that most values for $dis_1(clu_w, w)$ will be smaller than a threshold, but some will be bigger, even close to 1, this is because most contexts in which the mono-sense words occur would contain meaningful words for the senses, while other contexts contain much noise, and less words, even no words in the contexts are meaningful for the senses.

When estimating the parameter d_1 for the Chinese semantic space, we let n=5, i.e., we only take 5 words to the left or the right of a word as its context. Fig. 2 demonstrates the distribution of the values of $dis_1(clu_w, w)$, where X axle denotes the

distance, and Y axle denotes the percent of the distances whose values are smaller than $x \in [0, 1]$ among all distances. We produce a function $f(x)$ to model the distribution based on commonly used smoothing tools and locate its *inflection point* by setting $f''(x)=0$. Finally we get $x=0.378$, and let it be the threshold d_1.

4.2 Definition-Based Disambiguation

Given a word w in some context c, suppose CLU_w is the set of all the clusters in the semantic space *activated* by the context, the problem is to determine the correct sense of the word in the context, among all of its senses defined in the modern Chinese dictionary.

The activation of the clusters in CLU_w by the context c demonstrates that c is similar with the contexts of the clusters in CLU_w, so there should be at least one cluster in CLU_w, in which the senses are similar with the correct sesne of w in c. On the other hand, now that the senses in a cluster are similar in meaning, their definitions in the dictionary should contain similar words, which can be characterized as holding the same semantic codes in the thesaurus. So the definitions of all the words in the clusters contain strong and meaningful information about the very sense of the word in the context.

We first construct two *definition vectors* to model the definitions of all the words in a cluster and the definitions of w based on the semantic codes of the definition words[6], then determine the sense of w in the context by measuring the similarity between each definition of w and the definitions of all the words in a cluster.

For any $clu \in CLU_w$, suppose $clu=\{w_i | 1 \le i \le n\}$, let C_i be the set of all semantic codes of all the words in w_i's definition, C_T be defined as above, i.e., the set of all the semantic codes in the thesaurus, for any $c \in C_T$, we define its *salience* with respect to *clu*, denoted as *sal(c, clu)*, as (7).

$$(7)\quad sal(c, clu)=\frac{|\{w_i | c \in C_i\}|}{n}$$

We call (8) *definition vector* of *clu*, denoted as dv_{clu}.

$$(8)\quad dv_{clu}= < sal(c_1, clu), sal(c_2, clu), ..., sal(c_k, clu) >$$

Suppose S_w is the set of w's senses defined in the dictionary, for any sense $s \in S_w$, let C_s be the set of all the semantic codes of its definition words, we call $dv_s=<x_1, x_2, ..., x_k>$ *definition vector* of s, where for all i, if $c_i \in C_s$, $x_i=1$; otherwise $x_i=0$.

We define the distance between an activated cluster in the semantic space and the sense of a word as (9) again in terms of the cosine of the angle between their *definition vectors*.

$$(9)\quad dis_2(clu, s)=1-cos(dv_{clu}, dv_s)$$

Intuitively the distance can be seen as a measure of the similarity between the definitions of the words in the *cluster* and each definition of the word. Compared with the distance defined in (6), this distance is to measure the similarity between definitions, while the distance in (6) is to measure the similarity between contexts.

Thus it is reasonable to select the sense s^* among all as the correct one in the context, such that there exists $clu^* \in CLU_w$, and $dis_2(clu^*, s^*)$ gets the smallest value as (10), for $clu \in CLU_w$, and $s \in S_w$.

$$(10)\quad \underset{clu \in CLU_w, s \in S_w}{MIN} dis_2(clu, s)$$

5. Experiments and Results

In order to evaluate the application of the Chinese semantic space to WSD tasks, we make use of another Chinese lexical resource, i.e., Xiandai Hanyu Cihai (Zhang et al., 1994), a Chinese collocation dictionary. The sense distinctions in the dictionary are the same as those in the modern Chinese dictionary, and for each sense in the

[6] The words in the definitions are called definition words.

collocation dictionary, some words are listed as its collocations. We see these collocations as the contexts of the word senses, and evaluate our algorithm automatically. We randomly select 40 ambiguous words contained in the dictionary, and there are altogether 1240 words listed as their collocations. Table 2 lists the distribution of the number of the sense clusters activated by these collocation words.

Table 3 lists the distribution of the smallest distances between the word senses and the activated clusters, and the accuracy of the disambiguation. From Table 3, we can see that smaller distances between the senses and the activated clusters mean higher accuracy of disambiguation.

Number of activated clusters	Number of collocations
1	420
2	380
3	250
4	100
≥5	90
	All: 1240

Table 2. Collocation words and the number of activated clusters

Distance area	Percent(%)	Accuracy(%)
[0.0 0.2)	27.3	94.2
[0.2 0.4)	58.2	90.5
[0.4 0.6)	9.6	40.5
[0.6 1.0)	4.9	10.4

Table 3. Distribution of distances and disambiguation accuracy

In another experiment, we examine the ambiguous Chinese word 单薄 (/danbo/[7]), it has two senses, one is *less clothes taken by a man*, the other is *thin and weak*. We randomly select 100 occurrences of the word in the corpus, and implement our algorithm on them respectively. The result is 66 occurrences are tagged with the second sense (6 occurrences wrongly tagged), and the others tagged with the first sense (2 occurrences wrongly tagged). The overall accuracy is 92%. To examine the reasonableness of the result, we formalize four context vectors again based on semantic codes to represent the contexts of four groups of the occurrences:

cv_1: the context of the 60 occurrences correctly tagged with the second sense;

cv_2: the context of the 6 occurrences wrongly tagged with the second sense;

cv_3: the context of the 32 occurrences correctly tagged with the first sense;

cv_4: the context of the 2 occurrences wrongly tagged with the first sense;

The distances between these vectors are listed in Table 4:

	cv_1	cv_2	cv_3	cv_4
cv_1		0.364	0.914	0.825
cv_2	0.364		0.941	0.876
cv_3	0.914	0.941		0.320
cv_4	0.825	0.876	0.320	

Table 4. The distances between the contexts of the four groups

From Table 4, we find that both the distance between cv_1 and cv_4 and that between cv_2 and cv_3 are very high, which reflects the fact that they are not similar with each other. This demonstrates that one main reason for tagging errors is that the considered contexts of the words contain less meaningful information for determining the correct senses.

In third experiment, we implement our algorithm on 100 occurrences of the ambiguous word 编辑 (/bianji/), it also has two senses, one is *editor*, the other is *to edit*. We find the tagging accuracy is very low. To explore the reason for the errors, we

[7] The Piyin of the word.

compute the distances between its definitions and those of the words in the activated clusters, and find that the smallest distances fall in [0.34, 0.87]. This demonstrates that another main reason for the tagging errors is the *sparseness* of the clusters in the space.

6. Conclusions and Future work

In this paper, we propose a formal resource of language, *structured semantic space*, as a foundation for word sense disambiguation tasks. For a word in some context, the context can activate some sense clusters in the semantic space, due to its similarity with the contexts of the senses in the clusters, and the correct sense of the word can be determined by comparing its definitions and those of the words in the clusters.

Structured semantic space can be seen as a general model to deal with WSD problems, because it doesn't concern any language-specific knowledge at all. For a language, we can first make use of its mono-sense word to outline its semantic space, and produce a dendrogram according to their similarity, then word sense disambiguation can be carried out based on the dendrogram and the definitions of the words given in a dictionary.

As can be seen that ideal structured semantic space should be homogeneous, i.e., the clusters in it should be well-distributed, neither too dense nor too sparse. If it is too dense, there may be too many clusters activated by a context. On the contrary, if it is too sparse, there may be no clusters activated by a context, even if there is any, it may be the case that the senses in the clusters are not similar with the correct sense of the target word. So future work includes how to evaluate the homogeneity of the semantic space, how to locate the non-homogeneous areas in the space, and how to make them homogeneous.

Obviously, the disambiguation accuracy will be reduced if the cluster contains less words, because less words in the cluster will lead to invalidity of its definition vectors in revealing the similar words included in their definitions. But it seems to be impossible to ensure that every cluster contains enough words, with only mono-sense words taken into consideration when building the semantic space. In order to make the cluster contain more words, we must make use of ambiguous words. So future work includes how to add ambiguous words into clusters based on their contexts.

Another problem is about the length of the contexts to be considered. With longer contexts taken into consideration, there may be too many clusters activated. But if we consider shorter contexts, the meaningful information for word sense disambiguation may be lost. So future work also includes how to make an appropriate decision on the length of the contexts to be considered, meanwhile make out the meaningful information carried by the words outside the considered contexts.

References

J. Allen. 1995. *Natural Language Understanding*, The Benjamin/Cumming Publishing Company, INC.

Rebecca Bruce, Janyce Wiebe. 1994. Word sense disambiguation using decomposable models. In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics*, Las Cruces, New Mexico.

Y. Choueka and S. Lusignan. 1985. Disambiguation by short contexts. *Computers and the Humanities*, 19:147-157.

William Gale, Kenneth Ward Church, and David Yarowsky. 1992. Estimating upper and lower bounds on the performance of word-sense disambiguation programs. In *Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics*, Newark, Delaware.

Graeme Hirst. 1987. *Semantic Interpretation and the Resolution of Ambiguity.* Cambridge University Press, Cambridge.

Alpha K.Luk. 1995. Statistical sense disambiguation with relatively small corpora using dictionary definitions. In *Proceedings of the 33th Annual*

Meeting of the Association for Computational Linguistics, Cambridge, Massachusetts.

Susan W. McRoy. 1992. Using multiple knowledge sources for word sense disambiguation. *Computational Linguistics*, 18(1): 1-30.

J.J.Mei et al. 1983. *TongYiCi CiLin (A Chinese Thesaurus)*, Shanghai Cishu press, Shanghai.

Hwee Tou Ng and Hian Beng Lee. 1996. Integrating multiple knowledge sources to disambiguating word sense: an exemplar-based approach. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics.*

P. Resnik. 1995. Disambiguating noun groupings with respect to WordNet senses, In *Proceedings of 3rd Workshop on Very Large Corpus*, MIT, USA, 54-68.

H. Schutze. 1993. Part-of-speech induction from scratch. In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*, Columbus, OH.

Xiandai Hanyu Cidian. (a Modern Chinese Dictionary). 1978. Shnagwu Press, Beijing (in Chinese).

D. Yarowsky. 1992. Word sense disambiguation using statistical models of Roget's categories trained on large corpora, *Proceedings of COLING'92*, Nantas, France, 454-460.

David Yarowsky. 1994. Decision lists for lexical ambiguity resolution: Application to accent restoration in Spanish and French. In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics*, Las Cruces, New Mexico.

David Yarowsky. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33th Annual Meeting of the Association for Computational Linguistics*, Cambridge, Massachusetts.

Zhang et al. 1994. *Xiandai Hanyu Cihai. (a Chinese Collocation Dictionary)*, Renmin Zhongguo Press (in Chinese).