Vol. 37 No. 8 Transactions of Information Processing Society of Japan Aug. 1996

Regular Paper

Automatic Recognition of Verbal Polysemy

FUMIYO FUKUMOTO† and JUN'ICHI TSUJII††

Polysemy causes difficulties with the semantic clustering of words in corpus-based linguistics, and also leads to the problem of word-sense ambiguity in NLP systems. In this paper, we give a definition of polysemy from the viewpoint of clustering and propose an overlapping clustering algorithm which recognises verbal polysemies in a textual corpus. The main characteristic of our algorithm is that it explicitly introduces new entities called ***hypothetical verbs*** when an entity is judged polysemous, and associates them with the context of the original entity. We have conducted experiments in order to examine the effects of our algorithm, and the results demonstrate the effectiveness of the method.

1. Introduction

Much research has been done on automatic clustering of semantically similar words and on automatic recognition of collocations among them in corpora[5),10),15),17),18)]. Most of this work is based on similarity measures derived from the distribution of words in corpora. However, the facts that a single word may have more than one meaning and that the distribution of a word in a corpus reflects its usage in a variety of senses often hamper such attempts and also cause the problem of word-sense ambiguity in NLP systems.

The *meaning* of a word depends on the context in which it is used; the same word can be used differently in different contexts. Furthermore, a word that is polysemous in general is often not polysemous in a restricted subject domain. In fact, restriction of the subject domain usually makes polysemy less problematic. However, unlike nouns, verbs are often polysemous even in a restricted subject domain such as financial texts.

Because polysemous verbs are usually also high-frequency verbs, their treatment is crucial in actual applications. Polysemous verbs tend to have a harmful influence on the semantic clustering of nouns, because this is usually performed on the basis of the noun's collocation with verbs.

Although polysemy is said to be widespread in languages, the definition of polysemy is highly subjective. Polysemy can only be recognised by human intuition, and different linguists often identify different numbers of senses for the same word. In this paper, we give a definition of polysemy from the viewpoint of clustering, and propose a clustering method which *automatically* recognises polysemous words. The results of experiments are given to demonstrate the effectiveness of our method.

2. Related Work

There have been several attempts to extract semantically similar words from a given corpus by using statistical methods. However, few studies seriously deal with the problem of polysemy; of these, even fewer are based on real texts.

The techniques developed by Zernik[19)] and Brown[2)] seem to cope with the discrimination of polysemy, and are also based on real texts. Zernik used monolingual part-of-speech tagged texts consisting of about 1 million words. His method associates each sense of a polysemous word with a set of its co-occurring words. If a word has several senses, then the word is associated with several different sets of co-occurring words, each of which corresponds to one of the senses of the word. The limitation of Zernik's method, however, is that it relies solely on human intuition for identifying different senses of a word; that is to say, the human editor has to determine, by her/his intuition, how many senses a word has, and then identify the sets of co-occurring words that correspond to the different senses.

† Department of Electrical Engineering and Computer Science, Faculty of Engineering, Yamanashi University
†† Department of Information Science, Faculty of Science, University of Tokyo

Brown used bilingual texts. The results of Brown's technique, when applied to a French-English machine translation system, seem to show its effectiveness and validity. As he admits, however, the approach is limited because it can assign at most two senses to a word. More seriously, polysemy is defined in terms of translation; that is to say, a word is regarded as polysemous only when it can be translated by two different words in the target language. Moreover, the approach can be used only when a large parallel corpus is available, and the individual senses thus identified do not necessarily constitute single semantic units in the monolingual domain with which plausible semantic properties (i.e., semantic restrictions, collocations, etc.) can be associated.

The defects of these two methods show that it is crucial to have an appropriate definition of polysemy in terms of the distributional behaviours of words in monolingual texts. The approach proposed in this paper especially focuses on this problem. Like Brown's approach, our approach adopts a relativistic view of polysemy. That is, a word is regarded as polysemous in relation to other related words. However, while Brown's approach identifies polysemous words in relation to similar words in another language, we use semantically similar words in the same language to identify polysemous words. Whether a word is polysemous depends on whether there exists a set of semantically similar words whose distributions are among the *subsets* of the distribution of that word. Because the distributional behaviour of a word is characterised by its co-occurring words, the process of identifying such *subsets* essentially corresponds to the process performed manually by the human editor in Zernik's approach.

The experiments in this paper use a corpus annotated with parts-of-speech but not with syntactic structures. The clustering algorithm assumes only that words are semantically characterised by a vector in an n-dimensional Euclidean space so that the algorithm can be applied to any data satisfying this condition.

3. Framework

3.1 Polysemy in Context

The basic assumption of this work is the same as that of previous corpus-based approaches, namely, that semantically similar words appear in similar contexts[3),5),10),11)]. Semantically similar verbs, for example, co-occur with the same set of nouns. The following sentences from the Wall Street Journal corpus☆ show the point:

(s1) The thrift holding company said it expects to obtain regulatory approval and complete the transaction by year-end.

(s2) Simmons finally managed to get shareholder approval in December.

(s3) Since 1967, Italy has required Italian companies to gain governmental approval before entering negotiations for arms-export sales.

It is intuitively obvious that obtain, get, and gain are semantically related and that the semantic similarity of these three verbs is manifested by the fact that they co-occur with the same noun approval.

We can imagine that a verb is located in an n-dimensional Euclidean space, each dimension of which is associated with a specific noun. Following Church[5)], we assume that the i-th coordinate of the verb is the *mutual information* (which we call Mu) between the verb and the i-th noun, and not the co-occurrence frequency. Mu is based on mutual information in information theory[7),16)], and provides a better measure for characterising the semantic relationships between two words than mere co-occurrence frequencies[6)]☆☆.

If the basic assumption is correct, then semantically similar verbs should be located closer together than other verbs, and it should therefore possible to use a clustering algorithm in order to discover semantic classes of verbs.

However, this straightforward method is often hampered by the existence of polysemous words. Consider the following sentence pairs, which show polysemous usages of take:

(s4) In the past, however, coke has typically taken a minority stake in such ventures.

(s4') Guber and Peters tried to buy a stake in MGM in 1988.

(s5) That process of sorting out specifies is likely to take time.

☆ The Wall Street Journal corpus was prepared by the ACL (Association for Computational Linguistics' Data Collection Initiative) in 1991.

☆☆ Comparative experiments based on the co-occurrence frequency and on Mu are reported in Fukumoto[8)].

(s5') We spent a lot of time and money in building our group of stations.

(s6) People are queuing at the door to take his product but he doesn't have the working capital to make the thing.

(s6') Goodyear used Atwood trade credits to obtain chemicals and other products and services in the U.S.

We can make the following observations:

(1) Take and buy in (s4) and (s4'), take and spend in (s5) and (s5'), take and obtain in (s6) and (s6') co-occur with the noun stake, time and product, respectively, and the verbs of each of these pairs have almost the same sense.

(2) While certain usages of take have senses similar to buy, spend, and obtain, these three specific verbs have distinct senses and we hardly see synonymy among these verbs.

Let us now consider a three-dimensional Euclidean space spanned by three axes associated with stake, time, and product. Take co-occurs with the three nouns and has high *Mu* values with them, while buy, spend and obtain have high *Mu* values only with one of the three nouns. Therefore, the distances between take and these three verbs are large, and the synonymy of take with them does not hold.

To capture the synonymy of take with the three verbs correctly, one has to decompose the vector assigned to take into three component vectors, each of which corresponds to one of the three distinct usages of take. The decomposition of a vector into a set of its component vectors requires a proper decomposition of the context in which the word occurs. **Figure 1** shows the decomposition of take in the three-dimensional spaces formed with stake, time, and product. **Take1**, **take2**, and **take3** are the respective component vectors and they collectively constitute the vector assigned to take.

For the sake of simplicity, we assumed in Fig. 1 that the three nouns characterise the contexts where the verb take occurs and, at the same time, each of them characterises a distinct usage of take. However, in a general situation, a polysemous verb co-occurs with a large set of nouns and one has to divide the set of nouns into subsets, each of which correctly characterises the context for a specific sense of the polysemous verb; that is to say, one of the meanings of a polysemous verb can be characterised by a subset of nouns that co-occur with another semantically similar verb. Therefore, the algorithm has to be able to determine when and how the context of a word should be divided.

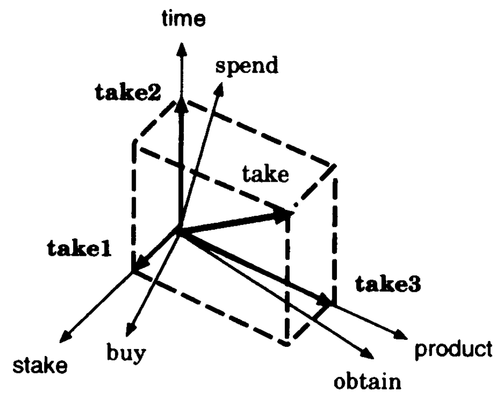

Fig. 1 Decomposition of the verb take.

3.2 Overlapping Clustering

There are clustering algorithms, called *overlapping clustering* algorithms, which allow an entity to belong to more than one cluster. The B_k ($k = 1,2,\cdots$) method, proposed by Jardine and Sibson[13], is a typical overlapping clustering algorithm, in which a maximum of k-1 entities can belong to several different clusters. This algorithm, however, makes an entity belong to several clusters if there is no cluster set in which the clusters have more than k-1 entities in common. Whether an entity belongs to several clusters or not depends on the value of k. As a result, a verb belongs to several clusters even if it is not polysemous. Let {obtain,get} be a cluster which has already been created. Let {get,gain} and {obtain,get,gain} be candidates for a new cluster. In B_2, a maximum 2-1 = 1 entity can belong to several clusters if there is no set of clusters where the clusters have more than a 2-1 = 1 entity in common. In the example, there are no clusters which have a 2-1 = 1 entity in common, since only one cluster ({obtain,get}) has already been created. Therefore, the algorithm makes one entity ('get') belong to another cluster ({get,gain}). As a result, 'get' is judged to be polysemous. Furthermore, these overlapping clustering algorithms assume that an entity belonging to more than one cluster is still a single entity; that is, they do not include explicit splitting of the contexts which characterise the entities. In (s4)−(s6'), for example, even if 'take' is judged to have three meanings, 'buy', 'spend', and 'obtain', the algorithms do not split 'take' into three different classes, as shown in Fig. 1. An entity belongs to several clusters, because it can be seen from several different viewpoints. In different contexts, a single entity, such as 'take', can be used to mean 'buy something', as in (s4), and 'get hold of something', as in (s6).

As we have seen, polysemous verbs can be captured more naturally by seeing them as *multiple entities* which happen to take the same surface form. **Take1**, **take2**, and **take3** are distinct entities (we call them *hypothetical verbs* in the following) with which different sets of nouns co-occur, and with which, therefore, different contexts are associated.

Our algorithm hence explicitly introduces new entities (i.e., hypothetical verbs), when an entity is judged polysemous, and associates them with contexts which are subcontexts of the context of the original entity. The algorithm has two basic operations, *splitting* and *lumping*. Splitting means dividing a polysemous verb into two hypothetical verbs, and lumping means combining two hypothetical verbs into a single verb.

3.3 Measuring the Deviation of a Set of Verbs

The algorithm should decide when a verb has to be split into two hypothetical verbs. The decision is based on the semantic deviation of a set of verbs. Semantic deviation is a measure of the deviation of the set in an n-dimensional Euclidean space, where n is the number of nouns which co-occur with the verbs. Given a set of verbs, VG $= \{v_1, \cdots, v_m\}$, where m is the number of verbs in VG, the deviation of VG is defined as follows:

(1) Let $v_i \in$ VG, and let the vector assigned to v_i be $(v_{i1}, \cdots, v_{in})$. Each v_{ij} $(1 \leq j \leq n)$ is computed by the following formula:

$$v_{ij} = \begin{cases} Mu(v_i, n_j) & \text{if } Mu(v_i,n_j) \geq \alpha \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

Here, $Mu(v_i, n_j) = \log \frac{P(v_i,n_j)}{P(v_i)P(n_j)}$, where $P(v_i)$ and $P(n_j)$ are the probabilities of the events v_i and n_j, $P(v_i, n_j)$ is the probability that the two events are observed at the same time, and α is a threshold value given in advance.

(2) The deviation dev_{temp}(VG) is defined as:

$$dev_{temp}(\text{VG}) = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}(v_{ij} - \bar{g}_j)^2} \tag{2}$$

Here, $\bar{g}_j = \frac{1}{m}\sum_{i=1}^{m} v_{ij}$ is the j-th value of the centre of gravity.

(3) Let us consider two sets, A and B, which have the same degree of deviation. Let $|\ \bar{g}\ | = \frac{1}{m}\sqrt{\sum_{j=1}^{n}(\sum_{i}^{m} v_{ij})^2}$ be the length of the centre of gravity. If $|\ \bar{g}\ |$ of A is larger than that of B, the absolute value of Mu calculated for A is larger than that of B. This means that the absolute probabilities of the co-occurrences of each noun and the verbs in A are larger than those of B. As a result, A should be judged to be semantically less deviant than B. Then, dev_{temp}(VG) is normalised as:

$$dev_{norm}(\text{VG}) = \frac{1}{|\ \bar{g}\ |}\sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}(v_{ij} - \bar{g}_j)^2} \tag{3}$$

(4) dev_{norm} shows that a set of a greater number of verbs tends to be semantically more deviant than a set of a smaller number of verbs. We further normalise the deviation (3) to compensate for this effect, as follows:

$$Dev(\text{VG}) = \frac{1}{|\ \bar{g}\ |(\beta * m + \gamma)}\sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}(v_{ij} - \bar{g}_j)^2} \tag{4}$$

β and γ are obtained by *least square estimation* as regards $dev_{norm}(\text{VG}) = \beta m + \gamma$. For instance, using the Wall Street Journal corpus, and setting α in (1) equal to 3.0, we obtain $\beta = 0.964$ and $\gamma = -0.495$.

In the following, a set with a smaller value of (4) is considered *semantically less deviant.*

3.4 Clustering Method

In this section, we present our clustering algorithm. We first explain the operations *splitting* and *lumping*. Then, we show the flow of the algorithm and explain how the whole algorithm works.

3.4.1 The Basic Idea

As stated in Section 3.1, a polysemous word behaves like a set of different words; that is to say, different word senses behave as if they were distinct words, and co-occur with different sets of words. Therefore, clusters of verbs will be less deviant when polysemous verbs are treated as multiple hypothetical verbs belonging to different clusters. Our method is thus an overlapping clustering method which explicitly exploits this feature of polysemy. The essential difference between our algorithm and the B_k method is that our algorithm explicitly introduces a condition on which a verb is judged polysemous. Furthermore, it introduces hypothetical verbs to determine whether a verb should be split and associated with several clusters. For example, to determine whether 'take' has two senses 'buy' and 'spend', our algorithm splits 'take' into two hypothetical verbs, **take1** and **take2**, which are distinct entities associated with different contexts.

3.4.2 Splitting and Lumping

Let v and w_p be verbs and $w_1, \cdots, w_n$ be verbs or hypothetical verbs. Suppose $Dev(v, w_i) \leq Dev(v, w_j)$ $(1 \leq i \leq j \leq n)$ and $Dev(v, w_1) \leq Dev(v, w_p)$. In our method, in order to determine whether v has two senses clustered with w_1 and w_p, we tentatively make two clusters by splitting, as shown in (5), and a merged cluster by lumping, as shown in (6), and compare the semantic deviation among the three resulting clusters.

$$\{v_1, w_p\}, \{v_2, w_1, \cdots, w_n\} \quad (5)$$

$$\{v, w_1, \cdots, w_p, \cdots, w_n\} \quad (6)$$

$w_1, \cdots, w_p, \cdots, w_n$ in (6) satisfy $Dev(v, w_i) \leq Dev(v, w_j)$ $(1 \leq i \leq j \leq n)$. v_1 and v_2 in (5) are new hypothetical verbs which correspond to two distinct senses of the same verb v. For example, in order to determine whether 'take' has two senses 'buy' and 'spend', we make two clusters $\{$**take1**, buy$\}$, $\{$**take2**, spend$\}$ by splitting, and a merged cluster $\{$take, buy, spend$\}$ by lumping, and compare the semantic deviation among the three resulting clusters.

In our current implementation, if the semantic deviation of each cluster in (5) is smaller than that of the cluster in (6), v is regarded as polysemous. Splitting and lumping operations are defined as follows:

(1) The function *split* applies to v, w_1, and w_p, and returns the hypothetical verbs v_1 and v_2 of the verb v☆.

$$split(v, w_p, w_1) = (v_1, v_2) \quad (7)$$

where

$$Dev(v, w_1) \leq Dev(v, w_p) \quad (8)$$

$$v_1 = \begin{bmatrix} v_{11} \\ v_{12} \\ \vdots \\ v_{1n} \end{bmatrix} \text{ s.t.}$$

$$v_{1j} = \begin{cases} v_j & \text{if } w_{pj} \neq 0 \quad (9) \\ 0 & \text{otherwise} \quad (9') \end{cases}$$

☆ The vector assigned to v is $(v_1, \cdots, v_n)$.

$$v_2 = \begin{bmatrix} v_{21} \\ v_{22} \\ \vdots \\ v_{2n} \end{bmatrix} \text{ s.t.}$$

$$v_{2j} = \begin{cases} v_j & \text{if } (w_{1j} \neq 0 \text{ or } w_{pj} = w_{1j} = 0) \quad (10) \\ 0 & \text{otherwise} \quad (10') \end{cases}$$

Note that if n_j co-occurs with v and also with both w_p and w_1, v_{1j} and v_{2j} are both $v_j = Mu(v, n_j)$ as shown in (9) and (10). Furthermore, if n_j co-occurs with v, but with neither w_1 nor w_p, v_j is assigned to v_{2j} as shown in (10). If v_j is included in neither v_1 nor v_2, $\{v_2, w_1\}$ will be less deviant than $\{v_1, w_p\}$. Then, in order to make v_1 and v_2 as equally deviant as possible, v_j is set to v_{2j}.

(2) The function *lump* has the opposite effect of *split*. That is, it merges two hypothetical verbs w_1 and w_2 into one verb w_p, as follows:

$$lump(w_1, w_2) = w_p \quad (11)$$

$$w_p = \begin{bmatrix} w_{p1} \\ w_{p2} \\ \vdots \\ w_{pn} \end{bmatrix} \text{ s.t.}$$

$$w_{pj} = \begin{cases} w_{1j} + w_{2j} & \text{if } w_{1j} \neq w_{2j} \\ w_{1j} & \text{if } w_{1j} = w_{2j} \end{cases}$$

3.4.3 The Flow of the Algorithm

Given a set of verbs, VG $= \{v_1, \cdots, v_m\}$, the algorithm produces a set of semantic clusters, which are sorted in ascending order of their semantic deviation. If v_i is non-polysemous, it belongs to exactly one of the resultant semantic clusters. If it is polysemous, the algorithm splits it into several hypothetical verbs, each of which belongs to exactly one semantic cluster. **Figure 2** shows the flow of the algorithm. As shown in '(' in Fig. 2, the function Make-Initial-Cluster-Set applies to VG and produces all possible pairs of verbs with their semantic deviation values. The result is a list of pairs called the ICS (Initial Cluster Set). The CCS (Created Cluster Set) shows the clusters which have been created so far. The function Make-Temporary-Cluster-Set retrieves the clusters from the CCS which contain one of the verbs of Set_i. The results (Set_β) are passed to the function Recognition-of-Polysemy, which determines whether or not a verb is polysemous. Let v be an element included in both Set_i and Set_β. To determine whether v has two senses w_p, where w_p is an element of Set_i, and w_1,

```
begin
  ICS := Make-Initial-Cluster-Set(VG)
   ( VG = {v_i | i = 1, ..., m}
   ( ICS = {Set_1, ..., Set_{m(m-1)/2}}
   ( where Set_p = {v_i, v_j} and Set_q = {v_k, v_l} ∈ ICS (1 ≤ p < q ≤ m)
   ( satisfy Dev(v_i, v_j) ≤ Dev(v_k, v_l)
  for i := 1 to m(m-1)/2 do
      if CCS = φ
           then Set_γ := Set_i
                 i.e. Set_i is stored in CCS as a newly obtained cluster
      else if Set_α ∈ CCS exists such that Set_i ⊂ Set_α
           then Set_i is removed from ICS and Set_γ := φ
      else if
           for all Set_α ∈ CCS do
               if Set_i ∩ Set_α = φ
                   then Set_γ := Set_i
                         i.e. Set_i is stored in CCS as a newly obtained cluster
               end_if
      else Set_β := Make-Temporary-Cluster-Set(Set_i,CCS)
           ( Set_β := Set_α ∈ CCS such that Set_i ∩ Set_α ≠ φ
             Set_γ := Recognition-of-Polysemy(Set_i,Set_β)
      end_if
      end_if
      end_if
      if Set_γ = VG
         then exit from the for_loop ;
      end_if
  end_for
end
```

Fig. 2 Flow of the algorithm.

where w_1 is an element of Set_β, we make two clusters, as shown in (5), and their merged cluster, as shown in (6). The splitting function (7) is applied to v, w_1, and w_p to produce newly hypothetical verbs v_1 and v_2. If hypothetical verbs w_1 and w_2 exist in the process of making the cluster shown in (6), the lumping function (11) is applied to w_1 and w_2, and makes one verb w_p out of them.

The whole process is repeated until the newly obtained cluster, Set_γ, contains all the verbs in the input or the ICS is exhausted.

4. Experiment

In this section, we report the results of two experiments: Hypothetical-Verb-Exp, where the clustering algorithm introduces a hypothetical verb, and Not-Hypothetical-Verb-Exp, where this is not the case.

Let us recall that in order to determine whether v has two senses clustered with w_1 and w_p, we tentatively make two clusters by splitting, as shown in (5), and a merged cluster by lumping, as shown in (6). In Not-Hypothetical-Verb-Exp, we merely replace the hypothetical verbs v_1 and v_2 in (5) with v.

4.1 Data

The corpus we have used is the Wall Street Journal which consists of 182,992 sentences, 2,878,688 occurrences of part-of-speech tagged words and 73,225 different words[4]. From this corpus, we obtained 5,940,193 word pairs in a window size of 5 words, 2,743,974 different word pairs. A pair of x and y in a window size of 5 words means that x is followed by y within a five-word distance.

We used verb-noun pairs; that is to say, we assumed an n-dimensional Euclidean space, in which a verb is assigned a vector whose value of the i-th dimension is the value of Mu between the verb and the noun assigned to the i-th axis. This is because in small window sizes, the semantic relationship between two words might be quite strong, especially between a verb and its object, permitting the effective recognition of verbal polysemy. Inflected forms of the same word, such as 'time' (noun, singular) and 'times' (noun, plural), were treated as a single unit. From 2,743,974 superficially different pairs, we thus obtained 228,665 different verb-noun pairs, and from these, we selected 6,768 different verb-noun pairs, 701 different verbs, and 1,796 nouns. To obtain a reliable statisti-

Table 1 'Typel' verb sets and the results of experiments.

[1-1]	<u>close</u> open end
[1-2]	<u>take</u> obtain spend buy
[1-3]	<u>lose</u> win miss
[1-4]	<u>get</u> gain receive
[1-5]	open <u>close</u> end buy sell
[1-6]	open <u>close</u> end buy solve
[1-7]	<u>close</u> open end <u>take</u> spend buy obtain cancel solve ride
[1-8]	see realize open <u>close</u> end
[1-9]	<u>see</u> look know
[1-10]	<u>come</u> go become
[1-11]	<u>find</u> receive see
[1-12]	<u>get</u> gain become
[1-13]	<u>leave</u> retire remain borrow lend
[1-14]	<u>grow</u> happen increase

Typel	Hypothetical-Verb-Exp			Not-Hypothetical-Verb-Exp		
	Results	Correct (%)	Incorrect (%)	Results	Correct (%)	Incorrect (%)
[1-1]	o			a		
[1-2]	o			a		
[1-3]	o			a		
[1-4]	o			a		
[1-5]	o			a		
[1-6]	o			a		
[1-7]	o	9(64.2)	5(35.8)	a	0(0)	14(100)
[1-8]	b			a		
[1-9]	a&b			a		
[1-10]	a&b			a&b		
[1-11]	a&b			a		
[1-12]	o			a		
[1-13]	a			a		
[1-14]	o			a		

'______': Polysemous verb.
'typel': Set of verbs that containing one or more polysemous verbs.
'o': Set of verbs that could be clustered correctly.
'a': Recognition of polysemous verbs as non-polysemous ones.
'b': Recognition of non-polysemous verbs as polysemous ones.

cal analysis,☆ we selected only those words and word pairs whose frequencies and Mu were not low ($f(x,y) \geq 5$ and $Mu(x,y) \geq 3$).

The verbs used in the experiments consisted of 26 sets. We classified these sets into two different types, 'typel' and 'type2'. Verb sets of the first type contained one or more polysemous verbs, and were used to determine whether our algorithm could recognise polysemous verb correctly. Verb sets of the second type did not contain any polysemous verbs, and were used to determine whether our algorithm was also effective for such sets. 'Typel' and 'type2' verb sets are shown in **Tables 1** and **2**, respectively. In Table 1, polysemous verbs in the sets are underlined.

We selected polysemous verbs whose frequencies in the corpus are high, and then selected subcategorisation patterns[9] of the chosen polysemous verbs according to the Collins dictionary and thesaurus[14].

4.2 Results

The results are shown in Tables 1 and 2. In Tables 1 and 2, the symbol 'o' in the 'results' column means that the set of verbs was clustered correctly, while 'a' and 'b' denote two error types: 'a' refers to the recognition of polysemous verbs as non-polysemous ones and 'b' to the recognition of non-polysemous verbs as polysemous ones☆☆. The column 'correct' shows the number of sets of verbs which are clustered correctly, while 'incorrect' shows the number of

☆ The values of $f(x,y) \geq 5$ and $Mu(x,y) \geq 3$ are empirically determined.

☆☆ 'a & b' denotes the co-occurrence of 'a' and 'b' error types.

Table 2 'Type2' verb sets and the results of experiments.

[2-1]	increase reduce buy come begin
[2-2]	acquire own finance fund
[2-3]	gather gain offer accept
[2-4]	borrow lend limit reduce
[2-5]	oppose reject offer provide increase reduce
[2-6]	close end finish
[2-7]	reject accept receive
[2-8]	act behave find get
[2-9]	feel look cut keep arrive leave
[2-10]	sound feel seem come go
[2-11]	open close offer provide
[2-12]	look see offer provide

Type2	Hypothetical-Verb-Exp			Not-Hypothetical-Verb-Exp		
	Results	Correct (%)	Incorrect (%)	Results	Correct (%)	incorrect(%)
[2-1]	o			o		
[2-2]	o			o		
[2-3]	o			o		
[2-4]	o			o		
[2-5]	o			o		
[2-6]	b	9(75.0)	3(25.0)	b	11(91.7)	1(8.3)
[2-7]	o			o		
[2-8]	o			o		
[2-9]	b			o		
[2-10]	b			o		
[2-11]	o			o		
[2-12]	o			o		

'type2': Set of verbs that do not containing any polysemous verbs.
'o': Set of verbs that could be clustered correctly.
'a': Recognition of polysemous verbs as non-polysemous ones.
'b': Recognition of non-polysemous verbs as polysemous ones.

those that are not. The correctness of the results was judged by a native speaker.

4.3 Discussion

4.3.1 Effectiveness of the Method

In Tables 1 and 2, the total results of Hypothetical-Verb-Exp show that 18 (9 for each table) out of 26 clusters of verbs were obtained correctly, and the percentage attained was 69.2%. On the other hand, the results of Not-Hypothetical-Verb-Exp show that only 11 (0 for Table 1 and 11 for Table 2) out of 26 clusters were obtained. Comparison of the total ratios of correct judgements made in these two experiments shows that the ratio of the correct results in Hypothetical-Verb-Exp (1,800/26 = 69.2%) was significantly higher than that in Not-Hypothetical-Verb-Exp (1,100/26 = 42.3%). Furthermore, in Table 1, the results of Not-Hypothetical-Verb-Exp show that no verbs were recognised correctly as polysemous. This demonstrates the effectiveness of our clustering method, which explicitly introduces a hypothetical verb. Let v be a polysemous verb, and let x and y be subcategorisation patterns of v. Further, let $Dev(v,x) \leq Dev(v,y)$. Examining the fact that no verbs were recognised correctly as polysemous in Not-Hypothetical-Verb-Exp, we found that all of the polysemous verbs in 14 sets satisfied $Dev(v,x) \leq Dev(v,x,y) \leq Dev(v,y)$, while polysemous verbs in 9 out of 14 sets which were recognised correctly in Hypothetical-Verb-Exp satisfied $Dev(v_1,x) \leq Dev(v_2,y) \leq Dev(v,x,y)$. Comparison of the value of the difference between $Dev(v,x,y)$ and $Dev(v,y)$ with that of the difference between $Dev(v,x,y)$ and $Dev(v_2,y)$ for the polysemous verbs in these 9 sets show that the former was quite small for 3 of the 9 sets. For example, in [1-1], the former was 0.010, while the latter was 0.107. In the experiments, we selected polysemous verbs which had high frequencies and whose subcategorisation patterns were clearly distinct from each other. This seems to result in a high level of correctness. To guarantee the reliability of our algorithm, we will conduct further experiments using other polysemous verbs.

Figure 3 shows samples of the results of Hypothetical-Verb-Exp. Each cluster formation is annotated by the semantic deviation. The X-axis represents the semantic deviation.

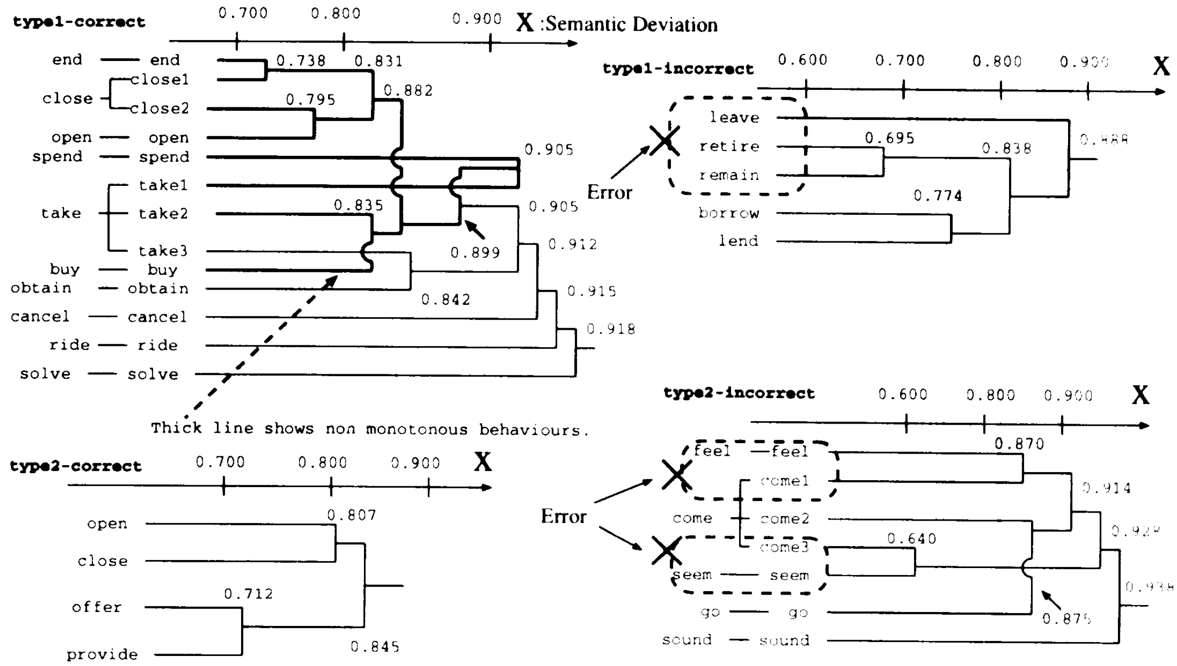

Fig. 3 Results of the clustering analysis.

In **type1-correct**, we can see that 'take' is recognised as polysemous and has three different subcategorisation patterns, which are the same as those of 'spend', 'buy', and 'obtain'. In a similar way, 'close' has two different subcategorisation patterns, 'end' and 'open'☆. In both **type1-correct** and **type2-correct**, the same subcategorisation patterns are grouped together.

We note that **type1-incorrect**, **type2-correct**, and **type2-incorrect** are monotonous clusters; that is to say, the degree of semantic deviation of larger clusters is larger than those of smaller clusters. On the other hand, **type1-correct** shows non-monotonous behaviour. In **type1-correct**, the larger cluster, which consists of eight verbs ('end', 'close1', 'close2', 'open', 'spend', 'take1', 'take2', and 'buy') has a smaller value (0.899) than the smaller cluster of 'spend' and 'take1' (0.905). The non-monotonous phenomenon is one of the characteristics of the *centroid method* to which our proposed method belongs; that is, the semantic deviation in a cluster is basically obtained by adding each verb's deviation from the centre of gravity[12].

☆ 'Close' has at least two different subcategorisation patterns, which are the same as those of 'end' and 'shut'. However, as 'shut' has a low frequency in the corpus, we selected the opposite sense of 'shut', 'open', and applied this to our clustering algorithm.

4.3.2 Problem of the Method

In Fig. 3, 'leave' and 'come' are incorrectly recognised as non-polysemous in **type1-incorrect** and as polysemous in **type2-incorrect**, respectively.

For example, in the corpus, 'leave' has at least two different subcategorisation patterns, 'retire' and 'remain':

(s6) Kaplan left his job at Warner-Lambert.

(s6') About 12 % have retired from a full-time job.

(s7) They can even leave a sticky problem, in the form of higher brokerage commissions.

(s7') ... but remain a serious problem.

However, **type1-incorrect** in Figure 3 shows that 'leave' is incorrectly recognised as a non-polysemous verb.

Table 3 shows the ICS of {leave, retire, remain, borrow, lend}.

The value in Table 3 shows the semantic deviation of the cluster. The semantic deviation of {retire, remain} is smaller than that of any other word pair, and the number of nouns which co-occur with 'retire' and 'remain' is larger than that for any other pairs of words. As a result, 'retire' and 'remain' are grouped together.

One possible cause of the above result is that the threshold values for frequencies and Mu are

Table 3 ICS of {leave, retire, remain, borrow, lend}.

Set_1	=	{retire, remain}	0.695	Set_2	=	{borrow, lend}	0.774
Set_3	=	{remain, borrow}	0.911	Set_4	=	{leave, remain}	0.966
Set_5	=	{leave, borrow}	0.987	Set_6	=	{retire, borrow}	0.987
Set_7	=	{retire, lend}	0.987	Set_8	=	{leave, retire}	0.987
Set_9	=	{leave, lend}	0.987	Set_{10}	=	{remain, lend}	0.987

not appropriate. We required that the frequencies and Mu should not be low ($f(x,y) \geq 5$ and $Mu(x,y) \geq 3$) to guarantee the reliability of statistical analysis, but it is not clear whether these thresholds are appropriate for yielding the best performance in an n-dimensional space.

Another possible cause is that the use of verb-object pairs alone is not appropriate. For example, in Table 2, [2-9] and [2-10] could not be clustered correctly in Hypothetical-Verb-Exp, and the results for Hypothetical-Verb-Exp were worse than those for Not-Hypothetical-Verb-Exp. Examination of the verbs belonging to [2-9] and [2-10] shows that *stative verbs* such as 'feel', 'sound', and 'seem' and *action verbs* such as 'arrive' and 'go' are included in these sets. Most adverbs of manner naturally refer to action verbs and not to stative verbs[1]. This shows that action verbs are modified by manner adverbs. We can therefore expect that adverbial adjuncts will be relevant to the correct distinction between these stative verbs and action verbs. We will conduct more experiments to confirm this.

5. Conclusion

We have defined polysemy from the viewpoint of clustering, and proposed an overlapping clustering method which *automatically* recognises verbal polysemies from a textual corpus. The results of the experiments demonstrate the effectiveness of our method for automatic recognition of polysemy. We will conduct further experiments to investigate issues related to accuracy and to apply this work to practical tasks such as word-sense disambiguation.

Acknowledgments

We would like to thank Professor Juan Carlos Sager and Dr. Kristiina Jokinen for their meticulous revision of our text. We also wish to express our gratitude to the reviewers for their valuable comments. Thanks are also due to the members of the Centre for Computational Linguistics at UMIST, especially to the following: Masaki Kiyono, Nancy Underwood and Jun'ichi Fukumoto for their technical comments and suggestions.

References

1) Alexander, L: *Longman English Grammar*, London, Longman (1992).
2) Brown, P.F., et al.: Word-Sense Disambiguation Using Statistical Methods, *Proc. the 29th Annual Meeting of the ACL*, pp.264–270 (1991).
3) Calzolari, N. and Remo, B.: Acquisition of Lexical Information from a Large Textual Italian Corpus, *Proc. the 13th International Conference on Computational Linguistics*, Vol.3, pp.54–59 (1990).
4) Church, K.W.: A Stochastic Parts Programme and Noun Phraser for Unrestricted Text, *Proc. the 2nd Conference on Applied Natural Language Processing*, pp.136–143 (1988).
5) Church, K.W. and Hanks, P.: Word Association Norms, Mutual Information, and Lexicography, *Computational Linguistics*, Vol.16, No.1, pp.22–29 (1990).
6) Church, K.W., et al.: Using Statistics in Lexical Analysis, *Lexical Acquisition: Exploiting On-line Resources to Build a Lexicon*, Zernik, U. (Ed.), pp.115–164, Lawrence Erlbaum Associates, London (1991).
7) Fano, R.: *Transmission of Information: A Statistical Theory of Communication*, MIT Press, Cambridge, MA (1961).
8) Fukumoto, F.: Automatic Recognition of Verbal Polysemy, MSc Thesis, CCL, UMIST (1993).
9) Grishman, R.: *Computational Linguistics: An Introduction*, Cambridge University Press (1986).
10) Hindle, D.: Noun Classification from Predicate-Argument Structures, *Proc. the 28th Annual Meeting of the ACL*, pp.268–275 (1990).
11) Inoue, N.: Automatic Noun Classification by Using Japanese-English Word Pairs, *Proc. the 29th Annual Meeting of the ACL*, pp.201–208 (1991).
12) Jain, A.K. and Dubes, R.C.: *Algorithms for Clustering Data*, Prentice-Hall, New Jersey (1988).
13) Jardine, N. and Sibson, R.: The Construction of Hierarchic and Non-hierarchic Classifications, *Computer Journal*, pp.177–184 (1968).
14) McLeod, W.T.: *The New Collins Dictionary and Thesaurus in One Volume*, HarperCollins

Publishers, London (1987).
15) Sekine, S., et al.: Linguistic Knowledge Generator, *Proc. the 14th COLING-92*, pp.560–566 (1992).
16) Shannon, C.E.: *A Mathematical Theory of Communication*, Univ. Ill. Press (1949).
17) Smadja, F.A.: Retrieving Collocations from Text: Xtract, *Computational Linguistics*, Vol. 19, No.1, pp.143–177 (1993).
18) Tsujii, J., et al.: Linguistic Knowledge Acquisition from Corpora, *Proc. the International Workshop on Fundamental Research for the Future Generation of Natural Language Processing*, pp.61–81 (1992).
19) Zernik, U.: Train1 vs. Train2: Tagging Word Senses in Corpus, *Lexical Acquisition: Exploiting On-line Resources to Build a Lexicon*, Zernik, U. (Ed.), Lawrence Erlbaum Associates, London, pp.91–112 (1991).

(Received March 14, 1994)
(Accepted April 12, 1996)

Fumiyo Fukumoto Graduated from Department of Mathematics in the faculty of Sciences, Gakushuin University, 1986. From 1986 to 1988, she joined R & D Department of Oki Electric Industry Co., Ltd. From 1988 to 1992, she joined Institute for New Generation Computer Technology (ICOT). She was at Centre for Computational Linguistics of UMIST (University of Manchester Institute of Science and Technology), England as a student and a visiting researcher, from 1992 to 1994, and awarded MSc. Since 1994, she has been a research assistant at Yamanashi University. Her current research interest is natural language processing, in particular corpus linguistics. She is a member of the Association for Natural Language Processing and the Association for Computational Linguistics.

Jun'ichi Tsujii Graduated from Department of Electrical Engineering, Kyoto University, 1971 and awarded MSc and Ph.D. from Kyoto University 1973 and 1979, respectively. He was Associate Professor of Department Electrical Engineering, Kyoto University until 1988 and then appointed as Professor of Computational Linguistics at Centre for Computational Linguistics of UMIST (University of Manchester Institute Science and Technology), England. He was appointed as Professor of Graduate School of Information Science, University of Tokyo at 1995. He also stayed at GETA (Groupe de Etude Traduction Automatique), CNRS, in Grenoble, France from 1981 to 1982 as visiting CNRS researcher. He is interested in Machine Translation, Natural Language Understanding and Artificial Intelligence. He is a member of International Committee of Computational Linguistics (ICCL) since 1992, and the program chairman of International Conference of Computational Linguistics 96 (Coling 96).