Automatic Recognition of Verbal Polysemy

Fumiyo FUKUMOTO * Jun'ichi TSUJII †
Centre for Computational Linguistics, UMIST
P.O.Box 88, Manchester M60 1QD, United Kingdom
E-mail fukumoto@ccl.umist.ac.uk tsujii@ccl.umist.ac.uk †

Abstract

Polysemy is one of the major causes of difficulties in semantic clustering of words in a corpus. In this paper, we first give a definition of polysemy from the viewpoint of clustering and then, based on this definition, we propose a clustering method which recognises verbal polysemies from a textual corpus. The results of experiments demonstrate the effectiveness of the proposed method.

1 Introduction

There has been quite a lot of research concerned with automatic clustering of semantically similar words or automatic recognition of collocations among them from corpora [Church, 1991], [Hindle, 1991], [Smadja, 1991]. Most of this work is based on similarity measures derived from the distribution of words in corpora. However, the facts that a single word does have more than one meaning and that the distribution of a word in a corpus is a mixture of usages of different meanings of the same word often hamper such attempts.

The *meaning* of a word depends on the domain in which it is used; the same word can be used differently in different domains. It is also often the case that a word which is polysemous in general is not polysemous in a restricted subject domain. In general, restriction of the subject domain makes the problem of polysemy less problematic. However, even in texts from a restricted domain such as *Wall Street Journal*[1], one encounters quite a large number of polysemous words. In particular, unlike nouns, verbs are often polysemous even in a restricted subject domain.

Because polysemous verbs are usually also high-frequency verbs, their treatment is crucial in actual applications. Furthermore, because of their high-frequency, polysemous verbs tend to have a harmful influence on the semantic clustering of nouns, because semantic clustering of nouns is usually performed based on their collocational behaviour with verbs.

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

2 Related Work

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

The techniques developed by Zernik [Zernik, 1991] and Brown [Brown, 1991] seem to cope with the discrimination of polysemy and be based on real texts. Zernik used monolingual texts which consist of about 1 million words tagged by part-of-speech. His method associates each word 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 solely relies on human intuition for identifying different senses of a word, i.e. 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 (*signatures*) that correspond to the different senses.

Brown used bilingual texts, which consist of 12 million words. The results of Brown's technique, when applied to a French-English machine translation system, seems to show its effectiveness and validity. However, as he admits, the approach is limited because it can only assign at most two senses to a word. More seriously, polysemy is defined in terms of translation, i.e. only when a word is translated into two different words in a target language, it is recognised as polysemous. The approach can be used only when a large parallel corpus is available. Furthermore, individual senses thus identified do not necessarily constitute single semantic units in the monolingual domain to which plausible semantic properties (i.e. semantic restrictions,

*FUKUMOTO is now at Department of Electrical Engineering and Computer Science, Faculty of Engineering, Yamanashi University. E-mail fukumoto@skye.esi.yamanashi.ac.jp

[1] *Wall Street Journal* was prepared by ACL(Association for Computational Linguistics' Data Collection Initiative) in 1991.

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 distributional behaviours of words in monolingual texts. The approach proposed in this paper focuses on this problem. Like Brown's approach, our approach adopts a relativistic view of polysemy. That is, a word is recognised as polysemous in terms of other related words. However, while Brown's approach identifies polysemous words in terms of related words of another language, we use semantically similar words of the same language to identify polysemous words. Whether a word is polysemous or not depends on whether a set of other, semantically similar words exists whose distributional behaviours correspond to a *subset* of the distributional behaviour of the 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 only by part-of-speech but not structurally annotated. However, the clustering algorithm, which automatically recognises polysemous words, only assumes that words are semantically characterised by a vector in an n-dimensional space so that it can be applied to any data satisfying this condition.

3 Polysemy in Context

The basic assumption of this work is the same as that made in previous corpus-based approaches, i.e. semantically similar words appear in a similar context. Semantically similar verbs, for example, co-occur with the same nouns. The following sentences from the *Wall Street Journal* corpus show the point:

(s1) New York Times said it offered to buy the shares of pop radio corp.
(s2) He may sell more shares in the open market or in private transactions.

It is intuitively obvious that buy and sell are semantically related and that the semantic closeness of these two verbs is manifested by the fact that they co-occur with the same noun shares. We can think of an n-dimensional space, each dimension of which is associated with a specific noun and in which a verb is assigned a vector whose value of the i-th dimension is the value of *mutual information* (*mu* in short) [Church, 1991] between the verb and the noun assigned to the i-th axis. If the basic assumption is correct, then semantically similar verbs form a cluster in the space, and therefore, statistical clustering algorithms can be applied to verb vectors in order to discover semantic classes of verbs.

However, this straightforward method is often hampered by the existence of polysemous words. The following sentences show polysemous usages of take.

(s3) In the past, however, coke has typically taken a minority stake in such ventures.
(s3') Guber and peters tried to buy a stake in mgm in 1988.
(s4) That process of sorting out specifics is likely to take time.
(s4') We spent a lot of time and money in building our group of stations.
(s5) People are queuing at the door to take his product but he doesn't have the working capital to make the thing.
(s5') 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 (s3) and (s3'), take and spend in (s4) and (s4'), take and obtain in (s5) and (s5') 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.

In the space spanned by the three axes, each associated with stake, time, and product, take does not constitute a cluster with any of the three verbs. 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 disappears.

In order 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 the three distinct usages of take. The decomposition of a vector into a set of its component vectors requires a proper decomposition of context in which the word occurs. Figure 1 shows the decomposition of the verb take in the three-dimensional spaces. **take1**, **take2**, and **take3** are the component vectors which collectively constitute the vector assigned to take.

For the sake of simplicity, we assume in the above 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 group of nouns and one has to divide the group of nouns into a set of subgroups, each of which correctly characterises the context for a specific sense of the polysemous word. The algorithm has to be able to determine when the context of a word should be divided and how.

There are clustering algorithms, called *overlapping clustering* [Jardine, 1991], which allow an entity to be-

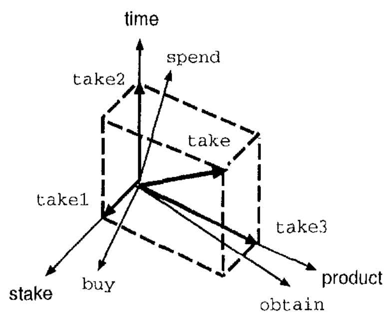

Figure 1: The decomposition of the verb take

long to more than one cluster. However, these algorithms assume that even an entity which belongs to more than one clusters is still a single entity. An entity belongs to several clusters because it can be seen from several different viewpoints. The same entity, for example, *egg*, can be seen as *food*, like *bread*, and as *ingredients-of-food*, like *flour*, at the same time.

However, as we saw in the above, 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.

Therefore, unlike standard overlapping clustering algorithms, our algorithm explicitly introduces new entities when an entity is judged polysemous and associates them with contexts which are subcontexts of the context of the original entity. Our algorithm has two basic operations, *splitting* and *lumping*. *Splitting* means to divide a polysemous verb into two hypothetical verbs and *lumping* means to combine two hypothetical verbs to make one verb out of them.

4 Measuring the Compactness of a Group of Verbs

The algorithm should decide when a verb has to be split into two hypothetical verbs. The decision is based on a measure of the semantic compactness of a group of verbs. The semantic compactness of a group of verbs is a measure which shows the degree of dispersion of the group in an n-dimensional space. The compactness of a group of verbs, VG= $\{v_1, v_2, \cdots, v_m\}$, is defined as follows.

1. Let v_i be one of the verbs $v_1, \cdots, v_m$, and a vector assigned to v_i be $(v_{i1}, \cdots, v_{in})$. Each $v_{ij} (1 \le j \le n)$ is computed by the following formula.

$$v_{ij} = \begin{cases} mu(v_i, n_j) & \text{if } mu(v_i,n_j) \ge \alpha \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

Here, $mu(v_i, n_j)$ is the value of *mutual information* defined in [Church, 1991] between v_i and n_j. α is a threshold value given in advance.

2. The centre of gravity of a group of verbs, $v_1, \cdots, v_m$, is the mean vector of the vectors assigned to the verbs, which is used to compute the dispersions of the individual verbs in the group. The centre of gravity $\bar{g} = (\bar{g_1}, \cdots, \bar{g_n})$, and the length of it $| \bar{g} |$, are defined as follows.

$$(\bar{g_1}, \cdots, \bar{g_n}) = (\frac{1}{m}\sum_{i=1}^{m} v_{i1}, \cdots, \frac{1}{m}\sum_{i=1}^{m} v_{in})$$

$$| \bar{g} | = \frac{1}{m}\sqrt{\sum_{j=1}^{n}(\sum_{i=1}^{m} v_{ij})^2} \quad (2)$$

3. The dispersion, $disp(v_1, \cdots, v_m)$, indicates the compactness of a group and is defined as:

$$disp(v_1, \cdots, v_m) = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}(v_{ij} - \bar{g_j})^2} \quad (3)$$

4. Let us think of two clusters of verbs, A and B, which have the same degree of dispersions. 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 co-occurrences of each noun and the verbs of A is larger than those of B; as a result, A should be judged to be semantically more compact than B. Therefore, the dispersion of (3) is normalised as:

$$disp_{nor}(v_1, \cdots, v_m) = \frac{disp(v_1, \cdots, v_m)}{| \bar{g} |} \quad (4)$$

5. $disp_{nor}$ of (4) is proportional to the number of verbs. This means that a cluster of a greater number of verbs tends to be judged to be less compact than those of a smaller number of verbs. Therefore, the dispersion of (4) should be further normalised to compensate the effect of the number of verbs in a group. This normalisation is done by *least square estimation*. The result is (5), which will be used to measure the compactness of a group of verbs.

$$Com(v_1, \cdots, v_m) = \frac{disp_{nor}(v_1, \cdots, v_m)}{(\beta * m - \gamma)} \quad (5)$$

$\beta * m - \gamma$ ($\beta = 0.964$, $\gamma = 0.495$) is a coefficient that is empirically determined by *least square estimation*[2].

In the following, we use (5) as the value which shows the compactness of a group. A group with a smaller value of (5) is judged *semantically more compact*.

[2] In this case, we set α in (1) equals to 3.0.

5 Clustering Method

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

5.1 The Basic Idea

The clustering algorithm proposed in this paper belongs to the overlapping type. The B_k ($k = 1,2,3,\cdots$) method, proposed by Jardine, is one of the typical overlapping clustering algorithms [Jardine, 1991]. The essential difference between our algorithm and the B_k method is that our algorithm explicitly introduces a condition when an entity (a verb) should be split and assigned to several clusters. In our method, whether a verb v has two senses or not is judged by comparing the semantic compactness values of groups of verbs to be produced. That is, there are possibilities of creating the following three clusters:

$$\{w_1, v_1\}, \{w_2, v_2\} \quad (6)$$

$$\{v, w_1, w_2\} \quad (7)$$

where v_1 and v_2 in (6) are new, hypothetical verbs which correspond to two distinct senses of the same verb, v. These two newly introduced verbs are supposed to appear in different contexts. Their contexts are actually hypothesised by dividing the set of nouns that co-occur with the verb v into two distinct sets of nouns. This division of the context of the original verb v is hypothesised based on the set of nouns that co-occurs with w_1 and the set of nouns that co-occurs with w_2.

5.2 *Splitting* and *Lumping*

The operations of *splitting* and *lumping* are defined as follows:

1. Function $split(v_i, v_p, v_q)$ returns $v\alpha$ and $v\beta$.

   v_i is a verb whose coordinate in an n-dimensional space is $(v_{i1}, \cdots, v_{in})$. $v\alpha$ and $v\beta$ are hypothesised verbs whose coordinates in the n-dimensional space are made from the coordinates of the original verb v_i by dividing the set of nouns that co-occur with v_i into two distinct sets. The division is made in terms of two sets of nouns: one is the set of nouns which co-occur with v_p, and the other is the set of nouns which co-occur with v_q.

$$\begin{aligned} split(v_i, v_p, v_q) &= (v\alpha, v\beta) \\ where\ Com(v_i, v_q) &\leq Com(v_i, v_p) \end{aligned} \quad (8)$$

$$v\alpha = \begin{bmatrix} v\alpha_1 \\ v\alpha_2 \\ \vdots \\ v\alpha_n \end{bmatrix} \text{ s.t. } v\alpha_j = \begin{cases} v_{ij} & \text{if } v_{pj} \neq 0 \\ 0 & \text{otherwise} \end{cases}$$

$$v\beta = \begin{bmatrix} v\beta_1 \\ v\beta_2 \\ \vdots \\ v\beta_n \end{bmatrix} \text{ s.t. } v\beta_j = \begin{cases} 0 & \text{if } (v_{qj} = 0 \text{ and } v_{pj} \neq 0) \\ v_{ij} & \text{otherwise} \end{cases}$$

   Note that if the noun associated with the dimension j which co-occurs with v_i also co-occurs with both of v_p and v_q, the values of the j-th dimension of $v\alpha$ and $v\beta$, ($v\alpha_j$ and $v\beta_j$), are the same value, i.e. the value of the mu between the noun associated with the j-th dimension and v_i. Furthermore, if the noun associated with the dimension j, which co-occurs with v_i, does not co-occur with both v_p and v_q, the value of the mu between the noun associated with the j-th dimension and v_i is set to the values of the j-th dimension of $v\beta$. Here, we call this value the surplus value. We recall that the compactness value of a group of v_i and v_q is smaller than that of v_i and v_p. This means that the former is more compact than the latter. If the surplus value is not set to both $v\alpha$ and $v\beta$, the group of $v\beta$ and v_q is more compact than that of $v\alpha$ and v_p. Therefore, in order to make $v\alpha$ and $v\beta$ as symmetrical as possible, the surplus value is set to $v\beta$.

2. Function $lump(v\alpha, v\beta)$ has the opposite effect of the function $split(v_i, v_p, v_q)$, i.e. it merges $v\alpha$ and $v\beta$. Function $lump(v\alpha, v\beta)$ returns v_i.

$$lump(v\alpha, v\beta) = v_i \quad (9)$$

$$v_i = \begin{bmatrix} v_{i1} \\ v_{i2} \\ \vdots \\ v_{in} \end{bmatrix} \text{ s.t. } v_{ij} = \begin{cases} v\alpha_j + v\beta_j & \text{if } v\alpha_j \neq v\beta_j \\ v\alpha_j & \text{otherwise} \end{cases}$$

5.3 Flow of the Algorithm

Given a group of verbs, $v_1, v_2, \cdots, v_m$, the algorithm produces a set of semantic clusters, which are ordered in the ascending order of their semantic compactness values. If v_i is non-polysemous, it belongs to at least one of the resultant semantic clusters. If it is polysemous, the algorithm splits it into several hypothetical verbs and each of them belongs to at least one of the semantic clusters. The flow of the algorithm is shown in Figure 2.

As shown in Figure 2, the algorithm is composed of three procedures: Make-Initial-Cluster-Set, Make-Temporary-Cluster-Set and Recognition-of-Polysemy.

1. Make-Initial-Cluster-Set

   The procedure Make-Initial-Cluster-Set produces all possible pairs of verbs in the input with their semantic compactness values. The result is a list

```
begin
  do Make-Initial-Cluster-Set
    for i (1 ≤ i ≤ m(m-1)/2)
      do Make-Temporary-Cluster-Set ;
      if a set of clusters which is retrieved by
          Make-Temporary-Cluster-Set exists
          then do Recognition-of-Polysemy ;
      end_if
      store the newly obtained cluster ;
      if the newly obtained cluster contains
          all the verbs in input
          then exit from the loop ;
      end_if
    end_for
end
```

Figure 2: The flow of the algorithm

of pairs which are sorted in the ascending order of their semantic compactness values. The list is called ICS (Initial Cluster Set). ICS contains $\frac{m(m-1)}{2}$ pairs. In the FOR-loop in the algorithm, a pair of verbs is retrieved from ICS, one at each iteration, and passed to the next two procedures.

2. Make-Temporary-Cluster-Set

   The procedure takes two arguments: The first argument is a pair of verbs from ICS and the second one is a set of clusters (CCS - Created Cluster Set). CCS consists of the clusters which have been created so far. When the algorithm terminates, CCS is the output of the algorithm. Make-Temporary-Cluster-Set retrieves the clusters from CCS which contain one of the verbs of the first argument (a pair from ICS). The clusters thus retrieved from CCS are passed to the next procedure for further consideration. If there is no CCS which contains one of the verbs of a pair from ICS, a pair of verbs from ICS is stored in CCS as a newly obtained cluster.

3. Recognition-of-Polysemy

   This procedure, which recognises a polysemous verb, also takes two arguments: the pair of verbs from ICS and a set of clusters retrieved by Make-Temporary-Cluster-Set.

   We recall the discussion in section 5.1. Let $\{v, w_1\}$ be the pair of verbs from ICS and $\{v, w_2\}$ be one of the clusters of the second argument, i.e. the clusters so far obtained which contain one of the verbs, v in the pair. We have to determine whether the verb v has two senses, which corresponds to w_1 and w_2, respectively. This is determined by comparing the semantic compactness values of the three different clusters shown in (6) and (7). The *splitting* function (8) is applied to v, w_1, and w_2 and produced newly hypothetical verbs, v_1 and v_2. The *lumping* function (9) is applied to v_1 and v_2 and makes one verb v out of them. If both of the semantic compactness values of each set shown in (6) are smaller than a set shown in (7), the sets (6) are selected, otherwise, (7) is selected and stored in CCS as a newly obtained cluster.

If the newly obtained cluster does not contain all the verbs in input, the next pair of verbs is taken from ICS, and then the whole process is repeated.

6 Experiments

We have conducted two experiments. The first experiment is concerned with the clustering technique and with verifying the effect of the proposed method. The second experiment is conducted to see how various part-of-speech pairs affect the clustering results.

6.1 Data for the Experiments

The corpus we have used is the *Wall Street Journal* which consists of 2,878,688 occurrences of part-of-speech tagged words [Church, 1991], 73,225 different words. From this corpus, we obtained 5,940,193 word pairs in a window size of 5 words, 2,743,974 different word pairs.

26 groups of verbs were used in the experiments, 108 verb tokens with 56 different original form of verbs. These groups contain 10 different polysemous verbs. The groups of verbs are divided into two different types, 'type1' and 'type2'; 'type1' is a set of verbs containing one or more polysemous verbs, and 'type2' does not contain any polysemous verbs. Each group is composed of 3 to 10 different verbs. The selection of verbs of 'type1' was made with the intention of processing verbs with wide usages, as identified in the Collins dictionary and thesaurus [McLeod, 1991]. Then, a number of synonyms of the chosen verbs were selected from the thesaurus. The clustering analysis is applied to each group of verbs. The same corpus and the groups of verbs are used throughout the experiments.

6.2 Experiment-I

In Experiment-I, we used verb-noun pairs, i.e. we assume an n-dimensional 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 the small window sizes, the semantic relationships between these two words might be quite strong, especially those between a verb and its object which permits the effective recognition of verbal polysemy. The inflected forms of the same nouns and verbs are treated as single units. For example, 'time'(noun, singular) and 'times'(noun, plural) are treated as single units. We obtained 228,665 different verb-noun pairs from 2,743,974 and from these, we selected 6,768 different verb-noun pairs, 701 different verbs and 1,796 nouns on condition that frequencies and mu are not low ($N_{xy} \geq 5$, $mu(x, y) \geq 3$) to permit a reliable statistical analysis and used them in the experiment[3]. The results are shown in Table 1.

Table 1: The results of Experiment-I

	group	correct	incorrect
type1	14	9	5
type2	12	9	3
total(%)	26( )	18(69.2)	8(30.8)

In Table 1, 'group' means the number of each group, type1 and type2; 'correct' means the number of groups of verbs which are clustered correctly; 'incorrect' means that they are not. Figure 3 shows each sample of the results, i.e. **type1-correct**, **type2-correct**, **type1-incorrect**, and **type2-incorrect**. Each value in Figure 3 shows the value of the semantic compactness of a group of verbs.

In Figure 3, under the heading **type1-correct**, we can see that 'take' is recognised as a polysemous verb and has three different senses, 'spend', 'buy', and 'obtain'. In a similar way, 'close' has two different senses, 'end' and 'open' and semantically close verbs are grouped together. Under the heading **type2-correct** semantically similar verbs are grouped together. However, under the heading **type1-incorrect** 'leave' is incorrectly recognised as a non-polysemous verb; also under the heading **type2-incorrect** 'come' is incorrectly recognised as a polysemous verb.

6.3 Experiment-II

We have conducted an experiment using the various parts-of-speech shown in Table 2.

Table 2: The type and the number of pairs

x-y	pair(1)	pair(2)	x	y
noun-verb	250,732	6,420	1,993	565
verb-adverb	23,248	1,200	250	320
adverb-verb	35,146	902	163	377
verb-preposition	29,658	3,197	1,338	58

In Table 2, x-y shows the type of part-of-speech pair of x and y in this order, where x and y are the part-of-speech of the words. 'pair(1)' shows the number of different part-of-speech pairs from 2,743,974 and 'pair(2)' shows the number of different part-of-speech pairs on condition that frequencies and mu are $N_{xy} \geq 5$, $mu(x, y) \geq 3$; x and y show the number of different word. We used these in Experiment-II. The results are shown in Table 3.

[3] Here, N_{xy} is the number of total co-occurrences of the words x and y in this order in a window.

Table 3: The results of Experiment-II

x-y	correct(%)	incorrect(%)
noun-verb	10(38.5)	16(61.5)
verb-adverb	5(19.2)	21(80.8)
adverb-verb	6(23.0)	20(77.0)
verb-preposition	6(23.0)	20(77.0)

7 Discussion

In Experiment-I, described in the previous section, 18 out of 26 groups of verbs are analysed correctly and the percentage attains 69.2 % in all. However, as shown in Table 1, there are 8 groups which could not be recognised correctly. The errors are classified into two types:[4] 1. Errors of recognition of polysemous verbs as non-polysemous ones; and 2. Errors of recognition of non-polysemous verbs as polysemous ones. The number of groups classified into each error type is 4 and 7, respectively. The cause of these errors is that co-occurring nouns shared by two verbs seem to be slanted in these data. For example, observing the corpus, we can see that 'leave' has at least two senses, 'retire' and 'remain'. The following sentences are from the *Wall Street Journal*.

(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. This error was caused by the fact that the value of the semantic compactness of 'retire' and 'remain' was smaller than that of any other pair of words and by the fact that the cardinality of a set of nouns which co-occur with 'retire' and 'remain' is larger than that of any other pair of words. We provisionally conclude that the use of verb-noun pairs alone is not appropriate for all the groups of verbs.

In Experiment-II, the overall results are not as good as those of Experiment-I. However, we could observe some interesting characteristics, namely, some groups which could not be analysed correctly by using verb-noun pairs could be analysed correctly by using verb-adverb pairs or verb-preposition pairs. The results show that 3 out of 8 groups such as **type1-incorrect** in Figure 3 which were incorrect in Experiment-I could be analysed correctly by using verb-adverb pairs. Also, an other 3 groups such as **type2-incorrect** could be analysed correctly by using verb-preposition pairs. We

[4] We do not consider here **general errors of semantic clusters**, i.e. the case of two verbs which are not semantically close but are judged to constitute a semantic cluster. Because this kind of error did not occur in the current experiments.

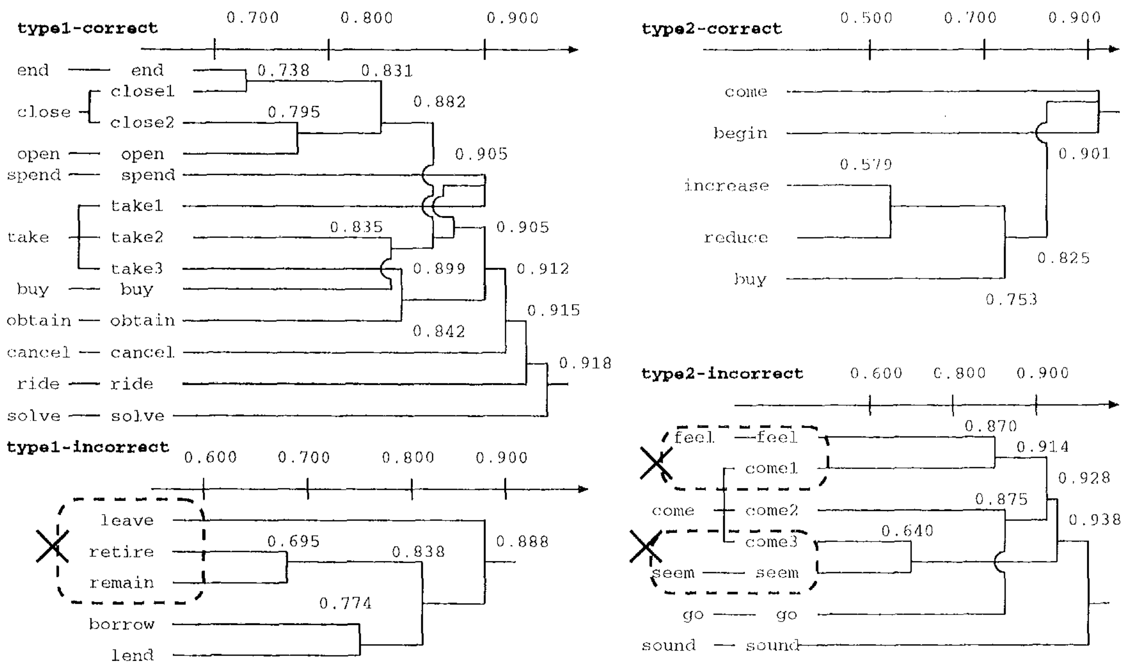

Figure 3: The results of the clustering analysis

can therefore expect that we may be able to obtain more accurate clusters by merging these three kinds of part-of-speech pairs into one larger set. Because these three different pairs show distinct characteristics of contexts in which a verb appears. We have been conducting more experiments on these.

8 Conclusion

We have given a definition of polysemy from the viewpoint of clustering, and proposed an overlapping clustering method which *automatically* recognises verbal polysemies from a textual corpus. The significant feature of our approach is that every separate meaning of a word is recognised in terms of other words that appear in the corpus. Whether a word is polysemous or not depends on whether a set of other words exists whose usage corresponds to one of the meanings of a polysemous word. As a result, our method can avoid human intuition in the judgement of distinct word meanings and thus, human intervention.

The results of the experiments demonstrate the applicability of automatic method of recognition of polysemous verbs. We have conducted more experiments by changing parameters such as the threshold values for frequencies (N_{xy}) and mu ($mu(x,y)$) in order to see how these parameters affect the performance of the clustering algorithm. We have also extended our technique to the disambiguation of word senses. We hope to report these results soon.

References

Church, Kenneth W. et al.. 1991. Using Statistics in Lexical Analysis. *Lexical acquisition: Exploiting on-line resources to build a lexicon.* (Zernik Uri (ed.)), pages 115-164, London, Lawrence Erlbaum Associates.

Hindle Donald. 1990. Noun classification from predicate-argument structures. In *28th Annual Meeting of the Association for Computational Linguistics*, pages 268-275, Pittsburgh, Pennsylvania. Association for Computational Linguistics, Morristown, New Jersey.

Smadja, Frank A.. 1993. Retrieving Collocations from Text: Xtract. In *Computational Linguistics*, pages 143-177, 19 (1993) 1.

Zernik Uri. 1991. Train1 vs. Train2: Tagging Word Senses in Corpus, *Lexical acquisition: Exploiting on-line resources to build a lexicon*, Uri Zernik(Ed.), pages 91-112, London, Lawrence Erlbaum Associates.

Brown, Peter F. et al.. 1991. Word-Sense Disambiguation Using Statistical Methods. In *Proc. of the 29th Annual Meeting of the ACL*, pages 264-270.

Jardine N. and Sibson R.. 1968. The construction of hierarchic and non-hierarchic classifications. In *Computer Journal*, pages 177-184.

Church, Kenneth W.. 1988. A stochastic parts programme and noun phraser for unrestricted text. In *Proc. of the 2nd conference on applied natural language processing*, pages 136-143.

McLeod, William T.. 1987. *The new Collins dictionary and thesaurus in one volume*, London, HarperCollins Publishers.