# Document Clustering with Semantic Analysis

Yong Wang[1] and Julia Hodges[2]
*Department of Computer Science & Engineering, Mississippi State University*
*Mississippi State, MS 39762-9637*
*[1]ywang@cse.msstate.edu, [2]hodges@cse.msstate.edu*

## Abstract

*Document clustering generates clusters from the whole document collection automatically and is used in many fields, including data mining and information retrieval. In the traditional vector space model, the unique words occurring in the document set are used as the features. But because of the synonym problem and the polysemous problem, such a bag of original words cannot represent the content of a document precisely. In this paper, we investigate using the sense disambiguation method to identify the sense of words to construct the feature vector for document representation. Our experimental results demonstrate that in most conditions, using sense can improve the performance of our document clustering system. But the comprehensive statistical analysis performed indicates that the differences between using original single words and using senses of words are not statistically significant. In this paper, we also provide an evaluation of several basic clustering algorithms for algorithm selection.*

## 1. Introduction

Data clustering partitions a set of unlabeled objects into disjoint/joint groups of clusters. In a good cluster, all the objects within a cluster are very similar while the objects in other clusters are very different. When the data processed is a set of documents, it is called document clustering. Document clustering is very important and useful in the information retrieval area. Document clustering can be applied to a document database so that similar documents are related in the same cluster. During the retrieval process, documents belonging to the same cluster as the retrieved documents can also be returned to the user. This could improve the recall of an information retrieval system. Document clustering can also be applied to the retrieved documents to facilitate finding the useful documents for the user. Generally, the feedback of an information retrieval system is a ranked list ordered by their estimated relevance to the query. When the volume of an information database is small and the query formulated by the user is well defined, this ranked list approach is efficient. But for a tremendous information source, such as the World Wide Web, and poor query conditions (just one or two key words), it is difficult for the retrieval system to identify the interesting items for the user. Sometimes most of the retrieved documents are of no interest to the users. Applying documenting clustering to the retrieved documents could make it easier for the users to browse their results and locate what they want quickly. A successful example of this application is VIVISIMO (http://vivisimo.com/), which is a Web search engine that organizes search results with document clustering.

The vector space model is a widely used method for document representation in information retrieval. In this model, each document is represented by a feature vector. The unique terms occurring in the whole document collection are identified as the attributes (or features) of the feature vector. Different term weighting methods may be used in the vector space model, such as the binary method, *tf* (term frequency) method [32], and *tf-idf* (Inverse document frequency) method [31].

Traditionally, the single words or compound words occurring in the document set are used as the features. Because of the synonym problem and polysemous problem, generally such a "bag of words" cannot reflect the semantic content of a document [14]. In this paper, we investigate using semantic analysis to identify the semantic content of documents. In our system, words are first mapped to word senses using a semantic relatedness based word sense disambiguation algorithm. Then these senses were used to construct the feature vector to represent the documents. Different semantic relatedness measures were evaluated in our experiments. Since the content of a document is directly related to the sense of its contained words, one can expect to capture the content of a document more precisely when the senses are used. One important prerequisite for semantic analysis is the construction of a large-scale thesaurus and dictionary. Currently, one of the most widely

0-7695-2507-5/06/$20.00 (C) 2006 IEEE

used thesaurus for English is WordNet [9, 23] and it was also used in our work.

The rest of this paper is organized as follows: A brief introduction to some basic clustering algorithms is described in section 2. In section 3 and section 4, a brief survey of current research in word sense disambiguation and semantic relatedness measures is provided. We present our experimental data, evaluation methods, and experimental results in section 5. Section 6 lists our final conclusions and describes promising future work.

## 2. Document Clustering Algorithms

Generally, clustering algorithms can be divided into hierarchical methods, partitioning methods, and other methods. Hierarchical clustering generates a hierarchical tree of clusters. This tree is also called a dendrogram [4]. Hierarchical methods can be further classified into agglomerative methods and divisive methods. In an agglomerative method, originally, each object forms a cluster. Then the two most similar clusters are merged iteratively until some termination criterion is satisfied. In a divisive method, from a cluster which consists of all the objects, one cluster is selected and split into smaller clusters recursively until some termination criterion is satisfied. Steinbach, Karypis, and Kumar compared the performance of three agglomerative clustering algorithms, IST (Intra-Cluster Similarity Technique), CST (Centroid Similarity Technique), and UPGMA [34]. Their experimental results show UPGMA is the best of the three.

Partitioning clustering methods allocate data into a fixed number of non-empty clusters. All the clusters are in the same level. The most well-known partitioning methods following this principle are the K-means method and its variants. The basic K-means method initially randomly allocates a set of objects into a number of clusters. In every iteration, the mean of each cluster is calculated and each object is re-assigned to the nearest mean. This loop will stop when there is no change for any of the clusters. The use of the K-means method for document clustering can be found in [3, 16]. Some variants of K-means methods include the K-medoids method [14] and global k-means method [21].

The buckshot method is a combination of the K-means method and HAC method. In buckshot, $\sqrt{n}$ objects are selected randomly as a sample set of the whole collection. The HAC method is applied on the sample set. The centers of the K clusters on the sample set are the initial seeds for the whole collection. The K-means iterations are performed again to partition the whole collection. The buckshot method is successfully used in a well-known document clustering system, the Scatter/Gather (SG) system [7].

The K-means method can also be used to generate hierarchical clusters. Steinbach, Karypis, and Kumar proposed bisecting K-means algorithm to generate hierarchical clusters by applying the basic K-means method recursively [34]. The bisecting K-means algorithm is a divisive hierarchical clustering algorithm. Initially the whole document set is considered one cluster. Then the algorithm recursively selects the largest cluster and uses the basic K-means algorithm to divide it into two sub-clusters until the desired number of clusters is reached.

Besides these basic clustering algorithms, some algorithms designed especially for document clustering have been proposed. Zamir has described the use of a suffix tree for document clustering [38]. Beil, Ester, and Xu proposed two clustering methods, FTC (Frequent Term-based Clustering) and HFTC (Hierarchical Frequent Term-based Clustering), based on frequent term sets [2]. Fung proposed another hierarchical document clustering method based on the frequent term set, HIFC (Frequent Itemset-based Hierarchical Clustering), to improve the HFTC method [11]. Hammouda proposed an incremental clustering algorithm by representing each cluster with a similarity histogram [13]. Weiss, White, and Apte described a lightweight document clustering method using nearest neighbors [36]. Lin and Ravikumar described a soft document clustering system called WBSC (Word-based Soft Clustering) [22].

## 3. Word Sense Disambiguation

The problem of finding the correct sense of a word in a context is called word sense disamiguation (WSD) [19]. There are two general types of methods for sense determination [10]. Some methods used the definition of each sense in a dictionary. In WordNet, the definition is the gloss assigned to each synset. Other approaches used the semantic relatedness in an existing semantic network.

Lesk proposed to disambiguate the sense of a polysemous word based on its context words and the definition of all senses in a dictionary [19]. For each sense of a word, a lexicon definition can be found in a dictionary. All the words occurring in this definition compose the sense bag for this sense. All the words occurring in the definitions of the senses of the context words compose the context bag for this word. The context bag of a word will be compared to each

sense bag of all candidate senses. The sense with the maximum match will be selected.

Fragos, Maistros, and Skourlas [10] proposed to improve Lesk's method by enriching the sense bag and the context bag. For each word in the sense bag, all the words that have a "hypernymy/hyponymy" relationship with it are also identified. The definitions of these newly identified words are incorporated into the sense bag. For the context words, all the words that have a "hypernymy" relationship with the current context word and their definitions are added into the context bag.

Gomes et al. [12] present a word sense disambiguation method based on the semantic distance among the context words in WordNet. Given a set of context words, $\{W_1, W_2, \ldots W_n\}$, from WordNet, we can get a set of senses for each word. Suppose the number of senses for word $i$ is $s_i$, then we get a set of senses $\{S(1, 1), S(1, 2), \ldots, S(1, s_1), S(2, 1), S(2, 2), \ldots, S(2, s_2), \ldots\ldots, S(n, 1), S(n, 2), \ldots S(n, s_n)\}$. With different measures, we can get the semantic distance between two senses, $SemanticDist(S(i, j), S(k, l))$. The shortest semantic distance between a sense $S(i, j)$ and a word $W_k$ is:

$$ShortestDist(S(i, j), W_k) = \min_{1<l<sl} \{SemanticDist(S(i, j), S(k, l))\}$$

For this sense, its synset score is defined as the sum of all shortest semantic distances between this sense and all the other words. The synset score reflects the context semantic distance between this sense and all context words.

$$SynsetScore(S(i, j)) = \sum_{k=1}^{n} ShortestDist(S(i, j), W_k)$$

Then for each word, the sense which has the minimum synset score will be selected as its sense.

$$Sense(W_k) = \min_{1<l<sl} \{SynsetScore(S(k, l))\}$$

Sussna [35] proposed a similar disambiguation method based on semantic distance. Instead of finding the shortest distance between one sense and one word, Sussna's method just uses all senses and tries to find a combination of candidate senses to achieve the minimum total pairwise semantic distance among the selected senses.

Li, Szpakowicz, and Matwin [20] proposed eight heuristic rules for sense disambiguation based on WordNet semantic relatedness. The basic idea in this method is try to find verb-noun pairs in the context with semantic relatedness in WordNet. Those related verb-noun pairs were used to determine the sense for each other.

Ramakrishnanan and Bhattacharyya [27] also proposed to use the synsets in WordNet to replace the original word for text representation. But they are using a soft sense disambiguation method in which each word will be assigned several senses instead of a single sense. Three algorithms, hubs and authorities, page ranking, and Bayesian inferencing, were described to rank the candidate synsets.

## 4. Semantic Relatedness Measures

A semantic relatedness measure is a criterion to scale the relatedness of two senses in a semantic network. It is also called semantic distance or semantic similarity in the literature. In a lot of word sense disambiguation algorithms, a semantic relatedness measure is a very important factor for the performance. Budanitsky presents a comprehensive overview of various semantic relatedness measures [5, 6]. Jiang and Conrath classify those methods into two categories, edge-based methods and node-based (information content-based) methods [17]. Edge-based methods attempt to measure the distance between two senses according to the length of the path between them in the semantic networks. The simplest method is to count the number of edges or nodes between them. Some others include Hso method [15], Lch method [18], Sussna's method [35], and Wup method [38]. Node-based methods measure the distance between two senses according to the statistical information contained in the nodes within the semantic network. There are many different methods used to calculate the information based on a corpus. Some generally used node-based methods include Res method [28, 29] and Lin method [8]. Jcn method is a combined method which considers both edge and node information [17]. Some other methods include Banerjee and Pedersen's method [1, 25], and Patwardhan's method [24]. WordNet::Similarity is a perl software package which consists of several sub-modules to implement different semantic relatedness measures [26]. Table 1 lists all these modules, their references, and their types.

## 5. Experiments

### 5.1 Experimental Data

Since the procedure of sense disambiguation is very time-consuming, we have to make the size of our data set to be relatively small. We collected 1,600

abstracts from journals belonging to ten different areas. For each area, 160 abstracts were collected. Table 2 lists the areas and the names of the journals. This data set was divided evenly into eight subsets and each subset contains 200 abstracts. Within each subset, there are exactly 20 abstracts from each of the 10 categories. These datasets are named DS 1 - 8.

**Table 1: Modules in WordNet::Similarity package**

| Module | Reference | Type |
|---|---|---|
| path | Counting the nodes in the path | Edge-based |
| hso | Hirst and St-Onge (1998) | Edge-based |
| lch | Leacock and Chodorow(1998) | Edge-based |
| wup | Wu and Palmer(1994) | Edge-based |
| res | Resnik (1995) | Node-based |
| lin | Lin (1998) | Node -based |
| jcn | Jiang and Conrath (1997) | Edge & Node |
| lesk | Banerjee and Pedersen (2002) | Using gloss |
| vector | Patwardhan (2003) | Context vector |
| random | Using a random measure | |

All these abstracts were cut into separate sentences with MXTERMINATOR, which is a JAVA tokenizer implemented by Adwait Ratnaparkhi [30]. This tool can be used to identify the sentence boundaries and separate the documents into sentences. The Penn Treebank tokenizer was used to identify the tokens from each sentence. The lemmatizer in WordNet was used to convert each token into its lemma [23]. Then all the stop words are filtered and the MXPOST (Maximum Entropy POS Tagger) part-of-speech tagger was used to assign a part-of-speech tag to each lemma. Finally, a document is converted into a list of lemmas with its POS tag. In our work, the sense of these lemmas was identified to construct the feature vector for each document.

**Table 2: Journal abstracts data set**

| Area | size | Journal Name |
|---|---|---|
| AI | 160 | Artificial Intelligence |
| Ecology | 160 | Journal of Ecology |
| Economy | 160 | Economic Journal |
| History | 160 | Historical Abstracts |
| Linguistics | 160 | Journal of Linguistics |
| Material | 160 | Journal of Electronic Materials |
| Nuclear | 160 | IEEE Transactions on Nuclear Science |
| Proteomics | 160 | PubMed |
| Sociology | 160 | Journal of Sociology |
| Statistics | 160 | Journal of Applied Statistics Regression Analysis |

## 5.2 Evaluation Methods

The evaluation methods entropy and F-measure, which have been used by a number of researchers, including Steinbach, Karypis, and Kumar [34], were used. Both entropy and F-measure are external quality measures.

Given a set of labeled documents belonging to $I$ classes, assume the clustering algorithm partitions them into $J$ clusters. Let $n$ be the size of the document set; $n_i$ be the size of class $i$; $n_j$ be the size of cluster $j$; and $n_{ij}$ be the number of documents belonging to both class $i$ and cluster $j$. Then for a document in cluster j, the probability that it belongs to class $i$ is $P(i,j) = n_{ij}/n_j$.

The entropy of cluster $j$ is:

$$E(j) = -\sum_{i=1}^{I} P(i,j)\log_2 P(i,j)$$

The entropy of all the clusters is the sum of the entropy of each of the clusters weighted by its size:

$$E = \sum_{j=1}^{J} \frac{n_j}{n} E(j)$$

For F-measure, the F-measure value of each class is calculated first, then they are combined to get the F-measure of the entire set. Given a cluster $j$ and a class $i$, assume cluster $j$ is the retrieval results of class $i$. Then the recall, precision, and F-measure of this retrieval are:

$$Recall(i, j) = \frac{n_{ij}}{n_i}, \; Precision(i, j) = \frac{n_{ij}}{n_j}$$

$$F(i, j) = \frac{2 \times Recall\ (i, j) \times Precision(i,j)}{Recall(i, j)\ +\ Precision(i, j)}$$

Since there is no one-to-one mapping relationship between each class and each cluster, any cluster can be considered the candidate retrieval result of a class. The best F-measure among all the clusters is selected as the F-measure for the query of a particular class:

$$F(i) = MAX_{0<j<J} F(i, j)$$

The F-measure of all of the clusters is the sum of the F-measures of each class weighted by its size:

$$\mathrm{F} = \sum_{i=1}^{I} \frac{n_i}{n} F(i)$$

In order to determine whether the difference between two methods is statistically significant or just a random variance, some statistical analysis methods should be adapted to evaluate the comparison results. The *t*-test and Wilcoxon signed rank test [37] are two of the most widely used methods for paired data. In

both of them, the null hypothesis $H_0$ is that two groups of data sets have the same median, which means the distribution of their data are same. There are two prerequisites for the data in order to use the *t*-test. The first the one is the population of data set should be large enough. Generally it should be larger than 20. The second one is that the variance of the data should satisfy a normal distribution. For our experiments, we have eight small data sets to do the comparison, which is too small for the *t*-test. Additionally our experimental results are observed from different data sets and do not satisfy the normal distribution. Because of these reasons, the Wilcoxon signed rank test was selected for statistical analysis in our research work. In Wilcoxon signed rank test, in order to reject the original hypothesis, a small W should be achieved. The detailed method for calculating W value can be found in [37]. For our datasets, since the size of our samples is 8, a W value equal or smaller than 6 should be achieved to prove that the difference is significantly different.

**Table 3: Comparison of clustering algorithms**

| Data Set | DS 1 | DS 2 | DS 3 | DS 4 | DS 5 | DS 6 | DS 7 | DS 8 |
|---|---|---|---|---|---|---|---|---|
| | F-measure | | | | | | | |
| K-means | 0.45 | 0.45 | 0.41 | 0.42 | 0.40 | 0.44 | 0.43 | 0.37 |
| Buckshot | 0.48 | 0.54 | 0.48 | 0.48 | 0.48 | 0.49 | 0.50 | 0.45 |
| Bisecting K-mean | 0.35 | 0.38 | 0.34 | 0.35 | 0.36 | 0.35 | 0.37 | 0.37 |
| HAC | **0.69** | **0.74** | **0.70** | **0.64** | **0.70** | **0.78** | **0.69** | **0.64** |
| | Entropy | | | | | | | |
| K-means | 1.46 | 1.41 | 1.55 | 1.54 | 1.55 | 1.47 | 1.47 | 1.63 |
| Buckshot | 1.34 | 1.20 | 1.39 | 1.38 | 1.37 | 1.37 | 1.32 | 1.45 |
| Bisecting K-mean | 1.65 | 1.56 | 1.72 | 1.67 | 1.65 | 1.69 | 1.62 | 1.64 |
| HAC | **0.74** | **0.51** | **0.80** | **0.79** | **0.75** | **0.60** | **0.77** | **0.96** |

## 5.3 Experimental Results and Analysis

*Comparison of clustering algorithms*

Four basic clustering algorithms, K-means, buckshot, HAC, and bisecting K-means, were selected for comparison. In this experiment, K-means method, buckshot method, and bisecting K-means method are executed 20 times to alleviate the effect of a random factor. The original words were used to construct the feature vector in this experiment. All the results are listed in table 3. The F-measure values and entropy values listed here are the average values of 20 different results.

In all eight data sets, the HAC method outperforms all the other methods for both F-measure and entropy. The results of the K-means method, buckshot method, and bisecting K-means method are quite similar to each other. Note that for all eight datasets, the performance sequence of four methods is same. When the Wilcoxon signed rank test is applied for each pair of these methods, we got the W value to be 0, which indicates that the different between them are statistically significant. This result is different from that of Steinbach, Karypis, and Kumar [34]. In Steinbach, Karypis, and Kumar's experiments, agglomerative hierarchival clustering performs poorly and bisecting K_means method achieved the best performance. The analysis provided by Steinbach, Karypis, and Kumar explains that in the document clustering problem, the nearest neighbors of a document belong to different clusters in most condition. This is the "nature of documents" [34]. The global adjusting function of K-means method can compensate this weakness. But in the HAC method, once two clusters were merged together, it will never be re-assigned.

**Table 4: Clustering results of large data set**

| | F-measure |
|---|---|
| K-means | 0.72 |
| Buckshot | 0.71 |
| Bisecting K-mean | **0.88** |
| HAC | 0.67 |
| | Entropy |
| K-means | 0.72 |
| Buckshot | 0.72 |
| Bisecting K-mean | **0.46** |
| HAC | 0.87 |

Then why does the HAC method work better here? We think the reason is the size of the data set. Generally, two documents sharing more of the same words are considered more similar with each other. When the size of the document set is larger, the problem caused by the "nature of documents" becomes more serious. In Steinbach, Karypis, and Kumar's experiments, there are eight data sets used for evaluation [34]. The largest data set contains about 3000 documents and the smallest one contains about 1000 documents. But in our dataset, there are only

200 documents in each of them. For a small data set, there are fewer shared words between documents. Thus it is more possible that two documents with shared words belong to the same cluster. This makes the HAC method the best one for small data sets. In order to verify our hypothesis, we combined all ten datasets into a large data set with 1600 documents. Four clustering algorithms were executed again on this large set. The result is the list in table 4. For this large data set, the bisecting K-means method got the highest F-measure and lowest entropy. On the contrary, HAC had the poorest performance. This result is consistent with that of Steinbach, Karypis, and Kumar's experiments [34].

**Table 5: Experimental results of using word sense**

| Data Set | DS 1 | DS 2 | DS 3 | DS 4 | DS 5 | DS 6 | DS 7 | DS 8 |
|---|---|---|---|---|---|---|---|---|
| | F-measure | | | | | | | |
| Original Form | 0.69 | 0.74 | ***0.70*** | 0.64 | 0.70 | 0.78 | 0.69 | 0.64 |
| Jcn-offset | **0.72** | **0.79** | 0.65 | 0.57 | **0.79** | 0.66 | **0.80** | 0.63 |
| Jcn-senseno | **0.79** | 0.73 | 0.67 | 0.62 | 0.78 | 0.77 | **0.80** | **0.69** |
| Lch-offset | **0.77** | **0.80** | 0.66 | **0.67** | **0.78** | 0.71 | **0.77** | **0.69** |
| Lch- senseno | **0.71** | **0.80** | 0.66 | 0.60 | ***0.85*** | 0.75 | **0.80** | **0.70** |
| Path-offset | **0.78** | **0.80** | 0.67 | **0.67** | **0.76** | 0.70 | **0.70** | **0.74** |
| Path- senseno | **0.72** | **0.80** | 0.65 | 0.60 | **0.83** | 0.76 | **0.75** | 0.60 |
| Wup-offset | **0.78** | **0.77** | 0.65 | 0.59 | 0.66 | 0.69 | **0.79** | 0.61 |
| Wup- senseno | **0.77** | **0.79** | 0.66 | 0.51 | **0.73** | 0.77 | **0.79** | ***0.76*** |
| Res-offset | 0.60 | 0.69 | 0.65 | 0.61 | **0.79** | 0.70 | 0.69 | **0.66** |
| Res- senseno | **0.70** | **0.81** | 0.57 | 0.59 | **0.74** | 0.73 | 0.65 | **0.66** |
| Lin-offset | **0.70** | **0.81** | 0.61 | 0.57 | **0.77** | 0.71 | ***0.80*** | **0.69** |
| Lin- senseno | ***0.79*** | ***0.81*** | 0.62 | 0.57 | **0.75** | 0.76 | **0.79** | **0.69** |
| Lesk-offset | **0.78** | **0.78** | 0.67 | ***0.67*** | **0.74** | ***0.78*** | **0.75** | 0.57 |
| Lesk- senseno | 0.65 | **0.81** | 0.66 | 0.61 | **0.73** | 0.77 | **0.77** | **0.66** |
| Random-offset | 0.69 | 0.71 | 0.52 | 0.61 | 0.58 | 0.72 | **0.77** | 0.62 |
| Random- senseno | 0.69 | 0.74 | 0.46 | 0.62 | 0.67 | 0.72 | **0.78** | 0.56 |
| | Entropy | | | | | | | |
| Original Form | 0.74 | 0.51 | 0.80 | ***0.79*** | 0.75 | 0.60 | 0.77 | 0.96 |
| Jcn-offset | **0.65** | 0.52 | 0.82 | 1.04 | **0.51** | 0.83 | **0.54** | **0.93** |
| Jcn-senseno | ***0.51*** | 0.59 | **0.79** | 0.90 | **0.57** | ***0.53*** | **0.57** | **0.80** |
| Lch-offset | **0.62** | **0.49** | 0.83 | 0.81 | **0.57** | 0.76 | **0.61** | **0.81** |
| Lch- senseno | 0.70 | **0.46** | 0.81 | 0.93 | ***0.42*** | 0.69 | ***0.53*** | **0.76** |
| Path-offset | **0.56** | **0.49** | 0.81 | 0.81 | **0.62** | 0.77 | 0.77 | **0.70** |
| Path- senseno | **0.70** | **0.48** | 0.85 | 0.93 | **0.50** | 0.66 | **0.61** | 0.98 |
| Wup-offset | **0.53** | 0.58 | 0.82 | 0.99 | 0.84 | 0.80 | **0.57** | 1.01 |
| Wup- senseno | **0.56** | 0.52 | ***0.79*** | 1.09 | **0.66** | 0.60 | **0.55** | ***0.68*** |
| Res-offset | 0.93 | 0.66 | 0.83 | 0.92 | **0.53** | 0.77 | 0.81 | **0.94** |
| Res- senseno | 0.72 | ***0.44*** | 0.99 | 0.99 | **0.62** | 0.64 | 0.86 | **0.88** |
| Lin-offset | **0.71** | **0.45** | 0.91 | 0.97 | **0.61** | 0.70 | **0.54** | **0.83** |
| Lin- senseno | **0.56** | **0.46** | 0.89 | 0.96 | **0.66** | 0.61 | **0.58** | **0.84** |
| Lesk-offset | **0.55** | 0.56 | 0.83 | 0.80 | **0.64** | **0.55** | **0.71** | 1.13 |
| Lesk- senseno | 0.79 | **0.45** | 0.83 | 0.97 | **0.68** | **0.59** | **0.62** | **0.82** |
| Random-offset | **0.74** | 0.60 | 1.13 | 0.98 | 1.00 | 0.73 | **0.62** | **0.95** |
| Random- senseno | 0.79 | 0.60 | 1.26 | 0.92 | 0.83 | 0.75 | **0.63** | 1.09 |

*Using sense of words*

The WordNet semantic network is used for sense disambiguation in our clustering system. Gomes et al.'s context words-based sense disambiguation method is used to find the sense for each word [12]. A context window is set to control the size of context. In our experiments, the window size is set to be 3, which means that only the two words adjacent to the current word will be used as context. Since Gomes et al.'s method is based on the semantic relatedness measure, all the semantic relatedness measures implemented in Pedersen's WordNet::Similarity package were evaluated.

The sense of a word is represented in two formats. The first one is using the number of senses in WordNet. For example, the noun 'course' has eight senses in WordNet. If the first sense is selected, then it will be represented as 'course#n#1'. This representation is helpful for the polysemy problem. In

WordNet, each synset has a unique offset in the database. This offset can be used as the unique ID for this sense. All the synonyms in this synset share the same offset. For example, the offset of the first sense of noun 'course' is 00831838. The offset representation format is helpful for both the synonym problem and the polysemous problem.

Table 5 lists the detailed experimental results for each data set. For each different semantic relatedness measure, the F-measure and entropy for two sense representation methods are also provided. In this table, if the result of using a sense form is better than the corresponding word form, a bold font is used.

The first comparison was to decide whether the use of sense can improve the clustering performance. We counted the number of data sets in which the use of the sense form achieves better performance than using the original word form. If the number of data sets is more than four, which is half of the whole collection, then we expect the use of sense can improve the performance for most data sets. The count results are listed in Table 6. When the 'offset' representation method was used, for both F-measure and entropy, within eight different semantic relatedness measures, five of them had better performance than the original form. When the 'senseno' representation method was used, six of them had better performance than the original form. This demonstrates that in most conditions, using sense to replace the original word can improve the performance of our document clustering system. Within eight different semantic relatedness measures, the *Random* measure had the worst measure. The performances of other measures were fairly good.

**Table 6: The number of datasets in which sense forms get better performance**

| | Offset | | SenseNo | |
|---|---|---|---|---|
| | F-measure | Entropy | F-measure | Entropy |
| Original | 1 | 1 | 1 | 1 |
| Jcn | **4** | **4** | 3 | **6** |
| Lch | **6** | **5** | **5** | **4** |
| Path | **6** | **4** | **4** | **4** |
| Wup | 3 | 2 | **5** | **5** |
| Res | 2 | 2 | **4** | 3 |
| Lin | **5** | **5** | **5** | **5** |
| Lesk | **6** | **4** | **4** | **5** |
| Random | 1 | 3 | 1 | 1 |

Another comparison was done between two sense representation methods, 'senseno' representation and 'offset' representation. In Table 6, 'senseno' representation had a performance similar to the performance of the 'offset' representation. Table 7 lists which sense representation method had the best F-measure and entropy in each data set. In this table, the 'senseno' representation method was the best performer in most data sets. We think the characteristic of our experimental data is the major reason for this result. All the documents used in our experiments are abstracts from technical journal papers. Most words are technical terms which are the second or the third sense of a word. Thus finding the correct sense of a word is most important. The 'senseno' representation method was helpful for distinguishing the same word with different senses. The 'offset' representation method was helpful for polysemous problem. Since some fixed terms are used in technical reports, the polysemous problem is not the major problem in the performance of our clustering system.

**Table 7: Best performer in each data set**

| | DS 1 | DS 2 | DS 3 | DS 4 |
|---|---|---|---|---|
| F-measure | senseno | senseno | word | offset |
| Entropy | senseno | senseno | senseno | word |
| F-measure | DS 5 | DS 6 | DS 7 | DS 8 |
| Entropy | senseno | offset | offset | senseno |
| F-measure | senseno | senseno | senseno | senseno |

Next we investigated whether the differences between using the sense of words and using the original words were significantly different. The Wilcoxon signed rank test were applied to our experimental results listed in Table 5. The results of each different semantic relatedness measures and sense representation methods were compared with the results of using the original words form and the W value were listed in Table 8. Since the sample size n of our dataset is 8, according to the standard of the Wilcoxon signed rank test, we know that the value of W must be equal to or smaller than 6 in order to reject the null hypothesis and prove the performance of two methods are statistically different. But instead, we got high W values ranging from 6 to 17. This indicates that although there are differences between the performance of using senses and original words in our experiments, we cannot draw the conclusion that using the senses is significantly better than using the original words. We also applied the Wilcoxon signed rank test to evaluate whether the performance differences between using different sense representation methods are statistically significant. For each semantic relatedness measures, we compared the results of using 'offset' and 'senseno' and the W values are listed in Table 9. For both F-measure and entropy, we got high W values. This indicates that in order to represent the Wordnet's sense of a word, there is no statistically significant difference between using the offset of a synset or just using the sense number in the sense list.

**Table 8: W values of Wilcoxon signed rank test for using word sense**

| Sense Methods | W value for F-Measure | W value for entropy |
|---|---|---|
| Jcn-offset | 17.5 | 16.5 |
| Jcn-senseno | 10 | 7 |
| Lch-offset | 6.5 | 10.5 |
| Lch- senseno | 8.5 | 10 |
| Path-offset | 7.5 | 13 |
| Path- senseno | 13.5 | 15.5 |
| Wup-offset | 16.5 | 12 |
| Wup- senseno | 12 | 11 |
| Res-offset | 9.5 | 9.5 |
| Res- senseno | 13 | 12 |
| Lin-offset | 16.5 | 12 |
| Lin- senseno | 11.5 | 9.5 |
| Lesk-offset | 10 | 13.5 |
| Lesk- senseno | 16 | 14 |
| Random-offset | 6 | 7 |
| Random- senseno | 6.5 | 6 |

**Table 9: W values of Wilcoxon signed rank test for sense representation methods**

| Sense Methods | W value for F-Measure | W value for entropy |
|---|---|---|
| Jcn | 7 | 10.5 |
| Lch | 13 | 13 |
| Path | 16 | 14.5 |
| Wup | 10 | 6 |
| Res | 16.5 | 14.5 |
| Lin | 10.5 | 15 |
| Lesk | 16 | 17 |
| random | 13 | 13 |

## 6. Conclusions and Future Work

In this paper, we investigate using the sense of a word to replace the original word form to solve the synonym problem and the polysemous problem in text data mining. The sense disambiguation method used in our experiments is based on the semantic relatedness among the senses. Eight different semantic relatedness measures implemented in WordNet:Similarity packages are used in our experiments. Some basic document clustering algorithms are evaluated for the algorithm selection. After describing the document preprocessing with some related natural language processing tools, the results of our experiments are listed. From these results, we can conclude:

For document clustering algorithm, the HAC method outperforms all the other methods for our small data sets. But for large data sets, the bisecting K-means method has the best performance. The differences between these methods are statistically significant.

Using the senses of words can improve the clustering performance for most of our data sets. But the statistical analysis performed on our results indicated that this improvement is not significant.

Two different sense representation methods, the 'senseno' method and 'offset' method, have similar performance, although the F-measure and entropy of the 'senseno' method are slightly higher than that of 'offset' representation. The results of the statistical testing method confirmed this conclusion.

Our future work focuses on the following two aspects:

*Performing syntactic analysis to find the important word in a context.* Currently all the words or compound words in a sentence are considered to be independent and of the same importance. Actually, words with different part-of-speech (POS) and syntactic attributes should be assigned different weights according to their relatedness to the content of the documents. One assumption may be that nouns, subjects, and objects are more important than other parts of speech for determining the topic of a document. There are a lot of syntactic analysis tools that can be used for mining more information from a raw text.

*Combining the semantic analysis and syntactic analysis to realize a further improvement.* Syntactic information is helpful for identifying the important words or key words from the raw text. By combining this with semantic information in the form of the senses of these key words, we expect to see an improvement in the performance of the document clustering system.

## 7. References

[1] S. Banerjee and T. Pedersen, "An Adapted Lesk Algorithm for Word Sense Disambiguation Using WordNet," In *Proc. of the Fourth International Conference on Computational Linguistics and Intelligent Text Processing (CICLING-02)*, Mexico City, 2002.

[2] F. Beil, M. Ester, and X. Xu, "Frequent Term-Based Text Clustering," In *Proc. of the 8th International Conference on Knowledge Discovery and Data Mining* (KDD 2002), Edmonton, Alberta, Canada, 2002.

[3] P. Bellot and M. El-Beze, *A Clustering Method for Information Retrieval,* Technical Report IR-0199, Laboratoire d'Informatique d'Avignon, France, 1999.

[4] P. Berkhin, "Survey of Clustering Data Mining Techniques," Accrue Software, http://citeseer.nj.nec.com/berkhin02survey.html (Access on 12 Feb. 2004).

[5] A. Budanitsky, *Lexical Semantic Relatedness and Its Application in Natural Language Processing*, technical report CSRG-390, Computer Systems Research Group, University of Toronto, 1999.

[6] A. Budanitsky and G. Hirst, "Semantic Distance in WordNet: An Experimental, Application-Oriented Evaluation of Five Measures," In *Workshop on WordNet and Other Lexical Resources, Second meeting of the North American Chapter of the Association for Computational Linguistics*, Pittsburgh, PA, 2001.

[7] D. Cutting, D. Karger, J. Pedersen, and J. Tukey, "Scatter/Gather: a Clusterbased Approach to Browsing Large Document Collection," In *Proc. of the 15th ACM SIGIR Conference,* Copenhagen, Denmark, 1992, pp. 318-329.

[8] D. Lin, "An Information-Theoretic Definition of Similarity," In *Proc. of the 15th International Conference on Machine Learning*, Madison, WI, 1998.

[9] C. Fellbaum, *WordNet: An Electronic Lexical Database*, MIT Press, 1998.

[10] K. Fragos, Y. Maistros, and C. Skourlas, "Word Sense Disambiguation using WordNet Relations," In *Proc. of the 1st Balkan Conference in Informatics*, Thessaloniki, Greece, 2003.

[11] B. C. M. Fung, *Hierarchical Document Clustering Using Frequent Itemsets,* Master Thesis, Dept. Computer Science, Simon Fraser University, Canada, 2002.

[12] P. Gomes, F. C. Pereira, P. Paiva, N. Seco, P. Carreiro, J. Ferreira, and C. Bento, "Noun Sense Disambiguation with WordNet for Software Design Retrieval," In *Proc. of the Sixteenth Canadian Conference on Artificial Intelligence (AI'03)*, Halifax, Canada, June-2003.

[13] K. M. Hammouda, *Web Mining: Identifying Document Structure for Web Document Clustering,* Master's Thesis, Department of Systems Design Engineering, University of Waterloo, Waterloo, Ontario, Canada, 2002.

[14] J. Han and M. Kamber, *Data Mining: Concepts and Techniques,* Morgan Kaufmann Publishers, 2001.

[15] G. Hirst and D. St-Onge, "Lexical Chains as Representations of Context for the Detection and Correction of Malapropisms," 1998, In C. Fellbaum, editor, WordNet: An Electronic Lexical Database, pp. 305-332, MIT Press.

[16] I. Iliopoulos, A.J. Enright, and C.A. Ouzounis, "Textquest: Document Clustering of Medline Abstracts for Concept Discovery in Molecular Biology," In *Proc. of the Sixth Annual Pacific Symposium on Biocomputing* (PSB 001), 2001.

[17] J. Jiang and D. Conrath, "Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy," In *Proc. of International Conference on Research in Computational Linguistics*, Taiwan, 1997.

[18] C. Leacock and M. Chodorow, "Combining Local Context and WordNet Similarity for Word Sense Identification," 1998, In C. Fellbaum, editor, WordNet: An Electronic Lexical Database, pp. 265-283, MIT Press.

[19] M. Lesk, "Automatic Sense Disambiguation: How to Tell a Pine Cone from an Ice Cream Cone," In *Proc. of the 1986 SIGDOC Conference,* New York, 1986, pp. 24-26. Association of Computing Machinery.

[20] X.B. Li, S. Szpakowicz, and S. Matwin, "A Wordnet-Based Algorithm for Word Sense Disambiguation," In *Proc. of IJCAI-95*, Montral, Canada, 1995.

[21] A. Likas, N. Vlassis, and J.J. Verbeek, "The Global K-Means Clustering Algorithm," *Pattern Recognition,* vol. 36, no. 2, 2003, pp. 451-461.

[22] K. Lin and R. Kondadadi, "A Word-Based Soft Clustering Algorithm for Documents," In *Proc. of 16th International Conference on Computers and Their Applications,* Mar. 2001.

[23] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. J. Miller, "Introduction to WordNet: An On-Line Lexical Database," *International Journal of Lexicography,* vol. 3, no. 4, 1990, pp. 235-312.

[24] S. Patwardhan, *Incorporating Dictionary and Corpus Information into a Vector Measure of Semantic Relatedness*, Master Thesis, University of Minnesota, Duluth, 2003.

[25] S. Patwardhan, S. Banerjee and T. Pedersen, "Using Semantic Relatedness for Word Sense Disambiguation," In *Proc. of the Fourth International Conference on Intelligent Text Processing and Computational Linguistics*, Mexico City, 2002.

[26] T. Pedersen, S. Patwardhan, and J. Michelizzi, "WordNet::Similarity - Measuring the Relatedness of Concepts," In Proc. of the

Nineteenth National Conference on Artificial Intelligence, San Jose, 2004.

[27] G. Ramakrishnanan and P. Bhattacharyya, "Text Representation with WordNet Synsets using Soft Sense Disambiguation," In *Proc. of 8th International Conference on Applications of Natural Language to Information Systems (NLDB 2003)* , Burg (Spreewald), Germany, 2003, pp. 214-227.

[28] P. Resnik, "Using Information Content to Evaluate Semantic Similarity," In *Proc. of the 14th International Joint Conference on Artificial Intelligence*, Montreal, 1995, pp. 448-453.

[29] P. Resnik, "Semantic Similarity in a Taxonomy: An Information-Based Measure and its Applications to Problems of Ambiguity in Natural Language," *Journal of Artificial Intelligence Research*, vol. 11, 1999, pp. 95-130.

[30] J. C. Reynar and A. Ratnaparkhi, "A Maximum Entropy Approach to Identifying Sentence Boundaries," *Proceedings of the Fifth Conference on Applied Natural Language Processing,* Washington, D.C., March 31-April 3, 1997.

[31] G. Salton, *The SMART Retrieval System – Experiments in Automatic Document Retrieval,* New Jersy, Englewood Cliffs: Prentice Hall Inc., 1971.

[32] G. Salton and C. Buckley, "Term-Weighting Approach in Automatic Text Retrieval," *Information Processing & management,* vol. 24, no. 5, 1988, pp. 513-523.

[33] G. Salton and M. J. McGill, *Introduction to Modern Information Retrieval,* McGraw-Hill, New York, 1983.

[34] M. Steinbach, G. Karypis, and V. Kumar, "A Comparison of Document Clustering Techniques," *KDD Workshop on Text Mining,* 2000.

[35] M. Sussna, "Word Sense Disambiguation for Free-text Indexing using a Massive Semantic Network," *In Proc. of the second international conference on Information and knowledge management*, Washington, D.C., United States, 1993, pp. 67 - 74.

[36] S. Weiss, H. White, and C. Apt'e, "Lightweight Document Clustering," In *Proc. of PKDD-2000,* Springer, 2000, pp. 665-672.

[37] E. Whitley and J. Ball, "Statistics review 6: Nonparametric methods," September 2002. Available online http://ccforum.com/content/6/6/509.

[38] Z. Wu and M. Palmer, "Verb Semantics and Lexical Selection," In *Proc. of the 32nd Annual Meeting of the Association for Computational Linguistics*, Las Cruces, New Mexico, 1994.

[39] O. Zamir, *Clustering Web Documents: A Phrase-Based Method for Group Search Engine Results,* Ph.D. dissertation, Dept. Computer Science & Engineering, Univ. of Washington, 1999.