Using Topic Keyword Clusters for Automatic Document Clustering

Hsi-Cheng Chang
Department of Electronic Engineering, Hwa Hsia Institute of Technology,
No.111, Gong Thuan Rd., Chung Ho City, Taipei 235, Taiwan, R.O.C. E-mail: hcchang@cc.hwh.edu.tw

Chiun-Chieh Hsu
Department of Information Management, National Taiwan University of Science and Technology,
No.43, Sec.4, Keelung Rd., Taipei 106,Taiwan, R.O.C.
E-Mail: cchsu@cs.ntust.edu.tw

Abstract

Data clustering is a technique for grouping similar data items together for convenient understanding. Conventional data clustering methods, including agglomerative hierarchical clustering and partitional clustering algorithms frequently perform unsatisfactorily for large text article collections, as well as the computation complexity of the conventional data clustering methods increase very quick with the number of data items. This paper presents a system for automatic document clustering by identifying topic keyword clusters of the text corpus. The proposed system adopts a multi-stage process. First, an aggressive data cleaning approach is employed to reduce the noise in the free text and further identify the topic keywords within the documents. All extracted keywords are then grouped into topic keyword clusters using the k-nearest neighbor graph approach and the keyword clustering function. Finally, all documents in the corpus are clustered based on the topic keyword clusters. The proposed method was assessed against conventional data clustering methods on a web news collection, indicating that the proposed method is an efficient and effective clustering approach.

1. Introduction

In information processing, document clustering and categorization have been extensively applied to information retrieval systems for enhancing performance and other intelligent applications [1,2]. Many clustering methods have been presented for browsing documents or organizing the retrieval results for easy viewing [3,4,5]. Some conventional clustering methods, such as agglomerative clustering methods start with all the documents as a separate cluster. Each step of the method involves merging the two most similar clusters. After each merge, the total number of clusters decreases by one. This step can be repeated until the desired number of clusters is obtained or the distance between the two closest clusters is above a certain threshold. Another clustering method, centroid-based approaches, *k*-means tries to assign documents to clusters to minimize the mean square distance of documents to the centroid of the assigned cluster.

These clustering methods have major limitations [6,7]. The agglomerative clustering methods such as the average link algorithm bases merging decisions on static modeling of the clusters to be merged; the single link algorithm considers only the minimum distance between the representative data of two clusters, and does not consider the aggregate interconnectivity between the two clusters. Restated these algorithms do not consider special properties of individual clusters and, thus may make wrong merging decisions when the underlying data do not follow the assumed model, or when noise is present. Centroid-based clustering methods, such as *k*-mean clustering algorithm are appropriate only for data in metric spaces (e.g., Euclidean space) in which a centroid of a given set of data can be computed. A major drawback of the *k*-mean schemes is that the *k* is difficult to define in the beginning and it fails for data in which items in a given cluster are closer to the center of another cluster than to the center of their own cluster. This phenomenon can occur in many natural clusters [6].

Moreover, the computational cost of the agglomerative document clustering algorithm is usually $O(wN^3)$ [8], where N denotes the number of documents in the corpus and w denotes the average number of features in the documents. The computation complexity grows exponentially with the length of the articles and the number of documents in the data collection. For a large document collection not only is the complexity very high but the memory requirement is also very large, maybe such that the data cannot fit in main memory.

Reducing the computation complexity of keyword comparison in documents clustering and increasing the clustering accuracy is the main objective of this study. This paper reverses the process of conventional data clustering methods. Initially, the system discovers a set of tightly relevant topic keyword clusters that are well scattered throughout the features space of the document collection. All documents in the document collection are then clustered according to these topic keyword clusters.

2. Data Pre-processing

Figure 1 illustrates an overview of the clustering system. Initially, natural language techniques segment sentences into meaningful multi-character words as well as identify phrases and name entities within articles. The topic keywords are extracted and mapped into a indirection weighted diagraph where an vertex denotes a keyword and an edge denotes an association between two keywords. The topic keywords with high component weights were selected as *candidate* topic keywords. Then, the *k*-nearest neighbor approach is employed to find germ-groups of the candidate topic keywords and further to form the topic keyword clusters of all the topic keywords. The generated topic keyword clusters are employed to find documents related to the topics.

Proceedings of the Third International Conference on Information Technology and Applications (ICITA'05)
0-7695-2316-1/05 $20.00 © 2005 IEEE

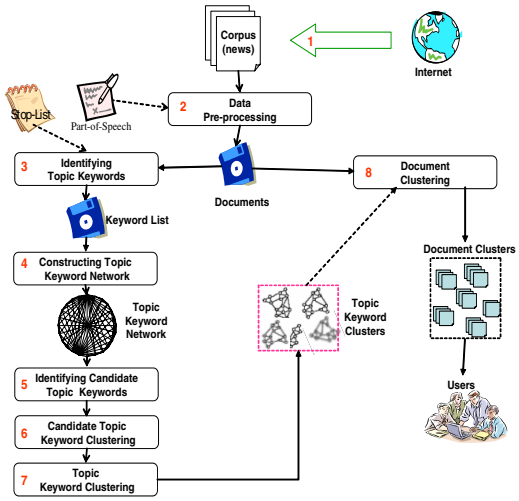

Figure 1 The document clustering system.

2.1 Word Segmentation

The articles in the test corpus were bi-lingual in English and Chinese. A *Term Parser* unit was used to partition the paragraphs into sentences and identify terms from sentences. Identified English terms were not stemmed, while a sentence in Chinese, which is a character-based language, must be segmented into meaningful single- and multi-character terms. Here the CKIP (Chinese Knowledge Information Processing) Chinese word segmentation program was used to process Chinese text and determine the part-of-speech of each term in the corpus.

2.2 Phrase Identification

Data must be identified which are rich in meaning, and which profit for topic representation and improve the possibility of obtaining human-understandable results [1,2]. Closely examining various corpora of data also reveals that documents related to the same topic/event usually share several name entities and word-pairs. For example, "wireless", "local" and "network" are ordinary terms in documents that discuss network application, and "local network", "wireless network" and "wireless local network" are the technical terms that appear most often. The phrases are more meaningful and more appropriate for representing the documents than ordinary terms. Based on these observations, a phrase identification program, a statistical technique and a modified DHP algorithm [9] that suitable for use in English and Chinese were developed and applied to identifying meaningful phrases in the document collection. Experiments were performed on a web news collection with phrases and non-phrases identification, and a precision improvement of 10% was obtained for clustering with phrases identification (see Section 4).

2.3 Feature Selection

Clustering is the unsupervised classification of data items into groups. Document clustering is a problem of high dimension data clustering. Bad feature selection has been shown to significantly negatively impact document clustering [10]. To maintain the computing cost of the clustering algorithm small and increase the clustering accuracy, removing the words that are not meaningful and discriminative among topics is very important.

Table 1 Parameter of a web news corpus.

Number of Documents (topics/article)	5200(26/200)
Total Number of words	950082
Average length of Documents	183
After remove single character terms (diff. words)	613772(21889)
After aggressive word removed (diff. words)	99045(6620)
Percentage of word removed	89.5751%

Four feature selection metrics were used to remove the meaningless terms from the documents.

1. All single-character terms were deleted since they are virtually always used as prepositions.
2. All function terms, that is modifiers such as adjectives and pronouns, were removed based on the part-of-speech information.
3. A *Zipfs* law-based [11] eliminator was used to remove terms that appear less than three times and that occur in over 5% of the documents in the corpus, as these are used in too many topics to effectively discriminate between topics.
4. The terms which term frequency in a document less than 2 are also removed. All such terms are useless for document clustering and cannot be employed to express the subject of the documents.

Table 2 Clustering accuracy: passive vs. aggressive feature selection.

Clustering Algorithm	passive	aggressive
Single-link	1.65385%	5.55556%
Complete-link	29.5385%	55.0393%
Average-link	22.3278%	43.6824%
K-Mean	10.7692%	40.9538%
Bisecting *K-Mean*	25.9423%	63.2006%
Proposed method	21.2018%	74.3486%

In practice, strict criteria were used to process feature-eliminating. Table 1 summarizes the experimental results, the word elimination rate approximate 90%.

The following question arises: Are such aggressive feature eliminating metrics a good form of feature selection? This was tested on the test corpus using conventional data clustering methods and the proposed clustering method as classifiers. Table 2 presents the test results. A precision of 21% was obtained for passive feature selection, compared with 74% for aggressive feature selection. Aggressive feature selection not only produced better clustering accuracy than passive feature selection but also significantly decreased the computation complexity.

2.4 Document Representation

After the text was preprocessed, each document was

represented using a set of features that includes salient phrases and all of the remained unique words. All of the documents are represented using a vector-space model [3,12]. In this model, each document is denoted as $d=\{w_1, w_2, \ldots, w_n\}$. The weights w_i of the terms are estimated as $tf_i \times idf_i$. Here *term-frequency* tf_i denotes the frequency of the i th term in the document, and idf_i is the *inverse document frequency* in the collection of documents. Finally, the weight of each document vector is normalized to unit length, such that $\left|\sum_{i=1}^{n}(tf_i \times idf_i)^2\right|^{1/2}$ to account for the fact that documents have different lengths.

In the vector-space model, the similarity between two documents d_x and d_y is commonly measured using the cosine function [3], Since the document vectors are of unit length, the above formula can be simplified to $cos(d_x, d_y) = d_x \cdot d_y$. The agglomerative clustering methods and *k*-mean clustering algorithm are based on this similarity measurement method.

3. Topic Keyword Clustering and Document Clustering

3.1 Identifying Topic Keyword

To identify the topic keywords of a corpus, there are two questions involved:

1. How many keywords reserved for a document can obtain the best clustering results; and
2. What is the best number of keywords for representing a topic?

Several studies [13,14,15] have experimented and given suggestions. Our earlier study [14] selected 5-40 keywords from each document in a collection and employed the clustering algorithm based on keyword clusters to cluster the documents in the collection. The experimental results, indicating that using 10-25 keywords to repent a document yields optimum clustering results.

Koller et al. [13] employed 10 and 20 keywords as representation of topics for hierarchical document clustering with a better classification results than using large number of keywords. Our previous study [15] collected a news corpus containing 20 topics and each topic with 100 documents, then applied a modified projecting clustering algorithm [16] for identifying the representation keywords of the original document clusters. Experimental results indicate that leaving more features will lower the coherence that is more features will incur more noise thus reducing the coherence. Above studies all indicate that words which do not discriminate for clustering documents, even if meaningful, must be removed.

To obtain the best meaningful and discriminating keywords for keyword clustering, based on these studies and experiments, a human-generated stop-words list was applied to remove the nouns and verbs that have general meaning but do not discriminate topics as inappropriate for representing topics. After this step, all remaining keywords are viewed as the topic keywords of the corpus and are used to construct a topic keyword associate network.

3.2 Estimating Semantic Correlation among Topic Keywords

In the proposed clustering algorithm, a keyword cluster refers to a group of keywords that occur together in multiple articles and can form a topic in the dataset. Co-occurrence of words has been shown to carry useful information [10,17]. This information shows correlated items rather than a topic. However, according to our results, correlated keywords frequently occurred within a recognizable topic – clustering the interesting correlations made topic identification possible.

A co-occurrence based correlation is generally based on the assumption that highly co-occurring words are likely to be used together to describe a certain topic. They can be reasonably grouped as a large semantic node as a topic in a collection of documents. The co-occurrence based correlation of two words t_i and t_j for topic T is computed as $r_{ij} = f(t_i \cap t_j) / \mathrm{MAX}(f(t_i), f(t_j))$, where $f(t_i \cap t_j)$ denotes the times at which terms t_i and t_j co-occur in the documents. Herein, the co-occurrence correlation was applied to estimate the strength of association among all topic keywords, to construct a topic keyword network.

3.3 Constructing Topic Keyword Network

When the topic keyword network had been constructed, the weighted graph was examined; many edges with small weights were found; that is, the association between the keywords was weak. Clusters of various document topics have various keywords. These weak relationships can be eliminated. The *mean association weight* of the edges in the graph is computed as a threshold to prune away the weak edges in the weighted graph, according to the formula $AW = \sum_{r_{ij} \in E(G)} r_{ij} \Big/ n$, where n denotes the total number of edges in the weighted graph and r_{ij} denotes the weight of the edges in the graph. The edges with weights below the mean association weight are removed from the graph, and the topic keyword network is modified as a sparse network. The keyword-clustering algorithm was applied on this graph to determine the keyword clusters in the collection of documents.

3.4 Topic Keyword Clustering

The keyword clustering algorithm has five main steps. The document clustering is then continued based on the identified keyword clusters.

3.4.1 Identifying Candidate Topic Keywords

The first step of the keyword clustering algorithm attempts to obtain the candidate vertices from the topic keyword network. The vertices become the central points of the topic keyword clusters. First, the *composite weight* of each vertex in the network, which is the sum of the weight of the keyword and the average weight of the edges incident

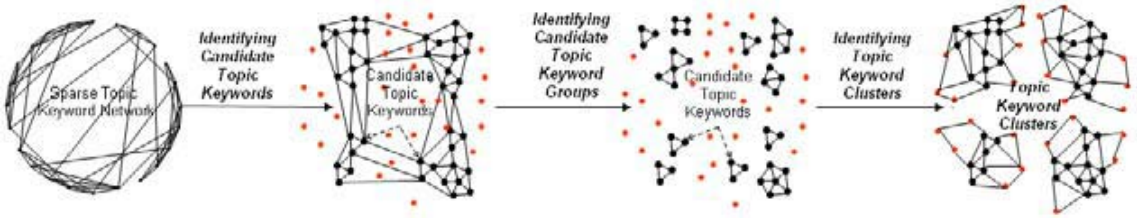

Figure 2 *K* nearest neighbor algorithm for candidate topic keyword clustering.

on the vertex is computed as $CW_i = w_i + \sum_{j=1}^{m} r_{ij} / m$,where w_i denotes the weight of vertex v_i ($tf_i \times idf_i$ of term v_i) in the graph and m denotes the *degree* of vertex v_i. The value r_{ij} denotes the weight of edge (v_i, v_j) directly connected to the vertex v_i in the graph. In the following, the *average composite weight* of the graph is calculated using $ACW = \sum_{i=1}^{k} CW_i / k$, where k denotes the number of vertices in the graph. The vertices whose composite weight exceeds the average composite weight are selected as *candidate topic keywords*. And then a candidate topic keyword sub-network of the topic keyword network was extracted as Figure 2 shown.

In practice, to obtain useful candidate topic keyword groups when the k nearest neighbor algorithm is applied, this threshold must be dynamically adjusted with the data collection to reserve an appropriate number of candidate keywords for obtaining the best topic keyword clusters. As section 2-2 describes, the optimal number of representation keywords of the original document clusters is 10-20 keywords.

3.4.2 Finding Candidate Keyword Groups

This paper uses the k nearest neighbor approach to identify the candidate keyword groups as the kernel of the keyword clusters. Figure 2 depicts the clustering process. The value of k determines the size of the candidate keyword group. If the nearest neighbor clustering process continues until all vertices are labeled or no additional labeling occurs, all of the vertices may cluster into one or few groups. Here, the value of k for identifying the candidate keyword groups can not be very large. In this experiment k is 2.

3.4.3 Finding the Connected Component of Each Candidate Keyword Groups

The candidate keyword groups identified in preceding step were applied as the central groups to find all of the keyword clusters of the topic keyword network. All the vertices directly connected to each candidate keyword group were found and formed a *connected component*, i.e. a *topic keyword sub-cluster* of the topic keyword network. The weight of each connected component was computed using $W_{G_k} = \sum_{r_{ij} \in G_k} r_{ij}$, that is the sum of the weights of all edges in the connected component G_k , where r_{ij} denotes the weight of an edge in the sub-cluster.

3.4.4 Keyword Clusters Merging

As soon as the topic keyword sub-clusters have been obtained, the keyword clustering algorithm switches to a greedy merge algorithm that searches and combines the sub-clusters that are strongly inter-connected. The similarity between each pair of sub-clusters G_i and G_j is determined by considering their *relative inter-connective* $RI(G_i, G_j)$. The relative inter-connectivity between a pair of sub-clusters G_i and G_j is defined as the absolute inter-connectivity between G_i and G_j normalized to their internal inter-connectivity. The absolute inter-connectivity of a pair of clusters G_i and G_j is defined as the sum of the weights of the edges that connect the vertices in G_i to the vertices in G_j, and is denoted as $E_{\{Gi,\ Gj\}}$. The internal inter-connectivity of a cluster defined as the weighted sum of the edges in the cluster. Therefore, the relative inter-connectivity of a pair of clusters G_i and G_j is $RI(G_i, G_j) = |W_{E_{\{G_i, G_j\}}}| / |W_{G_i}| + |W_{G_j}|$. The system merges the pair of keyword sub-clusters whose $RI(G_i, G_j)$ exceeds a threshold.

3.4.5 Refining keyword clusters

The number of keywords in each keyword cluster is unequal. In the document clustering phase, a keyword cluster with more keywords clusters more documents. Therefore, not only do the meanings of the keywords in the keyword clusters affect the results of document clustering, but the sizes of the keyword clusters also affect the accuracy. Hence, a criterion is required for refining the keyword clusters.

The *component weight* of the keyword clusters is applied to eliminate loosely intra-connected keywords from the keyword clusters. The *component weight* is given by $CW = CD(G) \times AS(G)$, where $CD(G)$ denotes the *connected density* and $AS(G)$ represents the *average strength* of a keyword cluster. The *connected density* is computed as $CD(G) = \frac{|E(G)|}{|V(G)| \times |V(G)-1| / 2}$, where $|E(G)|$ denotes the number of edges in the keyword cluster, and $|V(G)|$ denotes the number of vertices in the keyword cluster. The *average strength* of each cluster is estimated to be $AS(G) = \sum_{r_{ij} \in E(G)} r_{ij} / |E(G)|$, where $\sum_{r_{ij} \in G} r_{ij}$ is the sum of the weights of all edges in the cluster. The component weight was applied to refine keyword clusters whose keywords numbered more than the average number of keywords in all clusters. The loosely related keywords in the keyword clusters were thus removed.

3.5 Clustering Documents Based on Keyword

Clusters

The preceding process produced reasonable keyword clusters. Next, the documents in the document collection were clustered according to the measured similarity among the keywords in the document and each of the keyword clusters. The *cosine* similarity measurement was used to measure the distance between a topic keyword cluster and a document, and then choose the closest topic for each document.

In practice, the topic-to-document clustering was found to have two problems. First, that some documents were close to multiple topics. In some cases, this overlap was common and repeated; some documents referenced both topics. Cutoffs can be applied to tackle situations when a document is not close to any topic and allow multiple mappings if it is close to many for practical application. In this experiment, a document can be responsible for multiple topic keyword clusters, but for evaluating against the clustering results of the convention clustering methods a single topic must be identified, which would be that with the best similarity for each document. The other problem is that some documents can not map to any topics, i.e. the documents are far from each of the topic keyword clusters or the similarity is zero. This condition occurs when the words in the document are not prominent and most were eliminated in the feature selection stage, or when the keywords of the document do not form the keyword clusters. In the experiment all of these documents are regarded as the outliers of the corpus.

3.6 Performance Analysis

The computation cost of the agglomerative clustering algorithm is usually $O(wN^3)$ [8], where N is the number of documents in the corpus and w is the average number of keywords in the documents.

The overall computation complexity of the proposed document clustering algorithm depends on the amount of time it requires to perform the keyword clustering algorithm and the amount of time required to cluster the documents with the keyword clusters. In keyword clustering algorithm, the amount of time required to construct the topic keyword network is $O(m^2)$, where m denotes the number of topic keywords identified in the corpus. The amount of time required to compute the k-nearest neighbor graph depends on the dimensionality of the underlying candidate topic keywords. Algorithms based on k_d trees [19] can be used to quickly compute the k nearest neighbors. For the number of *candidate* keywords n, the overall complexity is $O(n \log n)$ [4].

The amount of time required to compute the relative inter-connectivity for each initial cluster of the keyword cluster merging step is proportional to the number of keywords r in each found cluster. The worst case complexity is obtained when the merging function repeatedly chooses the same cluster and merges it with another; *i.e.*, it grows a single large cluster. In this case, the amount of time is $O(rk(rk-1))$. The value of k is the number of the clusters initial found. Additionally, the amount of time needed to cluster the documents based on the topic keyword clusters is $O(CN)$. The value of C is the number of the keyword clusters. Thus, leading to an overall complexity of the proposed clustering algorithm is $O(CN + rk(rk-1) + n\log n + m^2)$. Since the number of the keyword clusters k and the number of keywords r in each cluster are small than the number of documents N in a large corpus, the computation complexity can be simplified as $O(CN + n\log n + m^2)$.

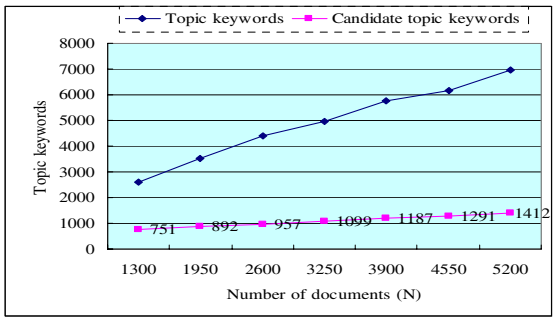

Figure 3 Relation of documents vs. keywords.

The computation complexity of the proposed document clustering system is essential for the document clustering algorithm, since the computation complexity of the keyword clustering algorithm grows with the number of topic keywords, not the number of the documents in the corpus. In general, the number of keywords increases with the number of documents is $N^{0.7}$ [12]. In practice, the corpus has duplicate (or near-duplicate) documents, including multiple news stories based on the same newswire article. These duplicate documents give a few topic keywords. Besides, authors generally repeat general words to describe the subjects in an article, rather than using many synonyms. The feature selection metrics can remove many useless words. Thus, the topic keywords will not increase quickly with the number of documents. Figure 9 shows that the number of topic keywords is only a little larger than N. In addition, the number of the candidate keywords n is smaller than that of the topic keywords, because the candidate topic keywords are selected from the topic keywords. In sum, the computation complexity of the proposed clustering method is in the order of CN, since the term CN dominates $n\log n$ as N becomes very large, where CN is smaller than the conventional data clustering algorithm.

4. Experiments and Evaluation

The testing data was accumulated from well-known web sites, including "*Yahoo!*" (http://tw.yahoo.com), "*Yam*" (http://www.yam.com/), and others. Some keywords, covering 26 topics, were entered as queries, and 200-220 documents about each topic were gathered. The test corpora were clustered using the proposed system and five well-known document-to-document similarity based clustering methods - *single-link*, *complete-link* as well as *average-link* agglomerative clustering methods and the *k-mean* as well as *bisecting k-mean* of partition clustering methods. Clusters were compared to the original manually

Table 3 Precision rate of corpora with different number of documents in each topic.

# doc.(topics/# doc. each)	1300(26/50)		2600(26/100)		3900(26/150)		5200(26/200)	
	1	2	1	2	1	2	1	2
Single Link	5.92308%	5.84615%	8.76923%	8.69231%	5.84615%	5.84615%	5.55556%	4.32692%
Complete Link	54.7692%	49.0769%	35.6154%	30.0769%	20.7949%	19.0256%	55.0393%	28.6923%
Average Link	63.2308%	50%	40.1538%	39.8846%	27.4872%	27.3846%	43.6824%	38.7414%
K-Mean	54.7692%	53.2308%	56.3077%	54.3846%	48.6154%	49.7179%	40.9538%	39.6346%
Bisecting K_Mean	58.0769%	51.9231%	64.5%	62.2308%	58.7692%	57.2308%	63.2006%	42.8077%
Proposed Method	72.3077%	69.4615%	77.2631%	71.5861%	74.4336%	70.1667%	70.3486%	66.0049%

grouped clusters of documents to estimate the precision of clustering. The clustering precision rate was computed as

$$precision\ rate = \frac{The\ number\ of\ documents\ found\ by\ a\ clustering\ method\ and\ belonging\ to\ the\ correct\ cluster}{The\ number\ of\ documents\ in\ the\ cluster}$$

Experiment-1: The experiment attempted to determine whether more articles in a topic cluster would increase the accuracy of the clustering. Four test corpora were constructed with 26 topics, each topic having 50, 100, 150 and 200 articles respectively. Table 3 summarizes the experimental results. More documents in a topic cluster do not increase consequentially the clustering accuracy in the news corpus, but the proposed clustering method outperforms the five well-known document-to-document similarity based clustering methods.

In practice, the number of the keyword clusters produced by the proposed clustering method did not exactly match the number of the clusters in the original corpora, for instance the system produced 31 clusters from the 5200(26/200) corpus. The performance of the agglomerative clustering method depends on the data in the collection. When more documents are included, indicating that the contents cover various topics, the accuracy of the clustering results decreases quickly. Accuracy of the proposed method, which is based on identifying topics, is always reliable in different test corpora.

5. Conclusion

Although numerous interesting document clustering methods have been extensively studied for many years, the high computation complexity and space need still make the document-to-document similarity based clustering methods inefficient. Hence, reducing the heavy computational load and increasing the precision of the unsupervised clustering of documents are important issues.

This paper presented a document clustering method, based on the topic of *keyword clustering*, for alleviating these problems satisfactorily. There are two major benefits of using keyword clustering to proceed document clustering. Firstly, only those discriminative and meaningful topic keywords were used, this greatly reduces the computation complexity and computing load does not increase with the number of the documents, so suitable for clustering large document collection, Secondly, the proposed clustering method obtains high precision rate compared with those produced by the other clustering methods.

References

[1] Y. S. Lai, and C. H. Wu, "Meaningful term extraction and discriminative term selection in text categorization via unknown-word methodology", *ACM Transactions on Asian language information processing*, vol. 1, no. 1, pp.34-64 March 2002.

[2] C. Clifton, R. Cooley, and J. Rennie, "TopCat: Data Mining for topic identification in a text corpus", *IEEE Transactions on knowledge and data engineering*, pp.2-17, 2003.

[3] G. Salton, "Automatic text processing: The transaction, analysis, and Rretrieval of information by computer", *Addison-Wesley*, 1989.

[4] Y. Yang, and C. G. Chute, "An application of least squares fit mapping to text information retrieval", *ACM SIGI*R, pp. 281-290, 1993.

[5] S. H. Lin, and M. C. Chen etc., "ACIRD: Intelligent internet document organization and retrieval", *IEEE transactions on Kknowledge and data engineering*, vol. 14, no. 3, pp.599-614, May/June 2002.

[6] G. Karypis, E. H. Han, and V. Kumar, "CHAMELEON: a hierarchical clustering algorithm using dynamic modeling", *IEEE Computer*, pp. 68-75, 1999.

[7] E.H. Han, G. Karypis, V. Kumar, and B. Mobasher, "Hypergraph based clustering in high-dimensional data sets: A summary of results", *Bulletin of the Technical Committee on Data Engineering*, vol. 21 no. 1, 1998.

[8] W. B. Frakes and B. Y. Ricardo, "Information retrieval data structures & algorithm", *Prentice Hall* press, 1992.

[9] J.S. Park, M. S. Chen, and P. S. Yu, "Using a hash-based method with transaction trimming for mining association rules", *IEEE transactions on knowledge and data engineering*, vol. 9, no. 5, pp. 813-825, 1997.

[10] Y. Yang, "Noise reduction in a statistical approach to text categorization", *ACM SIGIR*, pp. 256-263, 1995.

[11] H. P. Zipf, "Human behavior and the principle of least effort", *Addison-Wesley*, Cambridge, Massachusetts, 1994.

[12] B. Y. Ricardo, and R. N. Berthier, "Modern information retrieval", *Addison-Wesl,y press*, 1999.

[13] D. Koller, and M. Sahami, "Hierarchically classifying documents using very few words", *Proceedings of ICML 14th International Conference on Machine Learning*, 1997.

[14] H. C. Chang, C. C. Hsu, and Y. W. Deng, "Automatic document clustering based on keyword clusters using partitions of weighted undirected graph", *Proceeding of 2003 Symposium on Digital Life and Internet Technologies*, September 2003.

[15] S.M. Hsieh, S.J.Huang, C.C. Hsu, and H.C. Chang, "Personal documents recommendation system base on data mining techniques", *Proceeding of 2004 IEEE/WIC/ACM International Joint Conference on Web Intelligence*. September 2004.

[16] C.C. Aggarwal, C. Procopiuc, J.L. Wolf, P.S. Yu and J.S. Park, "A framework for finding projected cluster in high dimensional spaces", *ACM SIGMOD Conference on Management of Data*, 1999.

[17] C. Silverstein, S. Brin, and R. Motwani, "Beyond market baskets: Generalizing association rules to dependence rules", *Data mining and knowledge discovery*, vol. 2, no. 1, pp. 39–68, Jan. 1998.

[18] J. Liu, and T. S. Chua, "Building semantic perception net for topic spotting", *proceedings of ACL*, 2001.

[19] H. Samet. The design and analysis of spatial data structures. *Addison-Wesley*, 1990.