# Document Clustering with Hierarchical Algorithm

**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 is a widely used strategy for information retrieval and text data mining. Partitioning and hierarchical clustering methods are most widely used algorithms. Other investigators proposed to use bisecting K-means method for document clustering and their experimental results have indicated that the bisecting K-means method is the preferred method for document clustering [16]. However, in our research we have found that, whereas the bisecting K-means method has advantages when working with large datasets, a traditional hierarchical clustering algorithm still achieves the best performance for small datasets.

**Keywords**: Document Clustering, Information Retrieval.

## 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. Another application of document clustering is the automated or semi-automated creation of document taxonomies. A good taxonomy for Web documents is Yahoo (www.yahoo.com).

Traditional data clustering methods include partitioning algorithms and hierarchical clustering algorithms. 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 until there is no change for any of the clusters. The use of the K-means method for document clustering can be found in [2, 8]. Some variants of K-means methods include the K-medoids method [7] and global k-means method [9].

Hierarchical clustering generates a hierarchical tree of clusters. This tree is also called a dendrogram [3]. 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. This is a kind of bottom-up approach. 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. A divisive method is a kind of top-down method. Steinbach, Karypis, and Kumar compared the performance of three agglomerative clustering algorithms, IST (Intra-Cluster Similarity Technique), CST (Centroid Similarity Technique), and UPGMA [16]. Their experimental results show UPGMA is the best one among them. Maarek, Fagin, Ben-Shaul, and Pelleg proposed to HAC algorithm for on-line ephemeral web document clustering [12].

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 [4].

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 [16]. 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 particular algorithms for document clustering were proposed. Zamir has described the use of a suffix tree for document clustering [18]. 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 [1]. Fung proposed another hierarchical document clustering method based on the frequent term set, HIFC (Frequent Itemset-based Hierarchical Clustering), to improve the HFTC method [5]. Hammouda proposed an incremental clustering algorithm by representing each cluster with a similarity histogram [6]. Weiss, White, and Apte described a lightweight document clustering method using nearest neighbors [17]. Lin and Ravikumar described a soft document clustering system, which is called WBSC (Word-based Soft Clustering) [10].

In this paper, we report our comparison results of four different clustering algorithms: k-means, buckshot, HAC, and bisecting k-means method. Before presenting our experiments, the natural language processing tools used for data preprocessing in our system are provided in section 2. In section 3, we present our experiments results and analysis. Section 4 lists our final conclusions and promising future work.

## 2. Document Preprocessing

The tasks of data preprocessing for general text data mining problems include tokenization, morphological analysis, part-of-speech tagging, phrase identification, syntactic analysis, and semantic analysis. The fast development of NLP techniques provides good support for this step.

Tokenization is the very first step involved in most NLP processing tasks. A tokenizer separates a text into a set of component elements which is called tokens. The simplest tokenization method is splitting the text according to blanks and punctuation marks. A simple sed script implementation of tokenizer is provided by the Penn Treebank project group. MXTERMINATOR is a JAVA tokenizer implemented by Adwait Ratnaparkhi [15]. It is used to identify the sentence boundaries and separate the sentences in the text.

Morphology analysis converts the morphological variations of a word, such as inflections and derivations, to its base form. One of the traditional methods uses a stemmer. Stemmers try to identify the stem of a raw word in a text to reduce all such similar words to a common form, making the statistical data more useful. The process of stemming removes the commoner morphological and inflexional endings from words in English. For example, the phrases *analysis*, *analyzer*, and *analyzing* all have the stem form *analy*. The most widely used two stemmers are the Porter stemmer [14] and Lovins stemmer [11]. Another morphology analysis technique is lemmatization. A lemmatizer is a linguistic suffix stripper that is more accurate than a stemmer. Instead of identifying the stem of a word, a lemmatizer converts a word to its normalized form, called lemma. The implementation of a lemmatizer requires part-of-speech tagging, an extensive lexicon, and case normalization. For example, given the words *compute*, *computer*, *computing*, *computers*, and *computed*, a stemmer will convert all of them into *comput*. But for a lemmatizer, *compute*, *computing*, and *computed* have the same lemma *compute*, whereas *computer* and *computers* have the same lemma *computer*. A good lemmatizer is provided in WordNet [13].

## 3. Experiments

We collected 10,000 abstracts from journals belonging to ten different areas. For each area, 1000 abstracts were collected. This full data set was divided evenly into 5 subsets. Each subset contains 2000 abstracts and they are named as FDS 1 - 5. Another mini data set is selected from the full dataset. There are totally 1000 abstracts from 10 categories in this mini dataset. This mini dataset is partition into 5 groups evenly too. The size of each subset is 200. They are named as MDS 1 - 5.

All these abstracts were cut into the sentences with MXTERMINATOR. Then the tokens were identified from each sentence with the Penn Treebank tokenizer. The lemmatizer in WordNet was used to convert each token into lemma. All the stop words are filtered. Finally, a document is converted into a list of lemmas. These lemmas will be used to construct the feature vector for each document.

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 evaluation methods entropy and F-measure, which have been used by a number of researchers,

including Steinbach, Karypis, and Kumar [16], will be used. The detailed results are listed in table 1 and table 2.

The F-measure and entropy listed here are the average values of 20 different results. In five full data sets, we found that the bisecting method outperforms all the other methods. The K-Means method and buckshot method achieve similar results. The HAC method, only in the first dataset, gets a similar result to K-means and buckshot method. But for the other four datasets, the results of the HAC method are less than that of the K-means method and buckshot method by about 10-15 percentage points. In the five mini data sets, HAC achieves the best performance. The results of K-means, buckshot, and the bisecting K-means method are similar and low.

Table 1: Experimental Results – F-Measure

| | K-means | Buckshot | HAC | Bisecting |
|---|---|---|---|---|
| FDS1 | 0.77 | 0.73 | 0.74 | **0.9** |
| FDS2 | 0.72 | 0.73 | 0.55 | **0.85** |
| FDS3 | 0.79 | 0.75 | 0.6 | **0.87** |
| FDS4 | 0.72 | 0.74 | 0.59 | **0.86** |
| FDS5 | 0.77 | 0.73 | 0.74 | **0.9** |
| MDS1 | 0.41 | 0.51 | **0.77** | 0.38 |
| MDS2 | 0.48 | 0.51 | **0.77** | 0.4 |
| MDS3 | 0.42 | 0.47 | **0.78** | 0.34 |
| MDS4 | 0.39 | 0.48 | **0.63** | 0.36 |
| MDS5 | 0.41 | 0.51 | **0.77** | 0.38 |

Table 2: Experimental Results – Entropy

| | K-means | Buckshot | HAC | Bisecting |
|---|---|---|---|---|
| FDS1 | 0.6 | 0.67 | 0.73 | **0.4** |
| FDS2 | 0.73 | 0.7 | 1.19 | **0.5** |
| FDS3 | 0.6 | 0.65 | 0.98 | **0.46** |
| FDS4 | 0.74 | 0.72 | 1.09 | **0.51** |
| FDS5 | 0.6 | 0.67 | 0.73 | **0.4** |
| MDS1 | 1.5 | 1.28 | **0.58** | 1.6 |
| MDS2 | 1.35 | 1.24 | **0.51** | 1.55 |
| MDS3 | 1.5 | 1.38 | **0.52** | 1.71 |
| MDS4 | 1.61 | 1.37 | **0.84** | 1.63 |
| MDS5 | 1.5 | 1.28 | **0.58** | 1.6 |

Our results for the full data set are consistent with the results of Steinbach, Karypis, and Kumar. Steinbach, Karypis, and Kumar [16] concluded that the bisecting K-means technique is better than the standard K-means approach and as good as or better than the hierarchical approaches. A comprehensive analysis provided by Steinbach, Karypis, and Kumar explains that the nature of the document clustering problem is the reason for the worse performance of hierarchical approaches and the good performance of the bisecting K-means method. In all these clustering algorithms, two documents that share more common words will be considered as more similar to each other. The problem is that two documents consisting of the same set of words may be about two totally different topics. It is very possible that the nearest neighbors of a document may belong to different categories. In the HAC method, if two documents were assigned into the same group, they will always be in the same group. This assignment may be optimal in that step, but from the view of the whole partition, it may not be optimal. The HAC method just tries to get local optimality in each step with no attempt for global optimality. The advantage of the K-means method, buckshot method and bisecting K-means method is their adjusting of each cluster after each iteration. This reassignment is helpful for a global optimality. Compared with the K-means method and buckshot method, bisecting K-means can generate more evenly partitioned clusters. A balanced performance for each cluster is helpful for a higher global result.

Then why is the HAC clustering method the best one for the mini datasets? We think the major reason is the size of the data set. There are 2000 abstracts in each full data set and they are considered as 2000 clusters in the initial step of the HAC method. We know in each iteration of the HAC method, two nearest clusters are merged together to get local optimality. This optimality may be not helpful, and may even be harmful, for the next iteration. This loop will be repeated 1990 times to get the final 10 clusters; the advantages of each step will have counteracted with each other. Since there is no global reassignment procedure in the HAC method, the final partitions cannot be improved at any steps. In our mini data set, there are only 200 abstracts in each set. The iteration was repeated only 190 times. The advantage of the optimality in each step is still higher than that of the reassignment functions in the K-means, buckshot, and bisecting methods. When we checked the detailed debugging information for the K-means, buckshot, and bisecting methods, we found that for the mini data sets, K-means, buckshot, and the bisecting method had only 1 or 2 iterations. This means that, for those small datasets, the results of K-means, buckshot, and bisecting method are similar to that of a randomly partitioning. Notice that in the full data set, the performance of the buckshot method is similar to that of the K-means method. But in the mini data set, the performance of the buckshot method is always higher than that of K-means for all five mini datasets. This demonstrates that the use of HAC for seed selection is helpful for small data sets but not for the large data sets.

There are eight data sets were used for evaluation in Steinbach, Karypis, and Kumar's experiments [16]. The largest data set contains about 3000 documents and smallest one contains about 1000 documents. This size is similar to the size of our full data set. This also demonstrates that the good performance of bisecting K-means method achieved by in Steinbach, Karypis, and Kumar's experiments is also based on large data set.

# 4. Conclusions and Future Work

In this paper, we presented our experimental results for comparison of four different document clustering algorithms. We can conclude that for large data sets, the bisecting algorithm outperforms all the other methods. But for small data sets, the traditional hierarchical method gets the best performance. In the next step, we plan to improve the system from four aspects:

*Using compound words instead of single words.* Since the experimental data used in our experiments are abstracts of journal papers from different areas, and since a lot of science terms are compound words, we think that using compound words will be helpful for capturing the correct content of a document.

*Performing semantic analysis and using the sense of words or compound words instead of its original word form.* The synonym problem and polysemy problem are two major obstacles for text data mining. Most words used in technical papers are polysemous, and generally the correct senses of them are the second or the third senses. Finding the correct sense of words is important to understand the content of a document and distinguish one document from another.

*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 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 previous three approaches to realize a further improvement.*

# 5. References


[1] F. Beil, M. Ester, and X. Xu, "Frequent Term-Based Text Clustering," Proc. of the 8th International Conference on Knowledge Discovery and Data Mining, 2002.

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

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

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

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

[6] 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.

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

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

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

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

[11] J.B. Lovins, "Development of a Stemming Algorithm," Mechanical Translation and Computational Linguistics, vol. 11, 1968, pp. 22-31.

[12] Y.S. Maarek, R. Fagin, I.Z. Ben-Shaul, and D. Pelleg, Ephemeral Document Clustering for Web Applications, Technical Report RJ 10186, IBM Research, 2000.

[13] 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.

[14] M.F. Porter, "An Algorithm for Suffix Stripping," Program, vol. 14, no. 3, 1980, pp. 130-137.

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

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

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

[18] 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.