ISSN: 2277 – 9043
International Journal of Advanced Research in Computer Science and Electronics Engineering
Volume 1, Issue 5, November 2012

Clustering Web Documents using Hierarchical Method for Efficient Cluster Formation

I.Ceema[*1], **M.Kavitha**[*2], **G.Renukadevi**[*3], **G.sripriya**[*4], **S. RajeshKumar**[#5]

[*]**Assistant Professor, Bon Secourse College for women, Thanjavur, Tamilnadu, India.**

[#]**Assistant Professor, Bharath College of Science & Management, Thanjavur, Tamilnadu, India.**

***Abstract*— Document clustering aims to group, in an unsupervised way, a given document set into clusters such that documents within each cluster are more similar between each other than those in different clusters. It is an enabling technique for a wide range of information retrieval tasks such as efficient organization, browsing and summarization of large volumes of text documents. Cluster analysis aims to organize a collection of patterns into clusters based on similarity. Clustering has its root in many fields, such as mathematics, computer science, statistics, biology, and economics. In different application domains, a variety of clustering techniques have been developed, depending on the methods used to represent data, the measures of similarity between data objects, and the techniques for grouping data objects into clusters. The first part is a document index model, the Document Index Graph, which allows for incremental construction of the index of the document set with an emphasis on efficiency, rather than relying on single-term indexes only. It provides efficient phrase matching that is used to judge the similarity between documents. This model is flexible in that it could revert to a compact representation of the vector space model if we choose not to index phrases. The second part is an incremental document clustering algorithm based on maximizing the tightness of clusters by carefully watching the pair-wise document similarity distribution inside clusters. Both the phases are based upon two algorithmic models called Gaussian Mixture Model and Expectation Maximization. The combination of these two components creates an underlying model for robust and accurate document similarity calculation that leads to much improved results in Web document clustering over traditional methods.**

***Index Terms*—Document Clustering, Clustering, TF/IDF, Hierarchical algorithm, Cosine Similarity**

I. Introduction

Clustering is the classification of objects into different groups, or more precisely, the partitioning of a data set into subsets (clusters), so that the data in each subset (ideally) share some common trait - often proximity according to some defined distance measure. Data clustering is a common technique for statistical data analysis, which is used in many fields, including machine learning, data mining, pattern recognition, image analysis and bioinformatics. The computational task of classifying the data set into k clusters is often referred to as k-clustering.

Besides the term data clustering (or just clustering), there are a number of terms with similar meanings, including cluster analysis, automatic classification, numerical taxonomy, botryology and typological analysis.

Document clustering aims to group, in an unsupervised way, a given document set into clusters such that documents within each cluster are more similar between each other than those in different clusters. It is an enabling technique for a wide range of information retrieval tasks such as efficient organization, browsing and summarization of large volumes of text documents. Cluster analysis aims to organize a collection of patterns into clusters based on similarity. Clustering has its root in many fields, such as mathematics, computer science, statistics, biology, and economics. In different application domains, a variety of clustering techniques have been developed, depending on the methods used to represent data, the measures of similarity between data objects, and the techniques for grouping data objects into clusters.

Document clustering techniques mostly rely on single term analysis of the document data set, such as the Vector Space Model. To achieve more accurate document clustering, more informative features including phrases and their weights are particularly important in such scenarios. Document clustering is particularly useful in many applications such as automatic categorization of documents, grouping search engine results, building taxonomy of documents, and others. For this Hierarchical Clustering method provides a better improvement in achieving the result. Our project presents two key parts of successful Hierarchical document clustering. The first part is a document index model, the Document Index Graph, which allows for incremental construction of the index of the document set with an emphasis on efficiency, rather than relying on single-term indexes only. It provides efficient phrase matching that is used to judge the similarity between documents. This model is flexible in that it could revert to a compact representation of the vector space model if we choose not to index phrases. The second part is an incremental document clustering algorithm based on maximizing the tightness of clusters by carefully watching the pair-wise document similarity distribution inside clusters. Both the phases are based upon two algorithmic models called Gaussian Mixture Model and Expectation

Maximization. The combination of these two components creates an underlying model for robust and accurate document similarity calculation that leads to much improved results in Web document clustering over traditional methods.

II. Related Works

Document clustering has been studied intensively because of its wide applicability in areas such as web mining, search engines, information retrieval, and topological analysis. Unlike in document classification, in document clustering no labeled documents are provided. Although standard clustering techniques such as k-means can be applied to document clustering, they usually do not satisfy the special requirements for clustering documents: high dimensionality, high volume of data, ease for browsing, and meaningful cluster labels. In addition, many existing document clustering algorithms require the user to specify the number of clusters as an input parameter and are not robust enough to handle different types of document sets in a real-world environment. For example, in some document sets the cluster size varies from few to thousands of documents. This variation tremendously reduces the clustering accuracy for some of the state-of-the art algorithms. Frequent Itemset-based Hierarchical Clustering (FIHC), for document clustering based on the idea of frequent itemsets proposed by Agrawal et. al. The intuition of our clustering criterion is that there are some frequent itemsets for each cluster (topic) in the document set, and different clusters share few frequent itemsets. A frequent itemset is a set of words that occur together in some minimum fraction of documents in a cluster. Therefore, a frequent itemset describes something common to many documents in a cluster. In this technique use frequent itemsets to construct clusters and to organize clusters into a topic hierarchy. Here are the features of this approach.

- *Reduced dimensionality*. This approach use only the frequent items that occur in some minimum fraction of documents in document vectors, which drastically reduces the dimensionality of the document set. Experiments show that clustering with reduced dimensionality is significantly more efficient and scalable. This decision is consistent with the study from linguistics (Longman Lancaster Corpus) that only 3000 words are required to cover 80% of the written text in English and the result is coherent with the Zipf's law and the findings in Mladenic et al. and Yang et al.

- *High clustering accuracy*. Experimental results show that the proposed approach FIHC outperforms best documents clustering algorithms in terms of accuracy. It is robust even when applied to large and complicated document sets.

- *Number of clusters as an optional input parameter*. Many existing clustering algorithms require the user to specify the desired number of clusters as an input parameter. FIHC treats it only as an optional input parameter. Close to optimal clustering quality can be achieved even when this value is unknown.

III. Hierarchical Analysis Model

A hierarchical clustering algorithm creates a hierarchical decomposition of the given set of data objects. Depending on the decomposition approach, hierarchical algorithms are classified as agglomerative (merging) or divisive (splitting). The agglomerative approach starts with each data point in a separate cluster or with a certain large number of clusters. Each step of this approach merges the two clusters that are the most similar. Thus after each step, the total number of clusters decreases. This is repeated until the desired number of clusters is obtained or only one cluster remains. By contrast, the divisive approach starts with all data objects in the same cluster. In each step, one cluster is split into smaller clusters, until a termination condition holds. Agglomerative algorithms are more widely used in practice. Thus the similarities between clusters are more researched

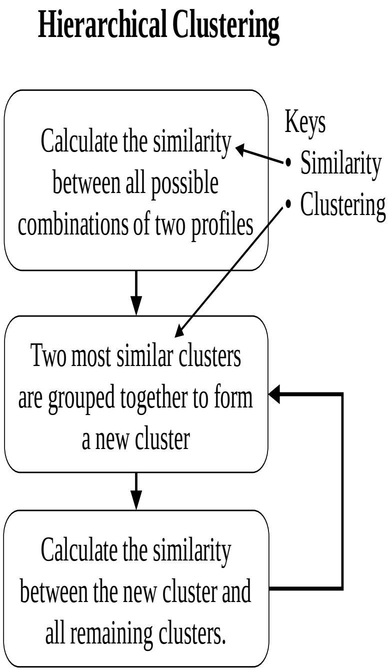

Hierarchical Clustering
Calculate the similarity between all possible combinations of two profiles
Keys
Similarity
Clustering
Two most similar clusters are grouped together to form a new cluster
Calculate the similarity between the new cluster and all remaining clusters.

Figure 1: Hierarchical Clustering Model

IV. How They Work

Given a set of N items to be clustered, and an N*N distance (or similarity) matrix, the basic process of hierarchical clustering is this:

STEP1 - Start by assigning each item to a cluster, so that if you have N items, you now have N clusters, each containing just one item. Let the distances (similarities) between the clusters the same as the distances (similarities) between the items they contain.

STEP2 - Find the closest (most similar) pair of clusters and merge them into a single cluster, so that now you have one cluster less with the help oh tf - itf.

STEP3 - Compute distances (similarities) between the new cluster and each of the old clusters.

STEP4 - Repeat steps 2 and 3 until all items are clustered into a single cluster of size N.

Step 3 can be done in different ways, which is what distinguishes single-linkage from complete-linkage and average-linkage clustering. In single-linkage clustering (also called the connectedness or minimum method), considering the distance between one cluster and another cluster to be equal to the shortest distance from any member of one cluster to any member of the other cluster.

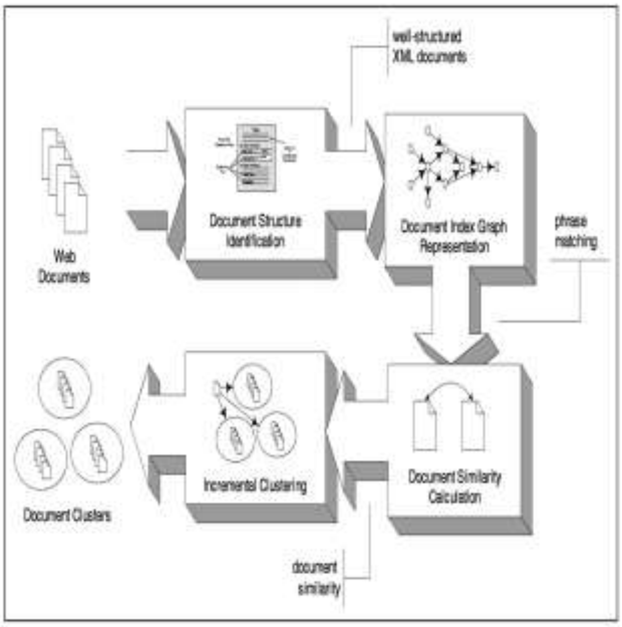
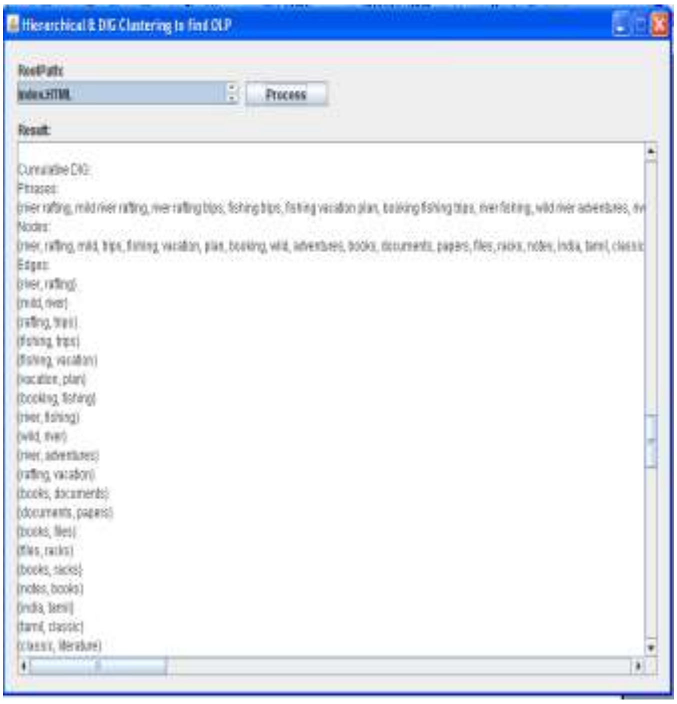

well-structured XML documents
Web Documents
Document Structure Identification
Document Index Graph Representation
phrase matching
Document Clusters
Incremental Clustering
Document Similarity Calculation
document similarity

Figure 2: Proposed Architecture

If the data consist of similarities, consider the similarity between one cluster and another cluster to be equal to the greatest similarity from any member of one cluster to any member of the other cluster. In complete-linkage clustering (also called the diameter or maximum method), consider the distance between one cluster and another cluster to be equal to the greatest distance from any member of one cluster to any member of the other cluster.
In average-linkage clustering, consider the distance between one cluster and another cluster to be equal to the average distance. This kind of hierarchical clustering is called agglomerative because it merges clusters iteratively. There is also adivisive hierarchical clustering which does the reverse by starting with all objects in one cluster and subdividing them into smaller pieces. Divisive methods are not generally available, and rarely have been applied.

Of course there is no point in having all the N items grouped in a single cluster but, once the complete hierarchical tree is obtained and need k clusters, k-1 longest links are eliminated.



```
Cumulative DIG:
Phrases:
(river rafting, mild river rafting, river rafting trips, fishing trips, fishing vacation plan, booking fishing trips, river fishing, wild river adventures, riv
Nodes:
(river, rafting, mild, trips, fishing, vacation, plan, booking, wild, adventures, books, documents, papers, files, racks, notes, india, tamil, classic
Edges:
(river, rafting)
(mild, river)
(rafting, trips)
(fishing, trips)
(fishing, vacation)
(vacation, plan)
(booking, fishing)
(river, fishing)
(wild, river)
(river, adventures)
(rafting, vacation)
(books, documents)
(documents, papers)
(books, files)
(files, racks)
(books, racks)
(notes, books)
(india, tamil)
(tamil, classic)
(classic, literature)
```

Figure 3: Node and Edge Identifcation

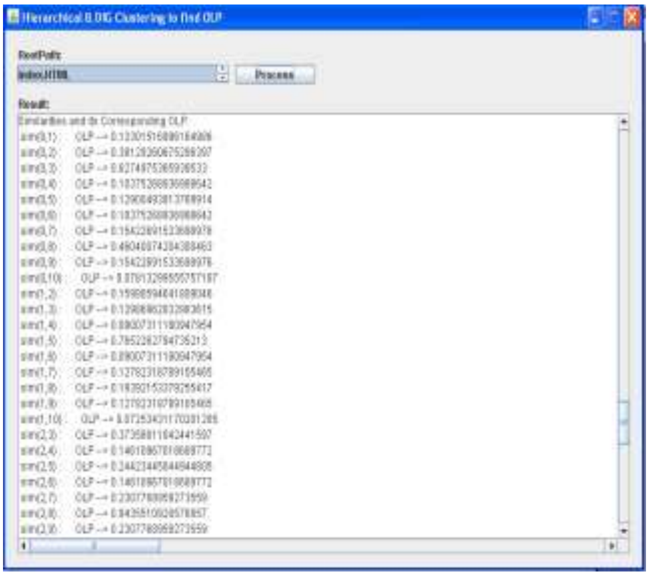
This involves the document similarity analysis and thereby finding the Overlapping Rate (OLP Rate).

By taking into account these two factors — term frequency (TF) and inverse document frequency (IDF) — it is possible to assign "weights" to search results and therefore ordering them statistically. Put another way, a search result's score ("ranking") is the product of TF and IDF:

TFIDF = TF * IDF where:

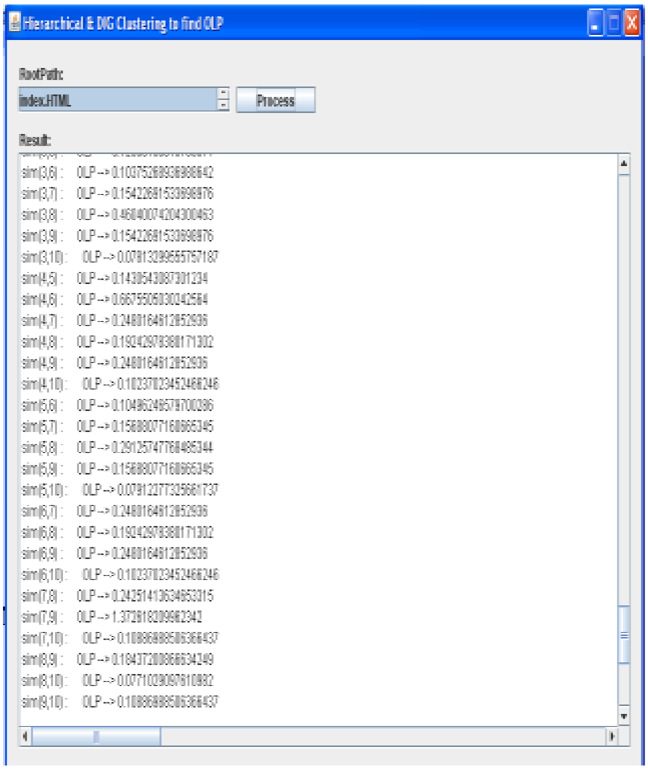
TF = C / T where C = number of times a given word appears in a document and T = total number of words in a document

IDF = D / DF where D = total number of documents in a corpus, and DF = total number of documents containing a given word



```
Similarities and its Corresponding OLP
sim(0,1) :   OLP --> 0.12301516880164606
sim(0,2) :   OLP --> 0.38128260675288297
sim(0,3) :   OLP --> 0.6274975365936533
sim(0,4) :   OLP --> 0.18375268836988642
sim(0,5) :   OLP --> 0.12900493813788914
sim(0,6) :   OLP --> 0.18375268836988642
sim(0,7) :   OLP --> 0.15422891533688978
sim(0,8) :   OLP --> 0.49040074284388463
sim(0,9) :   OLP --> 0.15422891533688978
sim(0,10) :   OLP --> 0.07813298555757187
sim(1,2) :   OLP --> 0.15988594641889048
sim(1,3) :   OLP --> 0.12908862832983615
sim(1,4) :   OLP --> 0.09007311190947954
sim(1,5) :   OLP --> 0.7852282794735213
sim(1,6) :   OLP --> 0.09007311190947954
sim(1,7) :   OLP --> 0.12782318789165465
sim(1,8) :   OLP --> 0.18392153378255417
sim(1,9) :   OLP --> 0.12782318789165465
sim(1,10) :   OLP --> 0.07253430117020128
sim(2,3) :   OLP --> 0.37358811642441597
sim(2,4) :   OLP --> 0.14618867818689772
sim(2,5) :   OLP --> 0.24423445844944808
sim(2,6) :   OLP --> 0.14618867818689772
sim(2,7) :   OLP --> 0.23077889689273959
sim(2,8) :   OLP --> 0.8435510920578857
sim(2,9) :   OLP --> 0.23077889689273959
```

Figure 4: Similarity Calcuation



```
sim(3,6) :    OLP --> 0.10375268936988642
sim(3,7) :    OLP --> 0.15422681533698976
sim(3,8) :    OLP --> 0.46040074204300463
sim(3,9) :    OLP --> 0.15422681533698976
sim(3,10) :    OLP --> 0.07813299655757187
sim(4,5) :    OLP --> 0.14305430873012 34
sim(4,6) :    OLP --> 0.66755060302425 84
sim(4,7) :    OLP --> 0.2480164612852936
sim(4,8) :    OLP --> 0.19242978388171302
sim(4,9) :    OLP --> 0.2480164612852936
sim(4,10) :    OLP --> 0.10237023452466246
sim(5,6) :    OLP --> 0.10496249657970296
sim(5,7) :    OLP --> 0.15688077160665345
sim(5,8) :    OLP --> 0.29125747768485344
sim(5,9) :    OLP --> 0.15688077160665345
sim(5,10) :    OLP --> 0.07912277325661737
sim(6,7) :    OLP --> 0.2480164612852936
sim(6,8) :    OLP --> 0.19242978388171302
sim(6,9) :    OLP --> 0.2480164612852936
sim(6,10) :    OLP --> 0.10237023452466246
sim(7,8) :    OLP --> 0.24251413634853315
sim(7,9) :    OLP --> 1.372618209962342
sim(7,10) :    OLP --> 0.10886885063668437
sim(8,9) :    OLP --> 0.18437200866634249
sim(8,10) :    OLP --> 0.0771023097810882
sim(9,10) :    OLP --> 0.10886885063668437
```

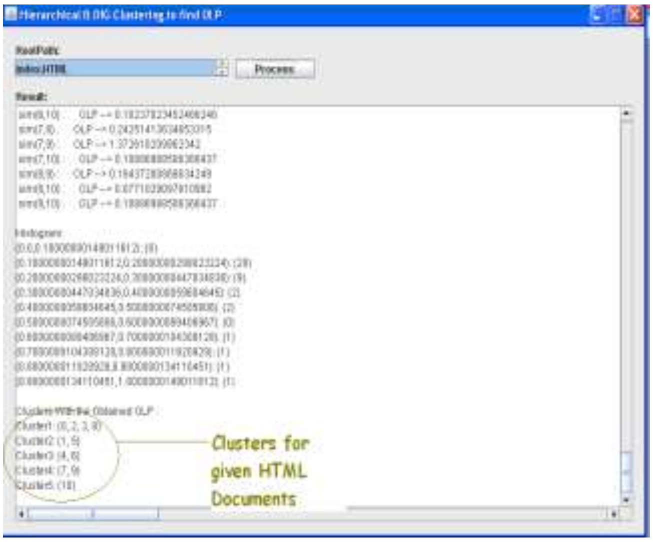
Figure 5: Similarity Calculation with OLP values

HISTOGRAM FORMATION

After finding the similarity and the OLP Rate, Histogram is formed. Histogram is also called as Dendogram.

CLUSTER FORMATION

Then the final step is the formation of clusters. This is shown in the below figure. Thus the Document clustering using Hierarchical Clustering is done and the causes are documented.


Hierarchical & DIG Clustering to find OLP
RootPath:
index.HTML
Process
Result:
Clusters for given HTML Documents

```
sim(6,10) :    OLP --> 0.10237023452466246
sim(7,8) :    OLP --> 0.24251413634853315
sim(7,9) :    OLP --> 1.372618209962342
sim(7,10) :    OLP --> 0.10886885063668437
sim(8,9) :    OLP --> 0.18437200866634249
sim(8,10) :    OLP --> 0.0771023097810882
sim(9,10) :    OLP --> 0.10886885063668437

Histogram :
[0.0,0.10000000149011612): (8)
[0.10000000149011612,0.20000000298023224): (28)
[0.20000000298023224,0.30000000447034836): (9)
[0.30000000447034836,0.4000000059604645): (2)
[0.4000000059604645,0.5000000074505806): (2)
[0.5000000074505806,0.6000000089406967): (0)
[0.6000000089406967,0.7000000104308128): (1)
[0.7000000104308128,0.800000011920929): (1)
[0.800000011920929,0.9000000134110451): (1)
[0.9000000134110451,1.0000000149011612): (1)

Clusters With the Obtained OLP :
Cluster1 : (0, 2, 3, 8)
Cluster2 : (1, 5)
Cluster3 : (4, 6)
Cluster4 : (7, 9)
Cluster5 : (10)
```

Figure 6: Efficient Cluster Formation

The clustering approach proposed here is an incremental dynamic method of building the clusters. An overlapped cluster model is adopted here. The key concept for the similarity histogram-based clustering method is to keep each cluster at a high degree of coherency at any time.Represention of the coherency of a cluster is called as Cluster Similarity Histogram.

Cluster Similarity Histogram is a concise statistical representation of the set of pair-wise document similarities distribution in the cluster. A number of bins in the histogram correspond to fixed similarity value intervals. Each bin contains the count of pair-wise document similarities in the corresponding interval [8].

The below graph shows a typical cluster similarity histogram, where the distribution is almost a normal distribution. A perfect cluster would have a histogram where the similarities are all maximum, while a loose cluster would have a histogram where the similarities are all minimum.

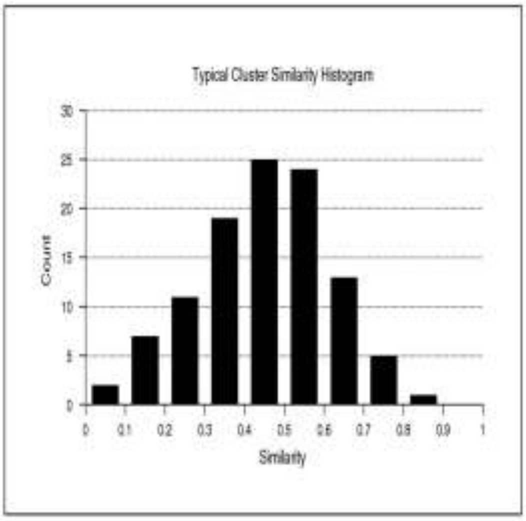


Figure 7: Efficiency Improved Results

V. Conclusion

Given a data set, the ideal scenario would be to have a given set of criteria to choose a proper clustering algorithm to apply. Choosing a clustering algorithm, however, can be a difficult task. Even ending just the most relevant approaches for a given data set is hard. Most of the algorithms generally assume some implicit structure in the data set. One of the most important elements is the nature of the data and the nature of the desired cluster. Another issue to keep in mind is the kind of input and tools that the algorithm requires. This report has a proposal of a new hierarchical clustering algorithm based on the overlap rate for cluster merging. The experience in general data sets and a document set indicates that the new method can decrease the time cost, reduce the space complexity and improve the accuracy of clustering. Specially, in the document clustering, the newly proposed algorithm measuring result show great advantages. The hierarchical document clustering algorithm provides a natural way of distinguishing clusters and implementing the basic requirement of clustering as high within-cluster similarity and between-cluster dissimilarity.

. In the proposed model, selecting different dimensional space and frequency levels leads to different accuracy rate in the clustering results. How to extract the features reasonably will be investigated in the future work.

There are a number of future research directions to extend and improve this work. One direction that this work might continue on is to improve on the accuracy of similarity calculation between documents by employing different similarity calculation strategies. Although the current scheme proved more accurate than traditional methods, there are still rooms for improvement.

REFERENCES

[1] Cole, A. J. & Wishart, D. (1970). An improved algorithm for the Jardine-Sibson method of generating overlapping clusters. The Computer Journal 13(2):156-163.

[2] D'andrade,R. 1978, "U-Statistic Hierarchical Clustering" Psychometrika, 4:58-67.

[3] Johnson,S.C. 1967, "Hierarchical Clustering Schemes" Psychometrika, 2:241-254.

[4] Shengrui Wang and Haojun Sun. Measuring overlap-Rate for Cluster Merging in a Hierarchical Approach to Color Image Segmentation. International Journal of Fuzzy Systems,Vol.6,No.3,September 2004.

[5] Jeff A. Bilmes. A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models. ICSI TR-97-021, U.C. Berkeley, 1998.

[6] E.M. Voorhees. Implementing agglomerative hierarchical clustering algorithms for use in document retrieval. Information Processing and Management, 22(6):465–476, 1986.

[7] Sun Da-fei,Chen Guo-li,Liu Wen-ju. The discussion of maximum likehood parameter estimation based on EM algorithm. Journal of HeNan University. 2002,32(4):35~41

[8] Khaled M. Hammouda, Mohamed S. Kamel , efficient phrase-based document indexing for web document clustering , IEEE transactions on knowledge and data engineering, October 2004

[9] Haojun sun, zhihui liu, lingjun kong, A Document Clustering Method Based On Hierarchical Algorithm With Model Clustering, 22nd international conference on advanced information networking and applications,

[10] Shi zhong, joydeep ghosh, Generative Model-Based Document Clustering: A Comparative Study, The University Of Texas.