# Document clustering on Hierarchical Methods

## Clustering with Multi viewpoint-Based Similarity Measure

Mr.N.Kiran
Pursuing M.Tech(CSE )
CMR Engineering college,Medchal
Hyderabad, India
neelikiran1@gmail.com

Mr G.Ravi Kumar
Associate.Professor(CSE )
CMR Engineering college,Medchal
Hyderabad, India
ravicmrcse@gmail.com

Mrs .B.Rajani
Associate.Professor(CSE )
CMR Engineering college,Medchal
Hyderabad, India
rajani.badi@gmail.com

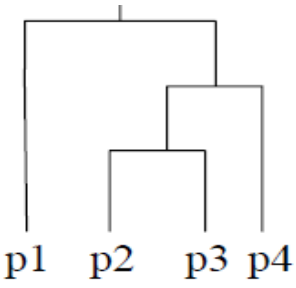
*Abstract*— **Clustering is a division of data into groups of similar objects. Representing the data by fewer clusters necessarily loses certain fine details, but achieves simplification. The similar documents are grouped together in a cluster, if their cosine similarity measure is less than a specified threshold. In this paper we mainly focuses on document clustering and measures in hierarchical clustering. 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**

**KEY Terms—Document clustering, text mining, similarity measure .Hierarchical Methods**

## INTRODUCTION

Document clustering is automatic document organization, topic extraction and fast information retrieval or filtering. It is closely related to data clustering. 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 involves the use of descriptors and descriptor extraction. Descriptors are sets of words that describe the contents within the cluster. Document clustering is generally considered to be a centralized process. 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 pairwise document similarity distribution inside clusters. Existing Systems greedily picks the next frequent item set which represent the next cluster to minimize the overlapping between the documents that contain both the item set and some remaining item sets. The clustering result depends on the order of picking up the item sets, which in turns depends on the greedy heuristic. This method does not follow a sequential order of selecting clusters. Instead, we assign documents to the best cluster. In proposed approach, The main work is to develop a novel hierarchal algorithm for document clustering which provides maximum efficiency and performance. It is particularly focused in studying and making use of cluster overlapping phenomenon to design cluster merging criteria. Proposing a new way to compute the overlap rate in order to improve time efficiency and the veracity‖ is mainly concentrated. Based on the Hierarchical Clustering Method, the usage of Expectation-Maximization (EM) algorithm in the Gaussian Mixture Model to count the parameters and make the two sub-clusters

combined when their overlap is the largest is narrated. Experiments in both public data and document clustering data show that this approach can improve the efficiency of clustering and save computing time.Hierarchical techniques produce a nested sequence of partitions, with a single, all inclusive cluster at the top and singleton clusters of individual points at the bottom. Each intermediate level can be viewed as combining two clusters from the next lower level (or splitting a cluster from the next higher level). The result of a hierarchical clustering algorithm can be graphically displayed as tree, called a dendogram. This tree graphically displays the merging process and the intermediate clusters. The dendogram at the right shows how four points can be merged into a single cluster. For document clustering, this dendogram provides a taxonomy, or hierarchical index.




***Fig;1 CHALLENGES IN HIERARCHICAL***

## DOCUMENT CLUSTERING**:**

*A. High dimensionality*

Each distinct word in the document set constitutes a dimension. So there may be 15~20 thousands dimensions. This type of high\dimensionality greatly affects the scalability and efficiency of many existing clustering algorithms

.

*B. High volume of data*

In text mining, processing of data about 10 thousands to 100 thousands documents are involved.

*C. Consistently high accuracy*:

Some existing algorithms only work fine for certain type of document sets, but may not perform well in some others.

*D. Meaningful cluster description***:**

This is important for the end user. The resulting hierarchy should facilitate browsing.

## 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)

*A. Agglomerative*:

Start with the points as individual clusters and, at each step, merge the most similar or closest pair of clusters. This requires a definition of cluster similarity or distance

*B. Divisive*:

Start with one, all-inclusive cluster and, at each step, split a cluster until only singleton clusters of individual points remain. In this case, we need to decide, at each step, which cluster to split and how to perform the split. Agglomerative techniques are more common, and these are the techniques that we will compare to K-means and its variants.

agglomerative hierarchical clustering procedure as follows:

*Simple Agglomerative Clustering Algorithm*

1. Compute the similarity between all pairs of clusters, i.e., calculate a similarity matrix whose *ijth* entry gives the similarity between the *ith* and *jth* clusters.
2. Merge the most similar (closest) two clusters.
3. Update the similarity matrix to reflect the pairwise similarity between the new

cluster and the original clusters.
4. Repeat steps 2 and 3 until only a single cluster remains.

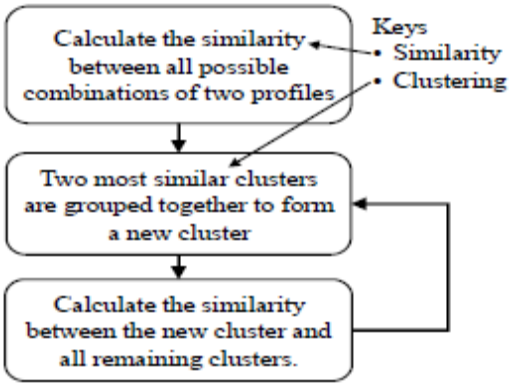



Fig:2. Hierarchical Clustering

*STEP 1* - 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
.
*STEP 2* - 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 of tf - idf.

*STEP 3* - Compute distances (similarities) between the new cluster and each of the old clusters.

*STEP 4* - 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. 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.

*Divisive 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.

Techniques:

*Intra-Cluster Similarity Technique (IST):*

This hierarchical technique looks at the similarity of all the documents in a cluster to their cluster centroid and is defined by

$$\text{Sim}(X) = \sum_{d \in X} cosine(\boldsymbol{d}, \boldsymbol{c})$$

where $d$ is a document in cluster, $X$, and $\boldsymbol{c}$ is the centroid of cluster $X$. The choice of which pair of clusters to merge is made by determining which pair of clusters will lead to smallest decrease in similarity. Thus, if cluster $Z$ is formed by merging clusters $X$ and $Y$, then we select $X$ and $Y$ so as to maximize $\text{Sim}(Z) - (\text{Sim}(X) + \text{Sim}(Y))$. Note that $\text{Sim}(Z) - (\text{Sim}(X)+\text{Sim}(Y))$ is non-positive.

*Centroid Similarity Technique (CST):*

This hierarchical technique defines the similarity of two clusters to be the cosine similarity between the centroids of the two clusters.

*UPGMA:* It defines the cluster similarity as follows

$$similarity(cluster1, cluster2) = \frac{\sum_{\substack{d_1 \in cluster1 \\ d_2 \in cluster2}} cosine(\boldsymbol{d}_1, \boldsymbol{d}_2)}{size(cluster1) * size(cluster2)}$$

where *d1* and *d2* are, documents, respectively, in cluster1 and cluster2.

## TERM FREQUENCY -INVERSE DOCUMENT FREQUENCY

The TF-IDF is a text statistical-based technique which has been widely used in many search engines and information retrieval systems. Assume that there is a corpora of 1000 documents and the task is to compute the similarity between two given documents (or a document and a query). The following describes the steps of acquiring the similarity .

*Document pre-processing steps:*

*Tokenization:* A document is treated as a string (or bag of words), and then partitioned into a list of tokens.
Removing stop words: Stop words are frequently occurring, insignificant words. This step eliminates the stop words.
*Stemming word:* This step is the process of conflating tokens to their root form

*Document representation*

Generating N-distinct words from the corpora and call them as index terms (or the vocabulary). The document collection is then represented as a N-dimensional vector in term space.

*Computing Term weights*

Term Frequency.
Inverse Document Frequency.
Compute the TF-IDF weighting.

## TFIDF Analysis

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:

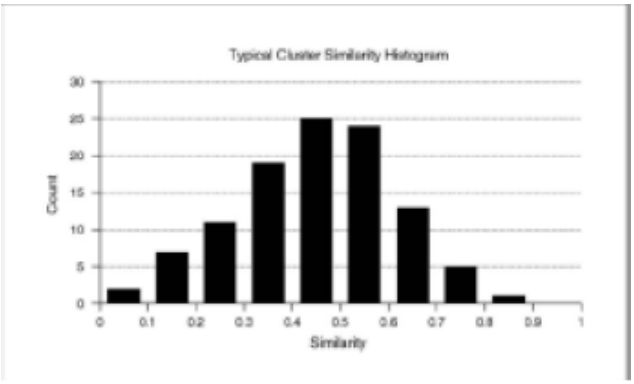
- TF = C / T where C = number of times a given word appears in a document and T = total number of words in a document.
- Document IDF = D / DF where D = total number of documents in a corpus, and DF = total number of documents containing a given word.

*Automatic classification:*

TDIDF can also be applied a priori to indexing/searching to create browse lists hence, automatic classification. Consider the table where each word is listed in a sorted TFIDF order: Given such a list it would be possible to take the first three terms from each document and call them the most significant subject "tags". Thus, Document #1 is about airplanes, shoes, and computers. Document #2 is about Milton, Shakespeare, and cars. Document #3 is about buildings, ceilings, and cleaning. Probably a better way to assign "aboutness" to each document is to first denote TFIDF lower bounds and then assign terms with greater than that score to each document. Assuming lower bounds of 0.2, Document #1 is about airplanes and shoes. Document #2 is about Milton, Shakespeare, cars, and books. Document #3 is about buildings, ceilings, and cleaning.

| Doc 1 | Doc 2 | Doc 3 |
|---|---|---|
| Word | Word | Word |
| Airplane | book | Building |
| Blue | car | Car |
| Chair | chair | Carpet |
| Computer | justice | Ceiling |
| Forest | milton | chair |
| justice | newton | cleaning |
| Love | pond | justice |
| Might | rose | libraries |
| Perl | shakespeare | newton |
| Rose | slavery | perl |
| Shoe | thesis | rose |
| Thesis | truck | science |

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. Representation of the coherency of a cluster is called as Cluster Similarity Histogram.




*Cumulative Document:*

The cumulative document is the sum of all the documents, containing meta-tags from all the documents. We find the references in the input base document and read other documents and then find references in them and so on. Thus in all the documents their meta-tags are identified, starting from the base document.

## 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.

## REFERENCES:

[1] Eui-Hong (Sam) Han, Daniel Boley, MariaGini, Robert Gross, Kyle Hastings, George Karypis, Vipin

[2] Kumar, B. Mobasher, and Jerry Moore, WebAce: A Web Agent for Document Categorization and Exploration. Proceedings Of The 2nd International Conference on Autonomous Agents (Agents'98).

[3] Daphe Koller and Mehran Sahami, hierarchical classifying documents using very few words, Proceedings of the 14th International conference on Machine Learning (ML), Nashville, tennessee,July 1997, Pages 170-178

[4] Gerald Kowalski, Information Retrieval System Theory and Implementation, Kluwer Academic Publishers, 1997.

[5] S. Zhong, "Efficient Online Spherical K-means Clustering," Proc.IEEE Int'l Joint Conf .Neural Networks (IJCNN), pp. 3180

[6] Banerjee, I. Dhillon, J. Ghosh, and Sra,clustering on the Unit Hypersphere using Vonmisesfisher Distributions,"J. Machin Learning Research, vol. 6Sept. 2005..

[7] I.S. Dhillon, S. Mallela, and D.S .Modha information-Theoretic Co-Clustering, Proc 9th ACM SIG KDD Int'l Conf. Knowledge Discovery and data Mining (KDD), pp. 89 98, 2003.

[8] S. Zhong and J. Ghosh, "A Comparative study of Generative Models for Document clustering," Proc. SIAM Int'l Conf. Data tering," Proc. SIAM Int'l Conf. Data Mining WorkshopClustering High dime-sional Data and Its Applications, 2003.

[9] Strehl, J. Ghosh, and R. Mooney,Impact of similarity Measures on Web-Page clustering Proc. 17th Nat'l Conf.ArtificiaIntelligence: for Web search (AAAI),pp. 58-64, July 2000.

[10] Moses Charikar, Chandra Chekuri, Tomas Feder, and Rajeev Motwani, Incremental Clustering and Dynamic Information Retrieval, STOC 1997, Pages 626-635 1997.

[11] Richard C. Dubes and Anil K. Jain,Algorithms for Clustering Data, Prentice Hall, 1988.

[12] A. El-Hamdouchi and P. Willet,Comparison of Hierarchic Agglomerative Clustering Methods for Document Retrieval The Computer Journal, Vol. 32, No. 3, 1989

[13] Sudipto Guha, Rajeev Rastogi, and Kyuseok Shim, (1998*), ROCK:* A Robust Clustering Algorithm for Categorical Attributes, In Proceedings of the 15th International Conference on Data eng.