International Journal of Emerging Trends in Engineering and Basic Sciences (IJEEBS)
ISSN (Online) 2349-6967
Volume 2, Issue 3(May-June 2015), PP 102-115

Soft Computing Approach for Hierarchical Document Clustering Based on Inter-Similarity Between Clusters

Ashish Jaiswal[1], Bireshwar Ganguly[2], Devashri Raich[3]

[1](Department of Computer Science and Engineering, PIGCE/RTM Nagpur University, Maharashtra, India)
[23](Dept.t of Information Technology, Rajiv Gandhi College of Engg,Chandrapur,/ Gondwana University, India

Abstract :- In today's world lot of text documents are stored and retrieved electronically and these text documents are increasing tremendously in the internet. To provide the user with relevant information Document Clustering is important tool for many versatile applications such as search engines and document browsers. Document Clustering is the process of grouping similar documents within a cluster. Hierarchical Document Clustering is one of the Clustering methods to clusters the documents and is often portrayed as better quality clustering approach. Though many clustering algorithms are available but the challenges of high dimensionality, clustering accuracy, high volume of data and meaningful cluster labels still exists. In this paper, we propose an algorithm which uses soft computing approach for hierarchical document clustering. Based on the key terms candidate clusters are defined, which further classifies documents into different level of clusters depending on inter-similarity value between the clusters. Result show that hierarchical document clustering performs better than FIHC, k means and UPGMA clustering algorithm on different datasets.

Keywords :- Hierarchical Document Clustering, Soft Computing, K-Means, Frequent Itemsets .

I. INTRODUCTION

In today's world there is an explosive growth in the electronic data and this vast amount of data are collected daily in various storage devices from the business, science and engineering, Medical and Health industries. The advancement in the technology of Computer hardware has provided us to store data in powerful data collection equipments. These huge numbers of databases are used for the management of transactions, retrieval of information and data analysis. It is very difficult to get the valuable information from this data efficiently. Data Mining is a powerful tool that uncovers valuable information from the tremendous amount of data. Document Clustering is a process of grouping text documents into clusters so that document within a cluster have high similarity in comparison to one another, but are dissimilar to documents in other clusters. Clustering is also known as unsupervised learning as no labelled documents are provided in clustering. The following are the typical requirements for a good clustering algorithm:

High dimensionality: While clustering, each keyword can be regarded as a dimension and there are thousands of keywords. Most clustering algorithms are good at handling low dimensional data but finding clusters in high dimensional space is challenging.

Scalability: Many clustering algorithms work well on small data sets containing hundred of documents but fail to handle large data sets containing millions of documents. Therefore highly scalable clustering algorithms are needed.

Accuracy: The documents within the same cluster should be similar and dissimilar to documents in other clusters i.e. there should be high intra cluster similarity and low inter cluster similarity.

Browsing with meaningful cluster description: Each cluster in the tree should have a cluster label which a user may get support for interactive browsing.

Prior domain knowledge: Many clustering algorithms require user to provide input parameters such as the number of clusters. Clustering results are sensitive to such input parameters.

Incremental clustering and insensitivity to input order: Incremental updates may arrive at any time. Some clustering algorithms cannot incorporate incremental updates into existing clustering structures and have

to recompute a new clustering from scratch. Clustering algorithms may return different clustering depending on the order in which documents are presented. Incremental algorithms and algorithms that are insensitive to the input order are needed.

1.1 Document clustering methods

Major clustering algorithms are divided into partitioning methods and hierarchical methods. Hierarchical method creates a hierarchical decomposition of given sets of documents. It can be further classified as agglomerative and divisive based on how the hierarchical decomposition is formed. Algorithms belonging to the family of agglomerative build the hierarchy bottom up by iteratively computing the similarity between all the pairs of clusters and then merging the most similar pair. The divisive approach builds the hierarchy in the top down fashion. In each iteration a cluster is split into small clusters until a certain termination condition is satisfied. Hierarchical clustering is often portrayed as the better quality clustering approach but is limited because of its quadratic time complexity.

Partitioning method partitions the documents into the given number of clusters. It then uses iterative relocation technique that attempts to improve the partitioning by moving documents from one cluster to another. K means and its variants are the best known partitioning methods. Partitioning methods have a time complexity that is linear to the documents but are thought to produce inferior clusters. The incorrect estimation of input parameter, may lead to poor clustering accuracy.

To get the best of both the worlds, sometimes, Agglomerative hierarchical and K-means are combined to get the best features of quality clustering from agglomerative, and run time efficiency from K-means.

II. REVIEW OF WORK

M. Steinbach, G. Karypis, V. Kumar [1] in 2000 studied experimentally agglomerative hierarchical clustering and K- means (both the standard and bisecting k means). The authors discovered that a simple and efficient Bisecting k- means produce better clusters of documents as compared to regular k- means and as good as or better than produce by agglomerative hierarchical clustering technique. Two metrics has been used for the evaluation of cluster quality: Entropy and F- Measure. Entropy provides a measure of goodness for un-nested clusters or for the clusters at one level of a hierarchical clustering. F- Measure, measures the effectiveness of a hierarchical clustering.

Florian Beil, Martin Ester, Xiaowei Xu [2] in 2002 introduced an approach which uses frequent itemsets for text clustering. Bisecting k means outperforms hierarchical clustering algorithms with respect to cluster quality [1] but it does not address the problems of high dimensionality, large size of databases and meaningful cluster labels [2]. To overcome these problems authors presented an approach which uses frequent itemsets. Frequent itemsets are sets of terms co-occurring in more than a threshold percentage of all documents of a database. Such frequent itemsets can be efficiently discovered using Apriori Algorithm. Authors present two Greedy algorithms FTC and HFTC for frequent term based clustering. FTC (Frequent Term-based Clustering) determines a flat clustering i.e. unstructured set of clusters covering the whole database. HFTC generates hierarchical clustering which are easy to browse and more comprehendible than hierarchies discovered by other comparison algorithms.

By focusing on frequent itemsets dimensionality of the document set is drastically reduced. B. Fung, Ke Wang and M. Ester [3] in 2003 proposed a novel approach; Frequent Itemset based Hierarchical Clustering (FIHC) for document clustering based on the idea of frequent itemsets.Each cluster is identified by some common words called frequent itemsets, for the documents in the cluster. Frequent itemsets are also used to produce a hierarchical topic tree for clusters.In FIHC each document is represented by a vector of weighted frequencies (term frequency x inverse document frequency). For each global frequent itemset an initial cluster

is constructed to include all the documents containing this itemset. FIHC utilizes this global frequent itemset as the cluster label to identify the cluster. For each document best initial cluster is identified and the document is assigned to the best matching initial cluster by means of score function. After this each document belongs to exactly one cluster. The next step is to build cluster tree where each cluster except the root node has exactly one parent. The topic of parent is more general than the child cluster. The cluster tree is built bottom up by choosing the best parent for the child cluster. The next step is to prune the tree. If two sibling clusters are very similar, they should be merged into one. Experimental results show that FIHC apparently outperforms all other algorithms in terms of accuracy, efficiency and scalability.

Yehang Zhu, Benjamin C. M. Fung, Dejun Mu, Yanling Li [5] in 2008 proposed an approach which is a hybrid version of partitioning and agglomerative clustering approaches. This method inherits the merit of efficiency from the partitioning approach and the hierarchical structure from agglomerative approach. Frequent itemsets clustering performs clustering operation on selected frequent items. Frequent itemsets clustering is scalable but it discards many useful non frequent words for cluster analysis. The approach proposed by authors consists of two phases. In the first phase by using partitioning method, group the document objects into lot of clusters. Then the agglomerative hierarchical clustering is applied to merge clusters based on their inter connectivity and closeness which is an idea adopt from CHAMELEON clustering algorithm. This is the key to achieve efficiency and scalability. This method utilizes all words of the document set as compared to frequent itemset clustering method. Experimental results show that proposed approach consistently outperforms others in most cases for cluster quality as well as for efficiency.

Anuj Sharma, Renu Dhir [6] in 2010 proposed WDC an efficient clustering algorithm based closed word sets. It uses hierarchical approach to cluster text documents having common words. FIHC fails when the number of frequent itemsets is large. Thus the authors presented a novel approach WDC in which first global frequent wordsets and frequent closed wordsets are searched in the documents. For each global frequent wordset, an initial cluster is formed containing documents that have that wordset. After that clusters are disjoint that contain similar sets of documents by means of score function. The resulting clusters will contain documents that share a similar set of words. Experimental results show that WDC outperforms the other competitors in terms of accuracy.

Rekha Baghel, Renu Dhir [7] in 2010 proposed an approach based on frequent concepts which is different from frequent items. Frequent Concepts based Document Clustering (FCDC) utilizes the semantic relationship between words to create concepts. this is the special feature of FCDC where it treats the documents as set of related words instead of bag of words. Different words share the same meaning is known as synonyms. Set of these different words that have same meaning is known as concept. This algorithm first creates a feature vector based on concepts identified by Wordnet ontology. Wordnet is a large lexical database, a combination of dictionary and thesaurus for English language. After creating feature vector based on concepts Apriori paradigm is utilized for finding frequent itemsets, to find frequent concepts from feature vector. Then the initial clusters are formed by assigning one frequent concept to each cluster. the algorithm processes the initial clusters makes final clusters arranged in hierarchical structure. The experiments were performed on Classis, Wap and Re0 Dataset and compared with FIHC, UPGMA and Bisecting K- Means. F- Measure is used for the evaluation of accuracy. The experimental results show that FCDC has higher F –Measure values as compared to other algorithms therefore FCDC provides more accuracy as compared to Bisecting K-Means, UPGMA and FIHC.

Yeupeng Cheng, Tong Li and Song Zhu [8] in2010 proposed a document clustering technique based on term clustering and association rule. In this technique first the words are extracted from document collection and then terms are clustered according to the Average Mutual Information between terms. Document vector space model is represented by term clustering and then association rule are applied to mine document clustering. This technique uses Direct hashing and pruning (DHP) algorithm to mine the association rules which is an advanced algorithm of Apriori. In the experiment this technique is compared with partitioning method to test the

effectiveness. The performance and quality of document clustering are better as compared to partitioning method.

S. Krishna, S. Bhavani [10] in 2010 proposed an efficient approach for text clustering based on Frequent itemsets where the text documents are pre-processed by removing stop words and applying stemming algorithm. Then top t-frequent words are extracted from each document and the binary mapped database is formed through the extracted words. Apriori algorithm is then applied to discover frequent itemsets having different length. The mined frequent itemsets are sorted in descending order based on their support level for every length of itemsets. The documents are split into partition using the sorted frequent itemsets. These frequent itemsets can be viewed as understandable description of the obtained partitions. The resultant cluster is formed within the partition using the derived keywords the performance was evaluated using F-Measure and the clustering performance of the approach was analyzed.

Rakesh Agrawal, Ramakrishnan Shikant [11] in 1994 considered the problem of discovering association rule between items in large database of sales transaction. The authors presented two algorithms, Apriori and AprioriTid for discovering all significant association rules between items in large databases of transactions. Experimental results show that both the algorithms always outperform the other known algorithms. The best features of two algorithms can be combined into a hybrid algorithm called AprioriHybrid. Scale-up experiments showed that AprioriHybrid scales linearly with the number of transactions. The execution time decreases a little as the number of items in the database increases. Experiments demonstrate the feasibility of using AprioriHybrid in real applications involving very large databases.

III. PROPOSED METHOD

Figure 1 shows the framework of our proposed approach where the whole process is divided into three parts.

Pre processing
Extraction of candidate clusters
Generation of target clusters

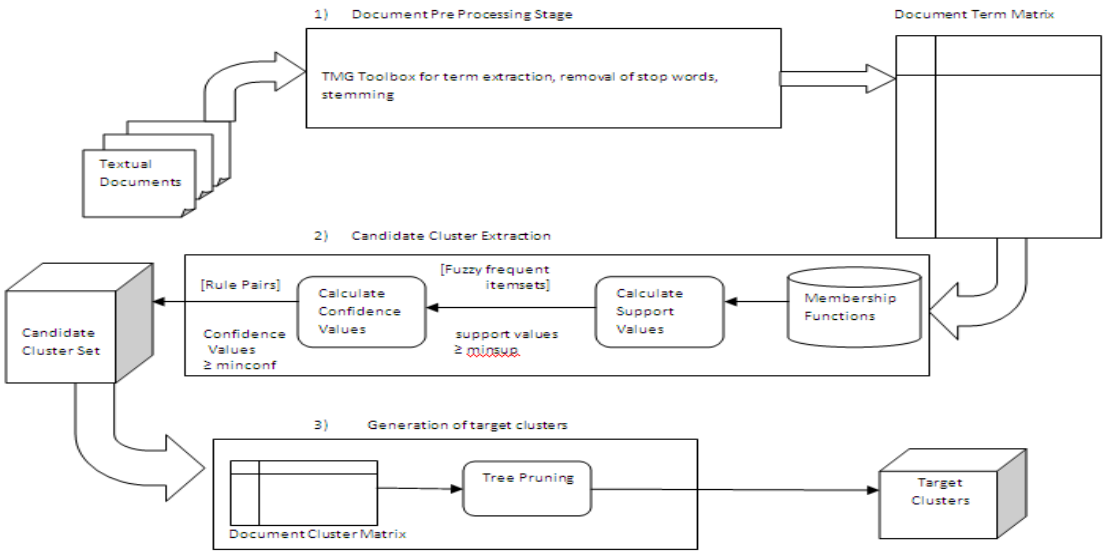

Figure 1.Framework of proposed approach

3.1 Pre- Processing by using TMG Toolbox

The preprocessing basically consists of a process to optimize the list of terms that identify the collection. The aim of the preprocessing phase is to prune from the documents all characters and terms with poor information that affects the quality of clustering algorithm. Preprocessing has several steps that take a text document as input and outputs the set of tokens to be used in feature vector. These steps involve dividing of sentences into terms, then removing the stop words i.e. Words that do not convey any significant information (for e.g. is, the, for), then, stemming of words for bringing the words to its stem (for e.g. association, associating can be stemmed to associate) and finally representation of documents where each document is represented by the term frequency vector in the vector space model.

These several steps are fulfilled in the proposed approach with the help of TMG toolbox. Dimitrius Zeimpekis, Efstratios Gallopoulos [20] built the TMG toolbox for the generation and incremental modifications of term-document matrices from text collections. The TMG toolbox is written entirely in MATLAB and is used in research and educational contexts to streamline document preprocessing and prototyping of algorithms for information retrieval. TMG parses single or multiple file or entire directories containing text, performs the necessary preprocessing, such as stopword removal and stemming and constructs a Term- Document Matrix (TDM) according to parameter set by user.

3.2 Extraction of Candidate Clusters

This stage takes a document set D i.e. the output of previous stage; set of predefined membership functions, minimum support value θ and minimum confidence value λ as input and it gives output a set of candidate clusters. The membership functions are used to convert each term frequency into a fuzzy set where the term- frequency fuzzy set of document is a pair (F, W) where F is a set and equals (FLij (nij)/ tj).low + (FMij (nij)/ tj).mid + (FHij (nij)/tj). High. The notation tj. r is called as the fuzzy region of tj.

The following are the predefined membership functions:

$$F^{L}_{ij}(n_{ij}) = \begin{cases} 0, & n_{ij} = 0 \\ 1 + n_{ij}/x_1, & 0 < n_{ij} < x_1 \\ 2, & n_{ij} = x_1, \quad x_1 = \min(n_{ij}),\ x_2 = \text{avg}(n_{ij}) \\ 1 + x_2 - n_{ij} / x_2 - x_1, & x_1 < n_{ij} < x_2 \\ 1, & n_{ij} \geq x_2 \end{cases}$$

$$F^{M}_{ij}(n_{ij}) = \begin{cases} 0, & n_{ij} = 0 \\ 1, & n_{ij} = 1 \\ 1 + n_{ij} - x_1 / x_2 - x_1, & x_1 < n_{ij} < x_2 \\ 2, \ n_{ij} = x_2, & x_1 = \min(n_{ij}),\ x_2 = \text{avg}(n_{ij}),\ x_3 = \max(n_{ij}) \\ 1 + x_3 - n_{ij} / x_3 - x_2, & x_2 < n_{ij} < x_3 \\ 1, & n_{ij} = x_2 \end{cases}$$

$$F^{H}_{ij}(n_{ij}) = \begin{cases} 0, & n_{ij} = 0 \\ 1, & n_{ij} \leq x_1 \\ 1 + n_{ij} / x_2 - x_1, & x_1 < n_{ij} < x_2, x_1 = \text{avg}(n_{ij}),\ x_2 = \max(n_{ij}) \\ 2, & n_{ij} = x_2 \end{cases}$$

The Membership Functions

The fuzzy region of each term is calculated for low, mid and high values in all the documents. Add all the low values of each term in all documents. Similarly add also mid and high values independently of each term in all documents. This gives the count value of each term for low, mid and high region. Now find the region of each key term with maximum count,

max-count = max (count low, count mid, count high).

For a document set D, a candidate cluster consists of documents that are subset of document set D which contains all the key terms $\tau = \{t1,t2,\ldots,tq\}$ which are subset of key term set. τ is a fuzzy frequent itemset for describing candidate cluster. Set of these candidate clusters is called as candidate cluster set. Now to find the fuzzy frequent-1 itemsets, we are providing here the minimum support value as 40%. If the minimum support value of τ is greater than 40%, which is obtained by division of max-count by number of documents, then those terms are included in the frequent-1 itemsets L1

The next step is to estimate the strength of association among key terms in the document set by using confidence values. We are providing here the minimum confidence value as 60%. In general the highly co-occurring terms are used together. Thus the algorithm computes confidence values of a rule pair to check the strong association of key terms (t1, t2,…,tq) of fuzzy frequent q- itemsets.

3.3 Generation of target clusters

The candidate cluster set generated in the above step can be considered as a set of topics with their corresponding sub topics in the document set. In this phase of the process, the Document Term Matrix (DTM) and Term Cluster Matrix (TCM) is constructed to obtain the Document Cluster Matrix (DCM) for assigning each document to the fitting cluster such that each cluster contains subset of document. For the documents in each of these cluster, the intra cluster similarity is minimized and the inter cluster similarity is maximized. We call these each cluster as the target cluster.

The Document Term Matrix (DTM) for a document set D, denoted [w max-Rj] is an n x p matrix such that is the weight of term tj in document di and tj €L1. Figure shows formal illustration of DTM.

$$
W = \begin{array}{c} \\ d_1 \\ d_2 \\ . \\ . \\ d_n \end{array}
\begin{array}{c}
\begin{array}{cccc} t_1 & t_2 & \ldots\ldots\ldots & t_p \end{array} \\
\begin{pmatrix}
W_{11}^{\max\text{-}Rj} & W_{12}^{\max\text{-}Rj} & \ldots\ldots & W_{1p}^{\max\text{-}Rj} \\
W_{21}^{\max\text{-}Rj} & W_{22}^{\max\text{-}Rj} & \ldots\ldots & W_{2p}^{\max\text{-}Rj} \\
. & . & . & . \\
. & . & . & . \\
W_{n1}^{\max\text{-}Rj} & W_{n2}^{\max\text{-}Rj} & \ldots\ldots & W_{np}^{\max\text{-}Rj}
\end{pmatrix}
\end{array}
\quad n \times p
$$

Illustration of Document Term Matrix

The Term Cluster Matrix (TCM) for a document set D of n documents is a p x k matrix, denoted by G = [gmax- Rj] where,

$$
g_{jl}^{\max\text{-}Rj} = \frac{\text{Score}(c_l^{\tilde{}q})}{\sum_{i=1 \text{ to } n} W_{ij}^{\max\text{-}Rj}}
$$

Where

$$\text{Score}(c_l^{\sim q}) = \begin{cases} \sum_{d_i \in c_l^{\sim 1},\, t_j \in L_1} W_{ij}^{\max - Rj} & \text{If } q = 1, \\ \sum_{d_i \in c_l^{\sim q},\, t_j \in L_1} W_{ij}^{\max - Rj} \Big/ \lambda & \text{else,} \end{cases}$$

Figure shows the formal representation of TCM

$$G = \begin{array}{c} \\ t_1 \\ t_2 \\ \vdots \\ t_p \end{array} \begin{pmatrix} c_1^{\sim 1} & \cdots & c_{l-1}^{\sim 1} & c_l^{\sim q} & \cdots & c_k^{\sim q} \\ g_{11}^{\max - Rj} & \cdots & g_{1l-1}^{\max - Rj} & g_{1l}^{\max - Rj} & \cdots & g_{1k}^{\max - Rj} \\ g_{21}^{\max - Rj} & \cdots & g_{2l-1}^{\max - Rj} & g_{2l}^{\max - Rj} & \cdots & g_{2k}^{\max - Rj} \\ \vdots & & \vdots & \vdots & & \vdots \\ g_{pl}^{\max - Rj} & \cdots & g_{pl-1}^{\max - Rj} & g_{pl}^{\max - Rj} & \cdots & g_{pk}^{\max - Rj} \end{pmatrix}$$

Illustration of Term Cluster Matrix

Document Cluster Matrix (DCM) for a document set D of n documents is the inner product of its DTM and TCM. It is an n x k matrix and can be defined as V = L1. G. The formal illustration of DCM is shown in figure.

$$V = \begin{array}{c} \\ d_1 \\ d_2 \\ \cdot \\ \cdot \\ d_n \end{array} \begin{array}{c} c_{11}^{\sim 1} \ldots\ldots\ldots\ldots c_{1l-1}^{\sim 2} \quad c_{1l}^{\sim q} \ldots\ldots\ldots c_{1k}^{\sim q} \\ v_{11} \ldots\ldots\ldots\ldots v_{1l-1} \quad v_{1l} \ldots\ldots\ldots v_{1k} \\ v_{21} \ldots\ldots\ldots\ldots v_{2l-1} \quad v_{2l} \ldots\ldots\ldots v_{2k} \\ \cdot \quad \cdot \quad \cdot \quad \cdot \\ \cdot \quad \cdot \quad \cdot \quad \cdot \\ v_{nl} \ldots\ldots\ldots\ldots v_{nl-1} \quad v_{nl} \ldots\ldots\ldots v_{nk} \end{array} \quad n \times k$$

Illustration of Document Cluster Matrix

3.4 Assigning document to the best cluster

If a low minimum support value and low minimum confidence value are used then the target cluster would be broad. The documents with the same topic may be spread to several small target clusters which gives low document clustering accuracy. To achieve higher document clustering accuracy, pruning method is used for merging similar target clusters at level 1. This includes inter cluster similarity measure to compute inter cluster similarity between two target clusters. The inter cluster similarity between two target cluster c_x^1 and c_y^1, $c_x^1 \neq c_y^1$, is defined by

$$\text{Inter\_Sim}(c_x^1, c_y^1) = \frac{\sum_{d_i \in c_x^1, c_y^1}^{n} v_{ix} \times v_{iy}}{\sqrt{\sum_{d_i \in c_x^1}^{n} (v_{ix})^2 \times \sum_{d_i \in c_y^1}^{n} (v_{iy})^2}}$$

The range of Sim is [0,1]. If the Inter Sim value is close to 1, then both clusters are considered nearly the same. The minimum Inter Sim will be used as a threshold to decide whether two target clusters should be merged. We have provided the minimum Inter-Sim value as 0.6. The target cluster pair with the highest Inter Sim value must be keep merging until the Inter sim value of all target cluster at level 1 becomes smaller than the minimum Inter Sim threshold.

IV. PROPOSED ALGORITHM

Input:

1. A document set D in terms of Document Term Matrix
2. Minimum support value θ.
3. Minimum confidence value λ.
4. Fuzzy Membership Functions
5. Minimum inter- sim threshold δ

Output: Set of target clusters

Method:

Step 1: Transform each term frequency into fuzzy set

F= (FLij (nij) / tj) .low + (FMij (nij) / tj). mid + (FHij (nij) /tj). High

Step 2: For all fuzzy regions, Calculate:

$$\text{count} = \sum_{i=1 \text{ to } n} F$$

Step 3: Find the region of each key term with maximum count

max-count = max (count low, count mid, count high)

max-R is the region with max-count for each key term.

Step.4: Find fuzzy frequent 1-itemset L1 where

$$\text{Sup}(\tau) = \text{count} / |D|$$

We will put only those items in L1,

$$L = \{\max - R \mid \sup(\tau) \geq \theta\}$$

Step 5: Generate the candidate set C2 from L1

5.1 Find all possible combination of terms

5.2 Calculate the confidence value of all terms by using the formula

$$\frac{\sum_{i=1 \text{ to } n} F}{\sum_{i=1 \text{ to } n} (F1 \Lambda F2 \Lambda F3 \Lambda \ldots\ldots\ldots \Lambda Fn)}$$

5.3 Hold those rules having confidence value ≥ λ

Step 6: Candidate cluster is generated based on the fuzzy frequent q-itemsets

Step 7: Build p x k term cluster matrix G = [gmax- R]

Step 8: Build n x k document cluster matrix V=L1 . G

$$= [Vil]$$

$$= \sum_{p=1 \text{ to } p} Wip\ gpl$$

Step 9: Assign a document to a best target cluster

9.1. Cl' = { di | vil = max {vi1, vi2,....vil} € cl~l where the number of vil is 1 };

otherwise apply rule 2.

9.2. Cl'= { di |vil = max {vi1, vi2,..., vil} € cl', where the number of vil > 1 and with cl~1 the highest fuzzy count value max-count1 corresponding to its fuzzy frequent itemsets}

V. EXPERIMENTAL RESULTS AND DISCUSSION

We have conducted experiments to compare the accuracy of proposed algorithm with FIHC, K-Means and UPGMA. F- Measure is employed to calculate the accuracy. The range of F- Measure is in between 0 to 1 and higher the F- Measure the better the clustering solution is. Below table shows comparison of the F- Measures values of proposed algorithm with FIHC, K-Means and UPGMA, conducted on different datasets.

5.1 Datasets used

We used the four datasets namely Classic 30, Tr11, Hitech and Wap which are widely adopted as standard benchmarks for the text categorization task. To find the key terms, stop words are removed and stemming was performed. Documents then were represented as term frequency vectors and unimportant terms were discarded. In this process there is significant dimensionality reduction without the loss of clustering performance.

5.2 Evaluation measures

F-Measure is a measure that combines the precision and recall ideas from information retrieval. It is commonly used as external measurement, which is employed to evaluate the accuracy of the produced clustering solutions for both flat and hierarchical clustering. More importantly, this measure balances the cluster precision and cluster recall. Hence we define a set of document clusters generated from the clustering result, denoted C and another set, denoted L, consisting of natural classes, such as each document is pre classified into a single class. Both set are derived from the same document set D. Let |D| be the number of all documents in the document set D; |ci| be the number of documents in the cluster ci € C; |lj| be the number of documents in the class lj € L; |ci ∩ lj| be the number of documents both in a cluster ci and a class lj. Fung (2002) measured the quality of clustering result C using the weighted sum of such maximum F-measures for all natural classes according to the cluster size. This measure is called the overall F-measure of C denoted F(C), and is defined as follows:

$$F(C) = \sum_{l_j \in L} \frac{|l_j|}{|D|} \max_{c_i \in C} \{F\}$$

Where

$$F = 2PR / (P+R)$$

$$P = |c_i \cap l_j| / |c_i|$$

$$R = |c_i \cap l_j| / |l_j|$$

In general the higher the F(C) values, the better the clustering solution is.

5.3 Evaluation Results

Table 1. Comparison of accuracy by using F- measure

		F - Measure			
Datasets	**No. of Clusters**	**Proposed Method**	**FIHC**	**K- Means**	**UPGMA**
Classic 30	3	0.26	N.A	0.33	N.A
	5	**0.75**	N.A	0.50	N.A
	6	N.A	N.A	N.A	N.A
Tr11	3	**0.75**	N.A	0.50	N.A
	15	**0.85**	N.A	0.60	N.A
	30	**0.75**	N.A	0.52	N.A
Hitech	3	**0.53**	0.48	0.39	0.33
	15	**0.56**	0.45	0.39	0.33
	30	**0.50**	0.46	0.40	0.47
Wap	3	**0.48**	0.37	0.30	0.39
	15	0.48	0.49	0.31	0.49
	30	**0.56**	0.56	0.40	0.48

5.4 Graphical Representation of Experimental Results

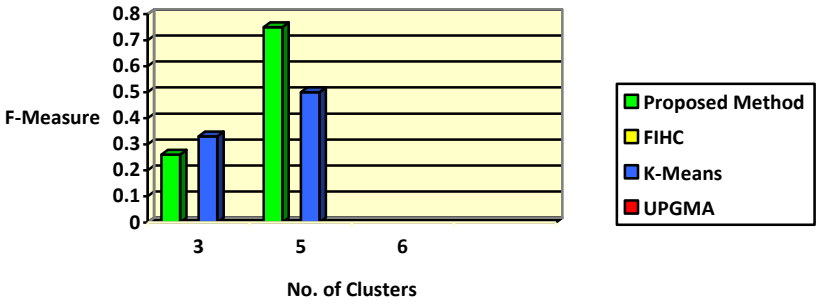

Figure 2. F measure comparison with Classic dataset

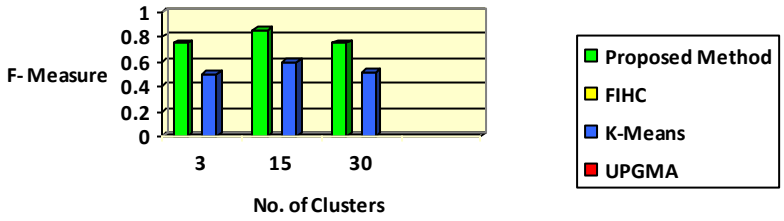

Figure 3. F-measure comparison with Tr11 dataset

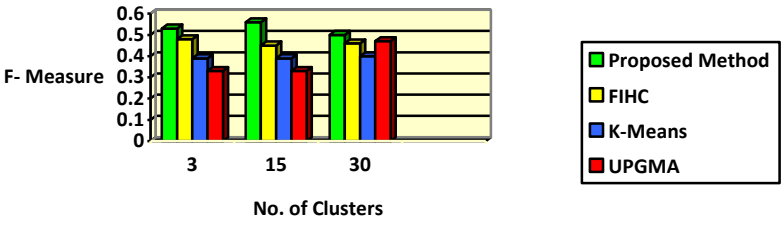

Figure 4. F measure comparison with Hitech dataset

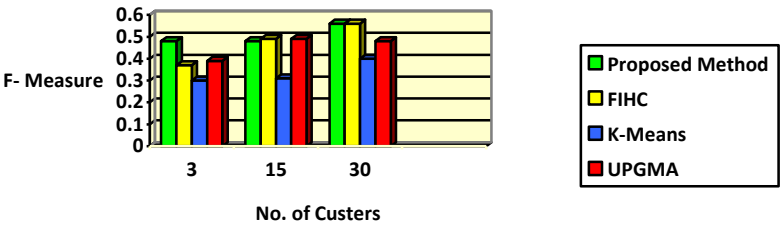

Figure 5. F measure comparison with Wap dataset

VI. CONCLUSIONS

Hierarchical document clustering is one of the promising method of document clustering. Here we proposed a new method of document clustering by using soft computing approach. Assigning fuzzy membership values and taking better confidence and threshold values improves the performance of the algorithm. F – Measure of all the algorithms have been considered to measure the accuracy. More the F- measure, better the algorithm. Results are taken on Classic30, Tr11, Wap and Hitech dataset and compared with FIHC, K-means and UPGMA clustering algorithms. In most of the cases our algorithm performs better than other.

REFERENCES

[1] Michael Steinbach, George Karypis, Vipin Kumar, (2003) "A Comparison of Document Clustering Algorithms", Army HPC Research Centre, University of Minnesota.

[2] Florien Beil, Martin Ester, Xiaowei Xu, "Frequent Term based Text Clustering", ACM *SIGKDD 02* Edmonton, Alberta, Canada.

[3] Benjamin C.M. Fung, Ke Wang and Martin Ester. (2003) "Hierarchical Document Clustering using Frequent Itemsets", Simon Fraser University, BC, Canada.

[4] Yehang Zhu, Benjamin C.M. Fung, Dejun Mu, Yanling Li, (2008) "An Efficient Hybrid Hierarchical Document Clustering Method" ,*Fifth International Conference on Fuzzy Systems and Knowledge Discovery*, IEEE 2008.

[5] Anuj Sharma, Renu Dhir., (2009) "A Wordsets based Document Clustering Algorithm for Large Datasets", *International Conference on Methods and Models in Computer Science*.

[6] Rekha Baghel, Renu Dhir, (2010) "A Frequent Concepts based Document Clustering Algorithm" *International Journal of Computer Applications (0975-8887),* Vol. 4 – No. 5, July 2010

[7] Yeupeng Cheng, Tong Li, Song Zhu. (2010), "A Document Clustering Technique based on Term Clustering and Association Rules" IEEE.

[8] V. Mary Amala Bai, D. Manimegalai. (2010) "An Analysis of Document Clustering Algorithms", ICCCCT-*10*, IEEE.

[9] S. Krishna , S. Bhavani, (2010) "An Efficient Approach for Text Clustering based on Frequent itemsets" , *European Journal of Scientific Research, ISSN 1450-216X*, Vol 42 No.3, p. 399-410.

[10] Rakesh Agrawal, Ramakrishna Srikant, (1994)"Fast Algorithms for Mining Association Rules" *In proceedings of the 20th VLDB Conference* Santiago Chile.

[11] Rakesh Agrawal, Tomasz Imielinski, Arun Swami, (1993) "Mining Association Rules between Sets of Items in Large Databases", *In proceedings of the ACM SIGMOD Conference* Washington DC, USA.

[12] Ramkrishna Srikant, Rakesh Agrawal, (1995) "Mining Generalized Association Rules", *In proceedings of the 21st VLDB Conference* ,Zurich, Switzerland.

[13] Chun- Ling Chen, Frank S.C. Tseng, Tyne Liang, (2010) "Mining Fuzzy Frequent itemsets for Hierarchical Document Clustering" *Information Processing and Management* 46, p.193-211.

[14] D.R. Cutting, D.R. Karger, J.O. Pedersen and J.W. Tukey, (1992) "Scatter/Gather : A Cluster based approach to browsing large document collections" *In proceedings of the Fifteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* p. 318-329.

[15] H.H. Malik, John Kender. (2006) "High Quality Efficient Hierarchical Document Clustering using Closed Interesting Itemsets" *In proceedings of the Sixth International Conference on Data Mining (ICDM'06)*, IEEE.

[16] X. Rui, Donald Wunsch, (2005)"Survey of Clustering Algorithms", *IEEE transactions on Neural Networks*, Vol 16, No. 3.

[17] A.K. Jain, M.N. Murty, P.J Flynn., "Data Clustering: A Review", ACM *Computing Surv*ey, Vol. 31, No.3, p. 264-323

[18] Jiawei Han, Micheline Kamber, Jian Pei. Data Mining: Concepts and Techniques. Morgan Kuafmann

[19] D. Zeimpekis and E. Gallopoulus, (2005) "TMG: A MATLAB Toolbox for Generating Tem Document Matrices from text collections"