SENTENCE-SIMILARITY BASED DOCUMENT CLUSTERING USING FUZZY ALGORITHM

G.Thilagavathi[#1], J.Anitha[#2], K.Nethra[#3],
PG Scholar[#1], Assistant Professor[#2], PG Scholar[#3],
Sri Ramakrishna Engineering College,
Coimbatore.
thilaga.apr@gmail.com, anitha.j@srec.ac.in, nethrakanagaraj@gmail.com

ABSTRACT

Due to the growth of information in web leads to drastic increase in field of information retrieval. Efficient information retrieval and navigation is provided by document clustering. Document clustering is the process of automatically grouping the related documents into clusters. Instead of searching entire documents for relevant information, these clusters will improve the efficiency and avoid overlapping of contents. Relevant document can be efficiently retrieved and accessed by means of document clustering. When compared with hard clustering, fuzzy clustering algorithms allow patterns to belong to all the clusters with differing degrees of membership. Fuzzy clustering is important in domains such as sentence clustering, since a sentence is related to more than one theme or topic present within a document or set of documents. In our proposed system, Fuzzy clustering algorithm operates on Expectation-Maximization framework in which the cluster membership probabilities for sentence in each cluster are identified. Results obtained while applying the algorithm to sentence clustering tasks demonstrate that the algorithm is capable of identifying overlapping clusters of semantically related sentences and its performance improvement can be proved by comparing with k-medoid. Performance measures such as purity, Entropy, Partition_Entropy and V_Measure are used to prove the performance improvement of document clustering and its application in document summarization.

Index Terms: Document Clustering, Information Retrieval, Sentence-Similarity, Stemming, Page Rank, Membership Probability

I. INTRODUCTION

Clustering is an unsupervised learning technique in which objects are grouped in unknown predefined clusters. The main goal of clustering is to group the similar objects. The high quality of clustering is to obtain high intra-cluster similarity and low inter-cluster similarity. Clustering can be used in many application such as Document Classification, Data compression, Image analysis, Bioinformatics, Academics, Search engines, Wireless sensor networks, Intrusion Detection etc. The major requirements that should be satisfied by clustering are scalability, dealing with different type of attributes, discovering cluster of arbitrary shapes, ability to deal with noise and outliers, high dimensionality and usability. The number of documents on the Internet is continuously increasing due to large amount of online sources available and it is very difficult for the users to go through all the sources and find the relevant information from the collection.

Document clustering (or Text clustering) is the process of automatically organizing the documents, extraction of topics and for fast information retrieval or filtering. To identify the relevant information in web search engine, it often returns thousands of pages in response to a broad query and making it difficult for users. Clustering methods are used to group the retrieved documents automatically into a list of meaningful categories. Document clustering is considered as centralized process which it uses descriptors and descriptor extraction [13].Descriptors are sets of words which describe the contents within the cluster.

The goal of a document clustering scheme is to minimizing intra-cluster distances between documents and maximizing inter-cluster distances (using an appropriate distance measure between documents). A distance measure (or, dually, similarity measure) thus lies at the heart of document clustering. The large variety of documents makes it almost impossible to create a general algorithm which can work best in case of all kinds of datasets. The major use of document clustering is to give users an overview of the contents of a document collection and reduce search space. If a collection is well clustered, search only the clusters that will contain relevant documents. Efficiency and effectiveness can be improved by searching through smaller collection itself.

In this paper, we focus on FRECCA Algorithm based document clustering. The rest of the paper is organized as follows. Section II introduces methods and algorithms used for clustering. Section III explains the FRECCA Algorithm. Section IV deals with performance comparison in which FRECCA provides better performance than k-medoid.

II. RELATED WORKS

Various methods and algorithms available for sentence similarity based document clustering are described here.

D. Wang, T. Li, S. Zhu, and C. Ding, (2009) [19] proposed Multi document summarization framework which is based on sentence -level semantic analysis and symmetric non-negative matrix factorization. Based on similarity matrix, symmetric non-negative matrix factorization algorithm is used to divide the sentence into groups for extraction. Within-cluster sentence selection is based on both internal (computed similarity between sentences) and external information (given topic information), otherwise performance will not be improved.

Kamal Sarkar (2009)[14] proposed multi-document text summarization based on the factors such as clustering the sentences, cluster ordering, and selection of representative sentences from clusters. After preprocessing, similarity between sentences is provided by uni-gram Matching-based similarity measure. To make system effective and portable in domain and language, during preprocessing stemming is not applied on input and features such as length, sentence position, and cue phrase are not incorporated.

Y. Li, D. McLean, Z.A. Bandar, J.D. O'Shea, and K. Crockett(2006)[16] presents method for measuring the similarity between sentences or very short text based on semantic and word order information. Semantic similarity is derived from lexical knowledge base and corpus. Word order similarity measures the number of different words as well as word pairs in different order. This method is inefficient and requires human input and is not adaptable to all application domains.

U.V.Luxburg(2007)[19] proposed Spectral Clustering algorithms, are based on matrix decomposition techniques. Data points are mapped onto the space defined by the eigenvectors associated with the top eigen values of the affinity matrix, and clustering is then performed in this transformed space, typically using a k-Means algorithm.

B.J. Frey and D. Dueck(2007)[6] proposed Affinity Propagation, a technique which simultaneously considers all data points as potential centroids (or exemplars). Frey and Dueck have showed how Affinity Propagation can be applied to the problem of extracting representative sentences from text.

R.J. Hathaway, J.W. Deven port, and J.C. Bezdek proposed Relational Fuzzy c-Means (RFCM) algorithm (1989) [10] which is a variant of Fuzzy c-Means, and it operates on relational data input. It requires that the relation expressed by this data be euclidean.R.Mihalcea, C. Corley, and C. Strapparava,(2006)[17] proposed Text Rank, a graph-based ranking model for text processing, and showed how this model can be successfully used in natural language applications.

T.Geweniger, D. Zuhlke, B. Hammer, and T. Villmann (2010) [7] proposed Median clustering methodology for prototype based clustering of similarity/dissimilarity data. In this, the median c-means algorithms with the fuzzy c-means approach are combined, which is only applicable for vectorial (metric) data in its original variant.

P. Corsini, F. Lazzerini, and F. Marcelloni (2005)[18] proposed a new fuzzy relational algorithm(ARCA), based on the popular fuzzy C-means (FCM) algorithm, which does not require any particular restriction on the relation matrix. ARCA partitions the data set minimizing the Euclidean distance between each object belonging to a cluster and the prototype of the cluster. ARCA determines the optimal partition minimizing the objective function.

III. PROPOSED SYSTEM

Sentence clustering plays an important role in many text processing activities. Incorporating sentence clustering into extractive multi document summarization helps avoid problems of content overlap, leading to better coverage [1][4][9][20].By clustering the sentences of those documents, at least one of the clusters to be closely related to the concepts described by the query terms; however, other clusters may contain information pertaining to the query in some way hitherto unknown to us, and in such a case new information is mined successfully. Irrespective of the specific task (e.g., summarization, text mining, etc.), most documents will contain interrelated topics or themes, and many sentences will be related to some degree to a number of these.

In the proposed system, initially preprocessing is done for the document in which stop words and stem words are removed. Stop words are removed by means of comparing with the database that contains stop words. Stem words are removed through Porter stemming Algorithm. Porter's algorithm provides how the words can be reduced to their root words. Once after preprocessing, similarity between sentences is calculated using Text rank measure. Similarity calculation is mainly based on number of terms common between two sentences by number of words present in both sentences. Based on sentence similarity, sentences with highest Page Rank value are taken through Page Rank algorithm [3]. Page Rank algorithm provides the importance of sentence i.e. how many times the sentence appears in the document .Then fuzzy clustering algorithm is applied. Sentence similarity, number of clusters and Page Rank value of each sentence is provided is input to FRECCA Algorithm. FRECCA algorithm operates on relational data. It works on Expectation-Maximization framework [5] in which cluster membership probabilities of each cluster is identified.

FRECCA [2] mainly operates by three steps: random Initialization, Expectation and Maximization step. In Initialization step, cluster membership values are initialized randomly, and normalized such that cluster membership for an object sums to unity over all clusters. Mixing coefficients are initialized such that priors for all clusters are equal. In expectation step, Page Rank value for each object in each cluster is calculated. Page Rank algorithm provides the importance of sentence i.e. how many times the sentence appears in the document. Maximization step involves only the single step of updating the mixing coefficients based on membership values calculated in the Expectation Step.

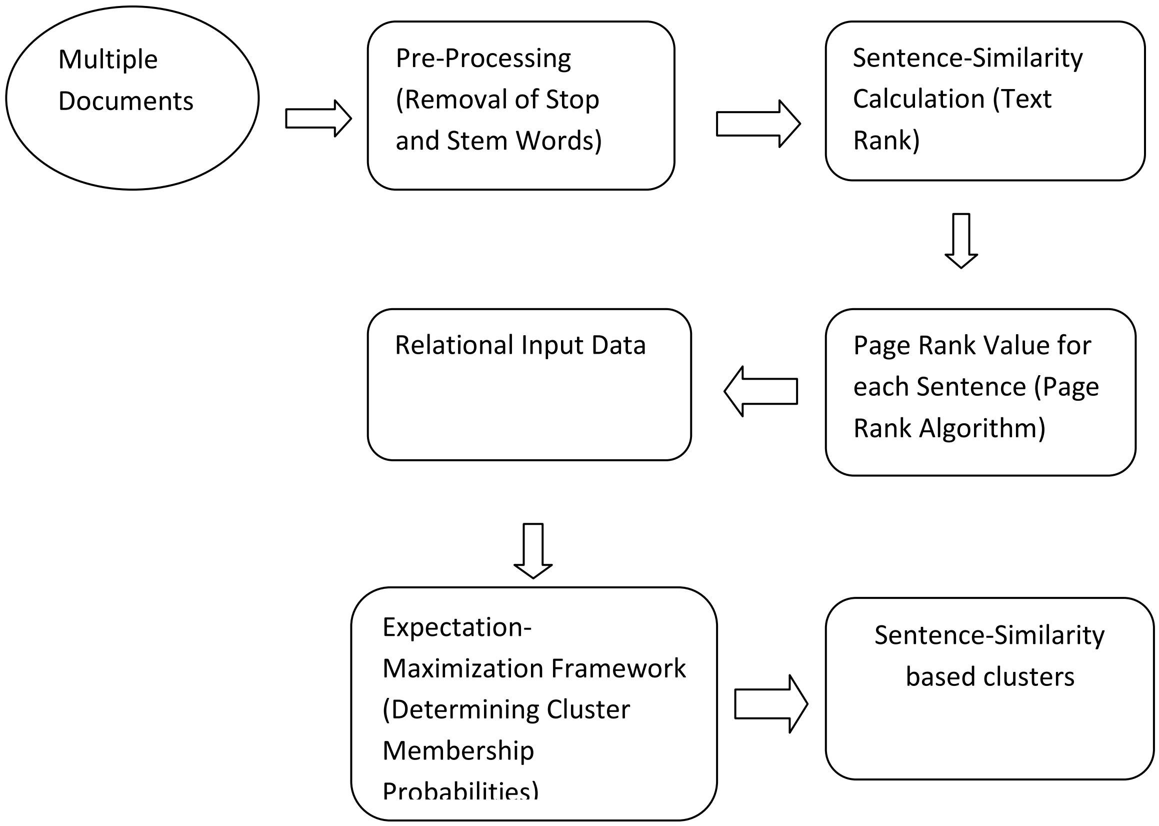
Multiple Documents
Pre-Processing (Removal of Stop and Stem Words)
Sentence-Similarity Calculation (Text Rank)
Page Rank Value for each Sentence (Page Rank Algorithm)
Relational Input Data
Expectation-Maximization Framework (Determining Cluster Membership Probabilities)
Sentence-Similarity based clusters

Fig 1. Architecture of FRECCA Algorithm

A. Preprocessing

Raw data is highly concerned with noise, missing values and inconsistency. The quality of data affects the data mining results. In order to improve the quality of data and consequently of the mining results, raw data is pre-processed so as to improve the efficiency and ease of mining process. In the proposed system, preprocessing for dataset is done to remove the stop words and stem words which are considered as less important and to improve quality and efficiency of data. Many of the most frequently used words in English are useless in Information Retrieval (IR) and text mining. These words are called 'Stop words'. Stop-words, which are language-specific functional words, are frequent words that carry no information (i.e., pronouns, prepositions, conjunctions). Examples of such words include 'the', 'of',' and', 'to', etc. These stop words are get stored in the database. Dataset (famous quotations) is loaded in to another database. Here stop words in data set (famous quotation) is removed by comparing with the stop word database.

Stemming or lemmatization is a technique for the reduction of words into their root. Many words in the English language can be reduced to their root word or base form e.g. agreed, agreeing, and agreement belong to agree. In this Porter Stemming Algorithm is applied for find the root words in the document.

B. Sentence-Similarity Calculation

The ability to accurately judge the similarity between natural language sentences is critical to the performance of several applications such as text mining, question answering, and text summarization. Given two sentences, an effective similarity measure should be able to determine whether the sentences are semantically equivalent or not, taking into account the variability of natural language expression. Similarity between two sentences is provided by Text rank measure.

Similarity (s_i,s_j) = {w_k| w_k€s_i ,w_k€ s_j}/log(|s_i|)+ log(|s_j|)   (1)w_k denotes number of terms common between two sentences (s_i,s_j).
log (|s_i|) denotes number of words in sentence i.
log (|s_j|) denotes number of words in sentence j.

C. Page Rank Computation for Sentence

By means of Page Rank algorithm, Page Rank value for each sentence is calculated. Calculating the importance of a sentence is that sentences which are similar to a large number of other important sentences are central. Thus, by ranking sentences according to their centrality, the top ranking sentences can then be extracted and provided as input to FRECCA algorithm. Page Rank value for each sentence is provided by:

$$\mathrm{PR}_i^m = (1 - \mathrm{d}) + \mathrm{d} \times \sum_{j=1}^{N} \mathrm{w}_{ji}^m \ (\mathrm{PR}_j^m / \sum_{k=1}^{N} \mathrm{w}_{jk}^m) \qquad (2)$$

w_{jk}^mdenotes total similarity value from one sentence to another sentence.
w_{ji}^mdenotes highest similarity value.
PR_j^mdenotes the random value.
d denotes the damping factor to finite the value.

D. FRECCA Algorithm Evaluation

FRECCA [2] works on Expectation-Maximization Framework. In this framework, cluster membership probability for each sentence in each cluster is determined. Number of clusters, Similarity between sentences and Page Rank value for each sentence is provided as input. Cluster Membership probability for sentence in each cluster is provided as output.

The FRECCA algorithm Steps:

Initialization

Assume here that cluster membership values are initialized randomly, and normalized such that cluster membership for an object sums to unity over all clusters. Mixing coefficients are initialized such that priors for all clusters are equal.

Expectation step

The E-step calculates the Page Rank value for each object in each cluster. Page Rank algorithm provides the importance of sentence i.e. how many times the sentence appears in the document.

Maximization step

Since there is no parameterized likelihood function, the maximization step involves only the single step of updating the mixing coefficients based on membership values calculated in the Expectation Step.

E. K-Medoids Algorithm Evaluation

K-medoids [21] is a clustering algorithm related to k-means algorithm and the medoid shift algorithm. Both k-means and k-medoid are partitional (breaking dataset into groups) and both attempts to minimize the distance between points labeled to be in cluster and point designated as the center of that cluster. In contrast to k-means algorithm, k-medoid chooses data points as centers (medoids) and works with an arbitrary matrix of distance between data points. K-medoid is a classical partitioning technique of clustering that clusters the data set of n objects into k clusters. A medoid can be defined as the object of a cluster, whose average dissimilarity to all the objects in the cluster is minimal i.e. it is a most centrally located point in the cluster.

K-medoid algorithm steps:

Initialize: randomly select *k* of the *n* data points as the medoids

Associate each data point to the closest medoid. ("closest" here is defined using any validdistancemetric most commonly Euclideandistance, Manhattandistance or Minkowskidistance)

For each medoid m

For each non-medoid data point o

Swap m and o and compute the total cost of the configuration

Select the configuration with the lowest cost. Repeat steps 2 to 4 until there is no change in the medoid.

Page Rank value of each sentence is provided as input. One or two page Rank value is taken as representative value and based on representative value nearby objects are grouped to form clusters.

IV. EVALUATION CRITERIA

The dataset used here is famous quotation dataset where a large number of documents are available for usage and they are analyzed offline. The four parameters which are used to evaluate the performance for FRECCA Algorithm and K-medoid are Entropy, Purity, Partition_Entropy and V_Measure. Let L= $\{w_{1,}w_{2,} \dots\}$ is the set of clusters, C= $\{c_{1,}c_{2,} \dots\}$ is set of classes and N is number of objects. The purity of cluster is defined as the fraction of the cluster size that the largest class of objects assigned to that cluster represents; thus the purity of cluster j is

$P_j = 1/|w_j|\ (\max\ (|w_j \cap c_i|))$ (3)

Overall purity is defined as the weighted average of the individual cluster purities.

Overall Purity = $1/\mathrm{N} \sum_{j=1}^{|L|}(|\ w_j| \times P_j)$ (4)

The entropy of a cluster j is a measure of how mixed the objects within the cluster are and is defined as

$E_j = -1/\log |C|\ (\sum_{j=1}^{|L|}(|\ w_j\ \cap c_i)/|w_j\ |\ \log\ (|w_j \cap c_i)/|w_j|)$ (5)

Overall entropy is weighted average of the individual cluster entropies:

Overall entropy = $1/\mathrm{N} \sum_{j=1}^{|L|}(|\ w_j| \times E_j)$ (6)

Entropy and purity measure how the classes of objects are distributed within each cluster, they measure homogeneity i.e., the extent to which clusters contain only objects from single class. However, it also measures completeness i.e., the extent to which all objects from a single class are assigned to a single cluster. High purity and low entropy are easy to achieve when the number of clusters is large.

V_Measure is defined as harmonic mean of homogeneity (h) and completeness (c).

$V = hc / (h+c)$ (7)

V_Measure is more reliable than purity and entropy when comparing with clustering with different number of clusters. It takes both homogeneity and completeness in to account.

Partition_Entropy determines how closely particular instance belong to particular cluster. It is defined as

PE = -1/N ($\sum_{i=1}^{N}\sum_{j=1}^{|L|}(u_{ij}log_a u_{ij})$ (8)

Where is the membership of instance i to cluster j. The value of this index ranges from 0 to |L|. The closer is the value is to 0, the crisper the clustering is.

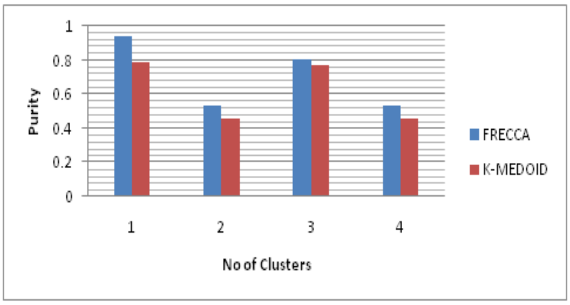

Fig 2.Performance Comparison between FRECCA and k-medoids based on Purity

When number of cluster increases, Purity is less in k-medoid when compared with FRECCA. Based on purity FRECCA achieves 40% whereas k_medoid achieves only 10%.

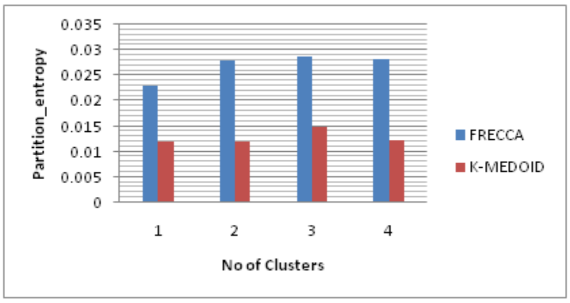

Fig 3.Performance Comparison between FRECCA and k-medoids based on Partition_Entropy

When number of cluster increases, Partition_entropy is less in k-medoid when compared with FRECCA. Partition_Entropy achieved in FRECCA is 10% more than k-medoid.

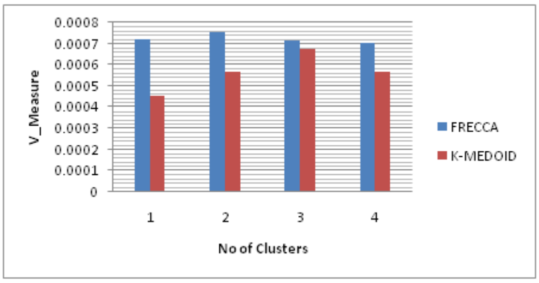

Fig 4.Performance Comparison between FRECCA and k-medoids based on V_Measure

When number of cluster increases, V_Measure is less in k-medoid when compared with FRECCA. V_Measure achieved in FRECCA is 30% more than k-medoid.

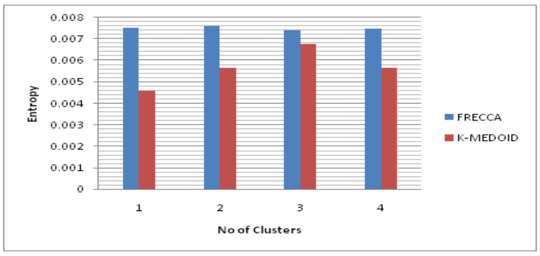

Fig 5.Performance Comparison between FRECCA and k-medoids based on Entropy

When number of cluster increases, Entropy is less in k-medoid when compared with FRECCA. Entropy achieved in FRECCA is 10% more than k-medoid.

V. CONCLUSION AND FUTURE WORK

When compared to the existing work, the proposed work avoids content overlap and able to achieve superior performance to k-medoids algorithms when externally evaluated on a challenging data set of famous quotations. FRECCA is a fuzzy clustering algorithm that can be applied to any relational clustering problem, and its application to several non-sentence data sets has shown its performance to be comparable to k-medoid benchmarks.

Like any clustering algorithm, the performance of FRECCA will ultimately depend on the quality of the input data, and in the case of sentence clustering this performance may be improved through development of better sentence similarity measures, which may in turn be based on improved word sense disambiguation, etc. FRECCA has a number of attractive features. First, based on empirical observations, it is not sensitive to the initialization of cluster membership values, with repeated trials on all data sets converging to exactly the same values, irrespective of initialization. This is in contrast to k-Means and Gaussian mixture approaches, which tend to be highly sensitive to initialization. Secondly, FRECCA algorithm appears to be able to converge to an appropriate number of clusters, even if the number of initial clusters was set very high.The FRECCA algorithm presented in this system identifies only flat clusters. The main future work is to extend these ideas to the development of a hierarchical fuzzy relational clustering algorithm.

VI. REFERENCES

[1] R.M. Aliguyev(2009), “A New Sentence Similarity Measure and Sentence Based Extractive Technique for Automatic Text Summarization,” Expert Systems with Applications, Vol. 36, pp. 7764- 7772

[2] Andrew Skabar and KhaladAbdalgader(2013), “Clustering Sentence-level Text Using a Novel Fuzzy Relational Clustering Algorithm” IEEE Transactions on Knowledge and Data Engineering, Vol 25, No. 1

[3] S. Brin and L.Page(1998), “The Anatomy of a Large-Scale Hypertextual Web Search Engine,” Computer Networks and ISDN Systems, Vol. 30,pp. 107-117,.

[4] P. Corsini, F. Lazzerini, and F. Marcelloni(2005), “A New Fuzzy Relational Clustering Algorithm Based on the Fuzzy C-MeansAlgorithm,” Soft Computing, Vol. 9, pp. 439-447.

[5] A.P. Dempster, N.M. Laird, and D.B. Rubin (1977), “Maximum Likelihood from Incomplete Data via the EM Algorithm,” J. the Royal Statistical Soc. Series B (Methodological), Vol. 39, No. 1, pp. 1-38.

[6] B.J. Frey and D. Dueck(2007), “Clustering by Passing Messages between Data Points,” Science, Vol. 315, pp. 972-976.

[7] T. Geweniger, D. Zu¨ hlke, B. Hammer, and T. Villmann(2010), “Median Fuzzy C-Means for Clustering Dissimilarity Data,” Neurocomputing, Vol. 73, Nos. 7-9, pp. 1109-1116.

[8] T. Geweniger, D. Zu¨ hlke, B. Hammer, and T. Villmann(2009), “Fuzzy Variant of Affinity Propagation in Comparison to Median Fuzzy c-Means,” Proc. Seventh International Workshop Advances in Self-Organizing Maps, pp. 72-79.

[9] V. Hatzivassiloglou, J.L. Klavans, M.L. Holcombe, R. Barzilay,and M. Kan(2001), “SIMFINDER: A Flexible Clustering Tool for Summarization,” Proc. NAACL Workshop Automatic Summarization, pp. 41-49.

[10] R.J. Hathaway and J.C. Bezdek(1989), "Relational Dual of the C-Means Clustering Algorithms," Pattern Recognition, Vol. 22, No. 2, pp. 205-212.

[11] R.J. Hathaway and J.C. Bezdek(1994), "NERF C-Means: Non-Euclidean Relational Fuzzy Clustering," Pattern Recognition, Vol. 27, pp. 429-437.

[12] JesúsAndrés-Ferrer ,GermánSanchis-Trilles,FranciscoCasacuberta "Similarity word-sequence kernels for sentence clustering" Proceeding SSPR&SPR'10 Proceedings of the 2010 joint IAPR international conference on Structural, syntactic, and statistical pattern recognition Pages 610-619 .

[13] X.Ji and W.Xu(2006), Document Clustering with prior knowledge .ACM SIGIR Conference.

[14] Kamal Sarkar(2009), "Sentence Clustering-based Summarization of Multiple Text Documents", TECHNIA – International Journal of Computing Science and Communication Technologies, Vol. 2, No. 1. (ISSN 0974-3375)

[15] U.V. Luxburg(2007), "A Tutorial on Spectral Clustering," Statistics and Computing, Vol. 17, No. 4, pp. 395-416.

[16] Y. Li, D. McLean, Z.A. Bandar, J.D. O'Shea, and K. Crockett (2006), "Sentence Similarity Based on Semantic Nets and Corpus Statistics," IEEE Trans. Knowledge and Data Eng., Vol. 8, No. 8, pp. 1138-1150.

[17] R. Mihalcea, C. Corley, and C. Strapparava(2006), "Corpus-Based and Knowledge-Based Measures of Text Semantic Similarity," Proc.21st Nat'l Conf. Artificial Intelligence, pp. 775-780.

[18] D.R. Radev, H. Jing, M. Stys, and D. Tam (2004), "Centroid-Based Summarization of Multiple Documents," Information Processing and Management: Vol. 40, pp. 919-938.

[19] D. Wang, T. Li, S. Zhu, and C. Ding (2008), "Multi-Document Summarization via Sentence-Level Semantic Analysis and Symmetric Matrix Factorization," Proc. 31st Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval, pp. 307-314.

[20] H. Zha, "Generic Summarization and Keyphrase Extraction Using Mutual Reinforcement Principle and Sentence Clustering (2002)," Proc. 25th Ann. Int'l ACM Conf. Research and Development in Information Retrieval, pp. 113-120.

[21] E.M.Mirkes, "K-means and K-medoids (Applet)", University of Leicester, 2011.