International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:3, No:4, 2009

Clustering Unstructured Text Documents Using Fading Function

Pallav Roxy, and Durga Toshniwal

***Abstract*—**Clustering unstructured text documents is an important issue in data mining community and has a number of applications such as document archive filtering, document organization and topic detection and subject tracing. In the real world, some of the already clustered documents may not be of importance while new documents of more significance may evolve. Most of the work done so far in clustering unstructured text documents overlooks this aspect of clustering. This paper, addresses this issue by using the Fading Function. The unstructured text documents are clustered. And for each cluster a statistics structure called Cluster Profile (CP) is implemented. The cluster profile incorporates the Fading Function. This Fading Function keeps an account of the time-dependent importance of the cluster. The work proposes a novel algorithm Clustering n-ary Merge Algorithm (CnMA) for unstructured text documents, that uses Cluster Profile and Fading Function. Experimental results illustrating the effectiveness of the proposed technique are also included.

***Keywords*—**Clustering, Text Mining, Unstructured Text Documents, Fading Function.

I. Introduction

DATA mining, and in particular text mining, has attracted much attention in recent years due to the vast amounts of data available, and the rate of growth. Data mining tools can be used to uncover patterns or hidden relations in the available data, and can potentially contribute greatly to business strategy decisions, knowledge bases, and scientific and medical research. In contrast to data mining, where one looks for patterns and knowledge in structured databases, text mining deals with unstructured, or semi structured, textual data such as reports, e-mails or web-pages. This project will focus on one aspect of text mining, namely clustering unstructured text document archives.

Clustering unstructured text document is an important issue in data mining community and has a number of applications such as document archive filtering, document organization and topic detection and tracing etc. During the course of time, several old clustered documents may become useless and several new issues might evolve. This problem is not considered in the most prevalent clustering approaches.

The document clustering problem has been well studied and several approaches have been proposed. Approaches can be easily classified into two categories: *similarity-based approaches* and *model-based approaches*. The increasing interest in processing larger collections of documents has led to a new emphasis on designing more efficient and effective techniques, leading to an explosion of diverse approaches to the document clustering problem, including the (multilevel) self-organizing map [1], mixture of Gaussians [2], spherical k-means[3], bi-secting k-means [4], mixture of multinomials [5,6].

Most of the existing work on clustering unstructured text documents does not take into account evolution and fading of relevance of clustered documents during a period of time.

This paper, proposes to use Fading Function for clustering of unstructured text documents. The idea of a Cluster Profile (CP) is used, which is a cluster statistics structure that contains the activity status, fading function and merge factor for the every cluster. The work also proposes a novel algorithm Clustering n-ary Merge Algorithm (CnMA) for clustering unstructured text documents based on CP.

The rest of the paper is organized as follows. In Section II discusses the related work. In Section III presents the proposed algorithm named Clustering n-ary Merge Algorithm (CnMA). Experimental Results are reported in Section IV. The Conclusions are discussed in Section V.

II. Related Work

Document (or text) clustering is a subset of the larger field of data clustering, which borrows concepts from the fields of information retrieval (IR), natural language processing (NLP), and machine learning (ML), among others. The process of document clustering aims to discover natural groupings, and thus present an overview of the classes (topics) in a collection of documents. The problem started with various approaches based on hierarchical agglomerative clustering using a suitable similarity measure such as cosine. The increasing interest in processing larger collections of documents has led to a new emphasis on designing more efficient and effective techniques, leading to an explosion of diverse approaches to the document clustering problem.

These several approaches can be classified into two major categories: *similarity-based approach* and *model-based approach*. In *similarity-based approaches*, one optimizes an objective function involving the pair-wise document similarities, aiming to maximize the average similarities

Pallav Roxy is a post-graduate student at the Department of Electronics and Computer Engineering, IIT Roorkee (e-mail: pallav.roxy@gmail.com).
Durga Toshniwal is Assistant Professor at the Department of Electronics and Computer Engineering, IIT Roorkee (e-mail: durgafec@iitr.ernet.in).

within clusters and minimize the average similarities between clusters. *Model-based approaches*, on the other hand, attempt to learn generative models from the documents, with each model representing one particular document group.

Several approaches have been so far proposed for document clustering since the mid-nineties. The increasing interest in processing larger collections of documents has led to a new emphasis on designing more efficient and effective techniques, leading to an explosion of diverse approaches to the document clustering problem, including the (multilevel) self-organizing map [1], mixture of Gaussians [2], spherical k-means[3], bi-secting k-means [4], mixture of multinomials [5,6].

K-means is one of the simplest unsupervised learning algorithms that solve the well known clustering problem. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters) fixed a priori. The main idea is to define k centroids, one for each cluster. These centroids should be placed in a cunning way because of different location causes different result. So, the better choice is to place them as much as possible far away from each other. The next step is to take each point belonging to a given data set and associate it to the nearest centroid. When no point is pending, the first step is completed and an early grouping is done. At this point we need to re-calculate k new centroids of the clusters resulting from the previous step. After we have these k new centroids, a new binding has to be done between the same data set points and the nearest new centroid. A loop has been generated. As a result of this loop it is seen that the k centroids change their location step by step until no more changes are done. In other words centroids do not move any more. Finally, this algorithm aims at minimizing an *objective function*, in this case a squared error function. The objective function (1)

$$J = \sum_{j-1}^{k} \sum_{i-1}^{n} \| x_i^{(j)} - c_j \|^2 \quad (1)$$

Where, $\| x_i^{(j)} - c_j \|^2$ is a chosen distance measure between a data point $x_i^{(j)}$ and the cluster centre c_j, is an indicator of the distance of the *n* data points from their respective cluster centres. The distance measure commonly used is Euclidean Distance.

k-means is used because of its simplicity and efficiency. The different improvements of k-means such as bisecting k-means and spherical k-means further improve the efficiency of the algorithm. Self-organizing map implements a similarity graph. It consists of finite set of models that approximate the open set of input data. The similarity graph is suitable for interactive mining or exploration tasks. Though, none of these clustering algorithms consider the time dependent importance of the clusters.

However some work done so far in clustering text streams, take into account the time-dependency of the significance of clusters. Various Online Algorithms have been proposed so far for clustering text data streams.

Online algorithms are useful for clustering a stream of documents such as news feeds, as well as for incremental learning situations. In [7,8], based on the traditional *k*-means, S.Guha *et al*. firstly propose the data streams clustering algorithm named STREAM. In STREAM, the centroid is used to represent the clusters, and a layered clustering strategy is used to enhance the algorithm efficiency. In [9], Aggrawal *et al.* propose a novel data stream clustering framework called as CluStream, which views the data streams clustering as an evolutionary process with time. This framework includes two sub-processes, that is, the online process and offline process. In [10] Aggarwal *et al*. introduce the concept of cluster droplet in order to store the real-time condensed cluster statistics information. When a document comes, it would be assigned to the suitable cluster and then the corresponding cluster droplet is updated. It also employs a concept of Fading Function to take into account the property of topics to grow old and go out of discussion.

In [11] Liu *et al.* incorporate properties of semantic smoothing model and fading functions to propose Online Clustering Algorithms OCTS and OCTSM to cluster text data streams. In OCTSM, the inactive clusters are given one chance before they are deleted. If any inactive cluster is found similar to an active cluster, it is merged to the active cluster. If the new document is not similar to any of the existing clusters, then the most inactive cluster is deleted and a new cluster is formed containing the new document. They employ a special cluster statistics structure referred to as Cluster Profile.

The proposed work intends to combine the ideas of a cluster statistics structure called Cluster Profile [15], Fading Function [15] and Merge Algorithm [18] for clustering unstructured text documents.

Apart from these, some more concepts, which will be helpful to understand the functioning and evaluation of the proposed algorithm are discussed below.

Entropy of a cluster is defined as the degree of disorder of the cluster. The more number of alike elements are in the cluster, the less the entropy is. It can be mathematically calculated as in (2),

$$H(\Omega) = -\sum_{k} \frac{|w_k|}{N} \log \frac{|w_k|}{N} \quad (2)$$

Where, $\Omega = \{w_1, w_2, \ldots, w_k\}$ is the set of clusters where w_i is a cluster and N is the total number of data points.

Normalized Mutual Information (NMI) is a measure for evaluating clustering quality. Higher the value of NMI, better is the clustering quality. For a set of clusters $\Omega = \{w_1, w_2, \ldots, w_k\}$ and a set of classes $C = \{c_1, c_2, \ldots, c_j\}$ where w_k is a cluster and c_j is a class, NMI can be calculated as:

$$NMI(\Omega, C) = \sum_{k} \sum_{j} \frac{I(\Omega, C)}{[H(\Omega) + H(C)]/2} \quad (3)$$

where, $I(\Omega,C)$ is the mutual information given by:

$$I(\Omega, C) = \sum_{k} \sum_{j} \frac{|w_k \cap c_j|}{N} \log \frac{N |w_k \cap c_j|}{|w_k||c_j|} \quad (4)$$

After reviewing the related work and basic concepts let us now proceed to the proposed algorithm.

III. Proposed Clustering Algorithm

The proposed work clusters unstructured text documents and is called Clustering n-ary Merge Algorithm (CnMA). In Section A briefly explains the basic concepts used in the proposed algorithm which is described in Section B.

A. Basic Concepts

The key issue in the clustering of text documents is the time-dependent importance of the topics of clusters. There may be some topics which are no longer discussed or are passive for us in the current context. And there may be some other topics which are discussed or are accessed time and again. In order to account for this time-dependent importance of clusters, we use a time sensitive weight for each cluster. It is assumed each cluster has a time-dependent weight defined by the function *f(t)*. The function *f(t)* is also referred to as the Fading Function (FF). The Fading Function *f(t)* is monotonic decreasing function which decays uniformly with time *t*. The value of FF keeps on decreasing at a certain rate, irrespective of the fact that a document is added to any given cluster or not. This rate is controlled by an attribute called Half Life (HL).

1) Fading Function (FF)

Consider the time point t, the fading function value is defined as $f(t) = 2^{-\zeta t}$, where $\zeta=1/t_0$ and t_0 is the Half Life.

2) Half Life (HL)

The half life t_o of a cluster is defined as the time at which, $f(t_0) = (1/2)f(0)$. The aim of defining a half life is to define the rate of decay of the weight associated with each cluster. The decay-rate is defined as the inverse of the half life. Similar to [18], we denote the decay rate by $\zeta=1/t_0$ and the fading function is defined as follows.

For the clustering purposes, we use a combined structure called Cluster Profile which integrates the time-dependent weights to the clusters. Another feature captured in a Cluster Profile is the Activity Status of the cluster recording whether the cluster was active or not in the past few loops. Also, we introduce a feature called Merge Factor which presents the zone in which the cluster falls based on its activity.

The basis of the cluster profile is a cluster which contains information such as the data points in the cluster, the centroid of the cluster and an identification number or ID.

3) Cluster Profile (CP)

A cluster profile *CP(t,C)* at time t for a document cluster *C* is defined as a pentuple (*id,C,f(t),as,mf*), where

- *id* is the number identifying the Cluster Profile.
- *C* is the cluster. It contains all the information related to the cluster like the data points, centroid etc.
- *f(t)* is the fading function related to cluster *C*.
- *as* is a boolean array that captures the activity status of the cluster during the past few document additions.
- *mf* is the merge factor positioning the cluster in a region based on the value of FF before running the merging algorithm.

The properties of Cluster Profile [18] regarding updatability, additivity and fading which have been used are described as follows:

Additivity: Additivity describes the variation of cluster profile after two clusters C1 and C2 are merged as C1∪C2. Suppose two cluster profiles are CP(t, C1) = (id1, C1, f1(t), as1, mf1) and CP(t, C2) = (id2, C2, f2(t), as2, mf2). Then CP(t, C1∪C2,) can be defined as (id1, C1∪C2,, f1(t), as1, mf1), where id1 has been taken because f1(t)>f2(t).

Updatability: Updatability describes the changed Cluster Profile after a new document has been added into the cluster having original CP(t,C) as (id, C, f(t), as, mf). Let us assume a document d is added to the cluster C, then the updated CP(t, C) becomes (id, C, 0.9, as, mf). If before addition as was 11000 then after addition its value will be 11001. The least significant bit signifies the latest activity. Here the 4 most significant bits are obtained by Left Shift of the original as and the least significant bit is a newly added bit, 0 signifies no addition and 1 signifies addition.

Fading: Fading Property describes the variation of cluster profile with time. Consider the cluster profile at the time t1 is CP(t1, C) = (id, C, f(t1), as, mf) and no document has been added to the cluster C during [t1; t2]. Then the cluster profile at time t2 is defined as CP(t2, C) = (id, C, f(t2), as, mf), where $f(t2) = 2^{-\zeta(t2-t1)}$

B. Clustering n-ary Merge Algorithm (CnMA)

The proposed clustering algorithm CnMA includes three phases:

1. Text Preprocessing (Fig. 1),
2. Initialization process (Fig. 2), and
3. Clustering process (Fig. 3).

In Phase I, the text documents are processed and formed into vectors of terms defined by a dictionary. From time to time this dictionary is updated in order to include the evolving terms. After this Phase comes the initial clustering phase, i.e. Phase II.

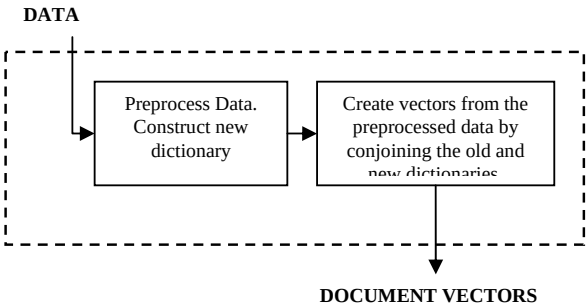

Fig. 1 Block diagram for Phase I

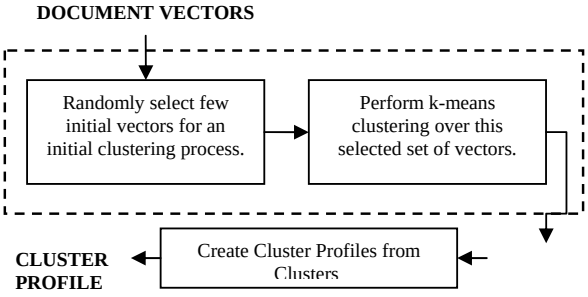

Fig. 2 Block diagram for Phase II

Phase II refers to the formation of the initial k clusters. This is done by k-means clustering. Few documents are chosen randomly and clustered with k-means. Cluster Profiles are then created for each of these clusters. The block diagram for this phase is given in Fig. 2.

Our main focus is on the Phase III where we propose the clustering algorithm with n-ary merge routine. After constructing the *k* basic cluster and their respective cluster profiles, we revisit the set of clusters after every time-stamp. There may or may not be any new documents arriving between two time-stamps. If there is no new document then the fading functions of all the clusters are changed as per the Fading property of CP.

Also, at every cycle we perform a merge routine to check whether any old inactive cluster is similar to any currently active cluster. The similarity measure used is the Euclidean Distance. The lower the value of Euclidean Distance, more similar the clusters are. In case there is such a pair of clusters, then they will be merged.

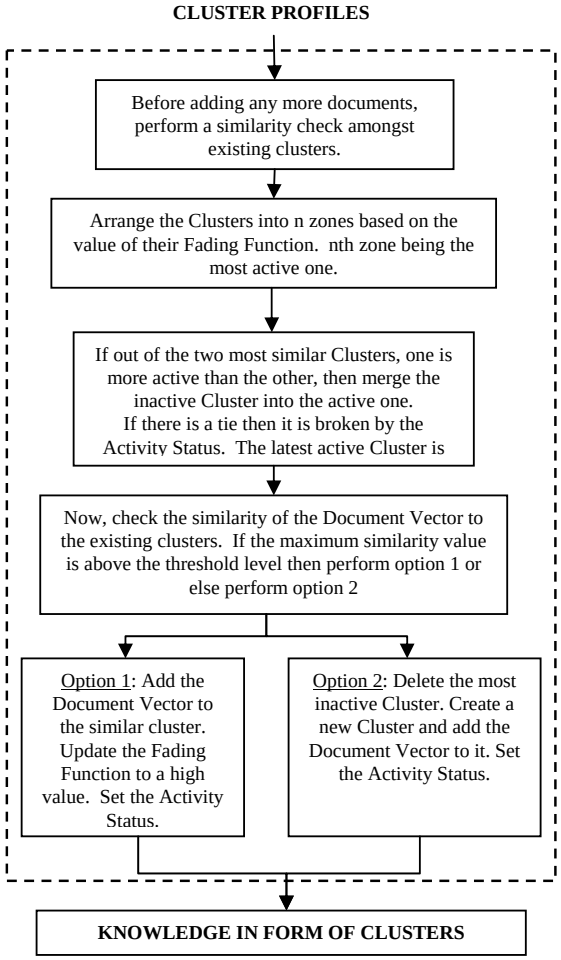

Fig. 3 Block diagram for Phase III

The merge algorithm divides the *k* clusters into *n* disjoint sets based on the value of the fading function of the cluster. The value of $n \neq k$, thus each set will contain zero or more clusters. Generally, the value of *n* will be less than *k.* The set with the most active clusters, i.e. the clusters having greater value of FF will be assigned the Merge Factor *n* and will fall in the *nth* set while the least active clusters will have 1 as their Merge Factor and will be located in the first set. To understand the merge algorithm in a better way, let us take an example where we have 6 clusters and n = 3. The illustration is given in Fig. 4.

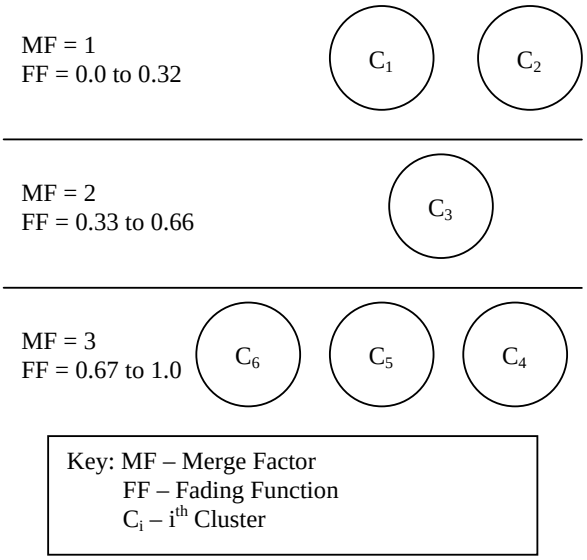

Fig. 4 Illustration for Merge

TABLE I
VERDICT FOR MERGING CLUSTERS

MF = 1 and FF = 0.0 to 0.32

Case	Activity Status	Proximity Value	Verdict
4a	Different	Different	No change.
4b	Different	Similar	Merge cluster which most recently active.
4c	Similar	Different	Merge cluster with lowest proximity value
4d	Similar	Similar	Randomly chose a cluster to merge.

In this example, the value of Fading Function for clusters C1 and C2 lies from 0.0 to 0.32 indicating that they have not been updated over a long period of time. We thus assign value 1 to the Merge Factors of clusters C1 and C2. Value of Fading Function for cluster C3 lies from 0.33 to 0.66 and thus it can be called a medium active cluster with Merge Factor as 2. Clusters C4, C5 and C6 have value of Fading Function lying from 0.67 to 1.0, thus highlighting their recent activity.

Following scenarios will occur, when we run Merge Algorithm:

Case 1: Cluster C1 is similar to cluster C4 and the proximity value is lowest, and no other cluster pair achieves this condition.

Verdict: Cluster C1 is merged with cluster C4.

Case 2: Cluster C3 is similar to cluster C5 and the proximity value is lowest, and no other cluster pair achieves this condition.

TABLE II
DESCRIPTION OF CNMA

S. No.	Steps
	Inputs: Unstructured text document archive *D*, *k* is the number of clusters, ε is the cluster radius, *n* is the number of activity zones.
1.	Take some initial documents and run k-means clustering to create *k* initial clusters based on the term frequency model for a given dictionary of terms.
2.	**While**(the archive is not empty) **do**
3.	**Begin**
4.	*t* = Get Current Time Stamp
5.	Update all cluster profiles with *t*
6.	*Merge(n)*
7.	**For** each cluster *i* **do**
8.	Calculate Euclidean distance of *d* with cluster *i* and store the min. distance.
9.	**If** the minimum distance is more than the ε **then do**
10.	**Begin**
11.	Delete the most inactive cluster C_{NID}
12.	Create a new cluster with *ID* = *NID* and build its cluster profile
13.	Assign document *d* to the new cluster profile
14.	**End**
15.	**Else If** there are two clusters which have the minimum distance **then do**
16.	Check which cluster was updated the latest.
17.	Add the document to the last updated cluster.
18.	**End**

TABLE III
DESCRIPTION OF *MERGE(N)*

S. No.	Steps
1.	Divide the k - cluster profiles into n zones based on the value of their fading function. 1 being the most inactive and n being the most active zone.
2.	**For** every cluster *i* and *j* where $i \neq j$ **do**
3.	Calculate Euclidean Distance between clusters *i* and *j*.
4.	Find the one pair with the minimum value.
5.	**If** there are more than one pair **then do**
6.	Find the pair where the inactive cluster was not updated for the longest time.
7.	**If** for some *i* and *j*, the distance is minimum and less than ε **then do**
8.	**If** either of the clusters falls in nth zone then merge the clusters.
9.	**Else if** the difference between the zones is more than 1 **then** merge them.
10.	**Else** take no action.

Verdict: Cluster C3 is merged with cluster C5.

Case 3: Cluster C2 is similar to cluster C3 and the proximity value is lowest, and no other cluster pair achieves this condition.

Verdict: The activity does not vary drastically between

clusters with Merge Factor 1 and 2, thus if a cluster with MF = 1 is similar to a cluster with MF = 2, they will not be considered for merging. As per this heuristics, cluster C2 and C3 will not be merged. Generalizing this verdict for any value of *n*, we would say that the clusters will be merged only if the difference in the values of their Merge Factors is greater than or equal to 2.

Case 4: Clusters C1 and C2, both are similar to cluster C5. In this case Table I describes the verdict taken, based on proximity value and Activity Status.

If a document arrives in a time-stamp, we check the Euclidean Distance of the document with the centroids of all the clusters. If there exists a cluster whose centroid's Euclidean Distance from the document is lowest and below a threshold value ε then the document is added to the cluster and the CP of cluster is updated using the Updatability Property. But if no such cluster exists, then the most inactive cluster will be deleted and a new cluster containing the document is added to the set of clusters and its cluster profile is created.

The values of *k* (number of clusters), n (number of disjoint sets), ε (cluster radius), and ζ (decay constant for FF) are values which will remain constant and should be consulted with the domain expert. Table II and III gives a step-by-step procedure for the CnMA and Merge algorithm.

The Phase II i.e. the initialization process corresponds to lines 1 of Table II. In detail, CnMA takes some initial documents and cluster them using simple k-means clustering algorithm and then compute there respective cluster profiles. The Phase III, the actual clustering process corresponds to lines 2~18. In this process, as a new text document arrives, firstly FF of all Cluster Profiles will be updated using Fading property. Next the Merge algorithm would be executed on all the cluster profiles. The Merge algorithm is given in Table 3. After executing Merge, the Euclidean distance of the new document from each present cluster is calculated. The document is added to the nearest cluster with distance below the threshold value. If no such cluster exists, then the most inactive cluster is deleted and the new cluster containing the new document is added to the groups.

IV. Experimental results

The proposed algorithm was evaluated on a real data set consisting of 150 text documents containing abstracts of grants approved by NSF during 2000-2008[12]. The documents were related to 7 different areas. We have used a dictionary comprising of 16 words. The number of documents in each of the 7 areas is listed in Table IV.

Out of the 150 text documents, we assume that the initial document archive consists of 50 text documents and the rest of the 100 text documents are added at various intervals of time. To have a better analysis of the algorithm, we have permuted these 100 documents into 3 possible ways and then clustered each set individually. Table V illustrates the initial clusters formed after clustering 50 documents using k-means clustering.

The values of the constants were taken as follows:

1. Value of *k:* A graph of entropy (2) vs *k* is shown in Fig. 5. It can be seen that the objective function in (1)
2. reaches a stable value at $k = 6$.
3. Value of *n:* For simplicity and convenience, we choose n as 3 in order to place the k clusters in less *active*, medium *active* and highly *active* zones, where by "*active*" means significant in the current time context. The value of *n* may be chosen depending on the application.
4. Value of ε: Value of ε has been chosen as 0.04. The reason for this being that this value for cluster radius depicts the natural groupings in the dataset quite well.

The final cluster output for the three permutations of 100 documents is given in Table VI, VII and VIII.

On comparing Table IV with Tables VI, VII and VIII, we observe that the number of documents in each of the area cluster roughly remains the same. Further, if we compare Table V with Table VI, the area cluster for "Image Processing" has faded away while the area cluster for "Parallel Algorithm" has evolved.

TABLE IV
Number of documents in the Data Set

S. No.	Area Cluster	No. of Documents
1.	Gene Database	24
2.	RFID Sensor Device	24
3.	Database Management System	24
4.	Wireless Sensor Network	24
5.	Data Mining Pattern	24
6.	Image Processing	14
7.	Parallel Algorithm	16

TABLE V
Initial Cluster Formation

S. No.	Area Cluster	No. of documents in the cluster
1.	Gene Database	9
2.	RFID Sensor Device	8
3.	Database Management System	8
4.	Wireless Sensor Network	9
5.	Data Mining Pattern	8
6.	Image Processing	8
	TOTAL:	50

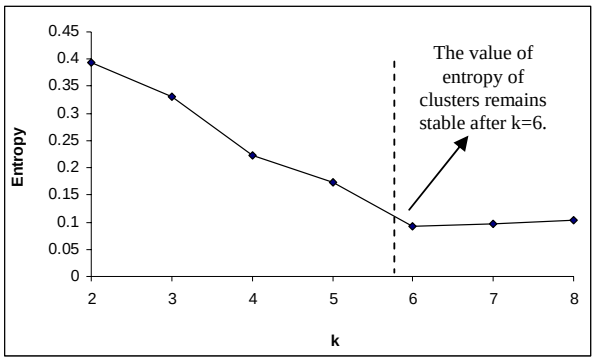

Fig. 5 Entropy vs. k

TABLE VI
CLUSTER OUTPUT FOR PERMUTATION 1

S. No.	Area Cluster	No. of Documents in the cluster
1.	Gene Database	23
2.	RFID Sensor Device	21
3.	Database Management System	21
4.	Wireless Sensor Network	22
5.	Data Mining Pattern	21
6.	Parallel Algorithm	16
	TOTAL:	124

TABLE VII
CLUSTER OUTPUT FOR PERMUTATION 2

S. No.	Area Cluster	No. of Documents in the cluster
1.	Data Mining Pattern	21
2.	Gene Database	23
3.	Database Management System	22
4.	RFID Sensor Device	16
5.	Wireless Sensor Network	20
6.	Image Processing	14
	TOTAL:	116

TABLE VIII
CLUSTER OUTPUT FOR PERMUTATION 3

S. No.	Area Cluster	No. of Documents in the cluster
1.	Gene Database	21
2.	Database Management System	23
3.	Wireless Sensor Network	22
4.	Parallel Algorithm	16
5.	Data Mining Pattern	23
6.	Image Processing	14
	TOTAL:	119

This may be due to the fact that the first permuted list of 100 text documents contains all the “Image Processing” text documents at earlier periods of time and “Parallel Algorithm” text documents evolved at a later stage. Similarly we can see that in Table VII the area cluster “Parallel Algorithm” has faded indicating that this area has not been lately active as per the Permutation 2. The less number of documents for area cluster “RFID Sensor Device” in Table VII indicates that this cluster was deleted and then evolved at later time.

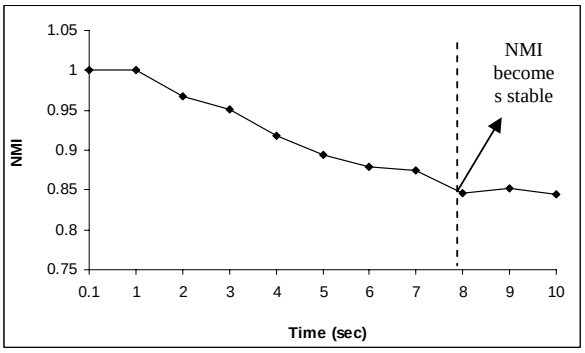

Fig. 6 NMI at every second for Permutation 1

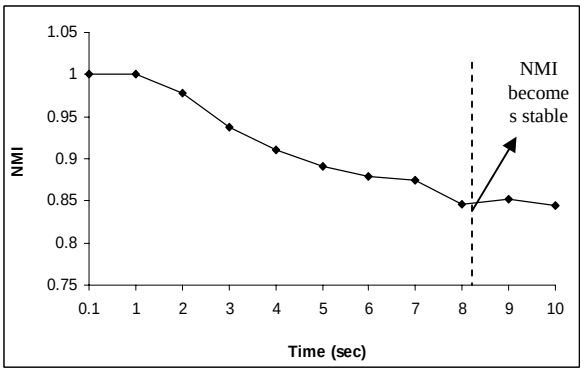

Fig. 7 NMI at every second for Permutation 2

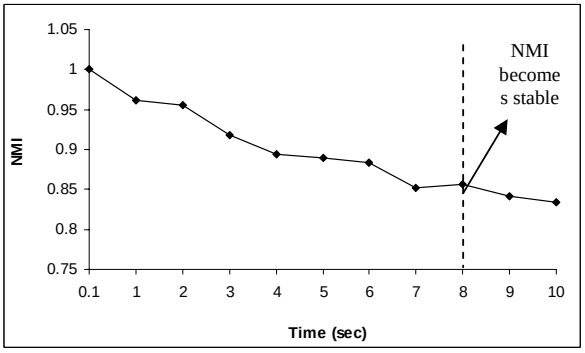

Fig. 8 NMI at every second for Permutation 3

The clustering quality was determined with the help of NMI evaluation function (3). Fig. 6, 7 and 8 present the NMI values for the three permutations. We started at NMI = 1 and a database of 50 text documents placed in their respective clusters. When we started adding new documents, the NMI reduced slightly. Finally when we had completed the addition of 100 new documents to the existing 50 documents, the NMI reduced slightly but stabilized at 0.8 which is an acceptable value.

Thus with the help of these results we can assert that the proposed algorithm is an effective technique for clustering unstructured text documents.

V. Conclusion

The proposed novel algorithm CnMA successfully clusters unstructured text documents keeping an account of the evolution of new topics during a time span.

The Euclidean Distance has been used as the proximity measure for k-means clustering. The description of a cluster is maintained in form of a cluster statistics structure called Cluster Profile. It consists of the Cluster Id, Cluster Identifier, Activity Status, Merge Factor and Fading Function. It also incorporates the concept of time-dependent significance of text documents using a Fading Function on clusters. Activity Status maintains the past boolean log of recent activity of the cluster. Any cluster with no/very few incoming new documents will be faded i.e. becomes less relevant over a period of time.

The Merge Algorithm takes care of the condition when an old inactive cluster may be similar and is in context with an active cluster. The old and less recently active cluster will be merged with the active cluster to retain the old documents for further use based on the values of Merge Factor, Fading Function and Activity Status.

In future, work may be extended on the following lines. The initial clustering done by k-means can be replaced by other appropriate clustering algorithms. The Fading Function used can be chosen from a variety of monotonically decreasing functions. The function being used can be chosen depending on the domain of application. The algorithm can be further analyzed by varying the merge rate.

References

[1] T. Kohonen, S. Kaski, K. Lagus, J. Salojrvi, J. Honkela, V. Paatero, A. Saarela, "Self organization of a massive document collection", *IEEE Trans. Neural Networks*, vol. 11, 2000, pp. 574-585.

[2] J. Tantrum, A. Murua, W. Stuetzle, "Hierarchical model-based clustering of large datasets through fractionation and refractionation", *Proc. 8th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining,* 2002, pp. 183-190.

[3] I. S. Dhillon, D. S. Modha, "Concept decompositions for large sparse text data using clustering", *Machine Learning*, vol. 42, 2001, pp. 143-175.

[4] M. Steinbach, G. Karypis, V. Kumar, "A comparison of document clustering techniques", *KDD Workshop on Text Mining*, 2000, pp. 109-110.

[5] S. Vaithyanathan, B. Dom, "Model-based hierarchical clustering", *Proc. 16th Conf. Uncertainty in Artificial Intelligence*, 2000, pp. 599–608.

[6] M. Meila, D. Heckerman, "An experimental comparison of model-based clustering methods", *Machine Learning*, vol. *42*, 2001, pp. 9–29.

[7] L. O'Callaghan, N. Mishra, A. Meyerson, S. Guha, "Streaming data algorithms for high-quality clustering", In *Proc. ICDE, San Jose, CA*, February 2002, pp. 685-704.

[8] S. Guha, N. Mishra, R. Motwani, L. O'Callaghan, "Clustering data streams", In *Proc. FOCS, California*, November 2000, pp. 359-366.

[9] C. C. Agrawal, J. Han, J. Wang, P. S. Yu, "A framework for clustering evolving data streams", In *Proc. VLDB, Berlin*, September 2003, pp. 81-92.

[10] C. C. Aggarwal, P. S. Yu , "A framework for clustering massive text and categorical data streams", In *Proc. SIAM Conference on Data Mining*, Bethesda, MD, April 2006, pp. 407-411.

[11] Y. B. Liu, J. R. Cai, J Yin., "Clustering text data streams", *Journal of Computer Science and Technology,* vol. 23(1), Jan. 2008, pp. 112-128.

[12] http://www.nsf.gov/awardsearch