IMPROVED TEXT CLUSTERING WITH NEIGHBORS

Sri Lalitha Y[1] and Govardhan A[2]

[1]Department of Computer Engineering,
Gokaraju Rangaraju Institute of Engineering and Technology, Hyderabad, Telangana
[2]Professor of CSE & Director School of Infromation Technology

ABSTRACT

With ever increasing number of documents on web and other repositories, the task of organizing and categorizing these documents to the diverse need of the user by manual means is a complicated job, hence a machine learning technique named clustering is very useful. Text documents are clustered by pair wise similarity of documents with similarity measures like Cosine, Jaccard or Pearson. Best clustering results are seen when overlapping of terms in documents is less, that is, when clusters are distinguishable. Hence for this problem, to find document similarity we apply link and neighbor introduced in ROCK. Link specifies number of shared neighbors of a pair of documents. Significantly similar documents are called as neighbors. This work applies links and neighbors to Bisecting K-means clustering in identifying seed documents in the dataset, as a heuristic measure in choosing a cluster to be partitioned and as a means to find the number of partitions possible in the dataset. Our experiments on real-time datasets showed a significant improvement in terms of accuracy with minimum time.

KEYWORDS

Similarity Measures, Coherent Clustering, Bisecting kmeans, Neighbors.

1. INTRODUCTION

Advanced technologies to store large volumes of text, on the internet and on a variety of storage devices has made text documents to be available to the users all over the world with a mouse click. The job of arranging this continuously growing collection of text documents for diverse requirement of the end user is a tedious and complicated task. Hence, machine learning techniques to organize the data for quick access is essential. In the literature, there are two main machine learning techniques proposed namely classification and clustering. Assignment of an unknown text document to a pre-defined category is called Classification. Assigning an unknown text document by identifying the document properties is called Clustering. Clustering a widely used technique in the fields of Pattern Recognition, Mining, Data from Databases, Extracting relevant information in Information Retrieval Systems and Mining Text Data from Text documents or on Web.

The Paper deals with background on text clustering in section 2, section 3 details document representation and the similarity measures, section 4 deals with our proposed approach to select seed documents, Section 5 presents an approach to find the number of clusters then Section 6 presents proposed clustering approach, Section 7 deals with experiment setup and Analysis and Section 8 concludes.

DOI : 10.5121/ijdkp.2015.5203

2. BACKGROUND

Document clustering is broadly categorized as Hierarchical and Partitional techniques. Hierarchical clustering is of two types agglomerative type and divisive type clustering. *Agglomerative:* Proceeds with one text document in one cluster and in every iteration, it combines the clusters with high similarity. *Divisive:* Proceeds with Single set, initially containing whole text documents, in each iteration it partitions the cluster into two and repeats the process until each cluster contains one document. Hierarchical clustering produces nested partitions [1], with a document dataset at the beginning of hierarchy called the root, and single document partitions at the end of hierarchy called the leafs. Each intermediate hierarchy called non-leaf partition is treated as merging of two partitions from the immediate lower hierarchy or partitioning an immediate higher hierarchy into two sub-hierarchies. The results of this type of clustering is graphically presented in a tree like structure, called the dendogram. The merging or splitting process of text documents presented in a dendogram provides an understanding of text document categorization and the relationships between different clusters.

Partitional clustering form flat partitions or in other words single level partitions of documents and are applicable in the datasets where inherent hierarchy is not needed. If number of clusters to form are K, partitional approach finds all the required partitions (*K)* at a time. In contrast hierarchal clustering, in each iteration splits or merges partitions based on divisive or agglomerative clustering type chosen. Using hierarchical clustering we can form flat sets of K partitions that is deriving partitional clustering using dendogram, and similarly hierarchical clustering is derived from repeated application of Partitional clustering. It is known that document clustering suffers from curse of high dimensionality and partitional clustering is best suitable technique in high dimensional data, hence variant of K-means are widely applied to Document clustering. K-means uses centroid to partition documents, centroid is a representative document of a cluster which is a mean or median of a set of documents.

Hierarchical clustering is a better clustering technique in terms of cluster quality, but is limited by its quadratic time complexity and the threshold value, where as linear time to the number of documents is achieved with different K-mean techniques. Bisecting k-means provides better performance than K-Means. It is better as it produces equal sized clusters rather than largely varying size clusters [2,10]. It starts with whole dataset and with each iteration splits partitions into two sub-partitions.

The proposed work is based on neighbors, hence a brief discussion of clustering algorithms [11, 12, 13, 14]using neighbors and link to cluster documents. The clustering algorithm proposed in [11], introduced the idea of snn (shared nearest neighbor), where, using a similarity matrix a graph is built. For a pair of points p1, p2,it determines k-near neighbors. It assigns p1, p2 to same cluster when there is a link between p1, p2 and share a set of minimum neighbors. Link between p1, p2 is established when p1, p2 are in the near neighbor list of each other.

Clustering algorithm in [12] is an extension of [11]where individual lists of nearest neighbor are ranked based on the similarity values of the data points. These ranks are added to determine the mutual neighborhood. The data points with highest mutual neighborhood are placed in same cluster using agglomerative clustering.

[13] Uses DBSCAN a density based approach to clustering , with near neighbor information. The algorithm looks for density reachable from a user specified radius. P2 is “density reachable”top1, if p1 is within the radius of p2 and radius of p2 contains minimum number of points. Then p1 and p2 are placed in the same cluster.

[14], A Link is established between P1,P2 points, when P1,P2 are in each other near neighbors list. The number of shared neighbors determines the strength of a link and a link is called strong if strength is greater than a specified value. To cluster the data points they have used strength and strong links.

In [15], Weighted Shared nearest neighbors Graph (*WSnnG*) is constructed, integrating different clustering's into a combined one also called cluster ensembles, to reveal the structure of data in clusters formed from different clustering methods. In this method to reduce the graph size a pre-defined number of neighbors are considered and built a graph. This graph is then used with redefined weights on vertices and edges of graph for clustering.

For clustering to happen, a clear measure of closeness of the pair of objects is essential. A number of similarity or distance measures were proposed in past but, their efficiency to text clustering is not so clear. In [5] effectiveness of a number of similarity measures on web pages is studied, our selection of similarity measure is based on [5],[8].

3. DOCUMENT REPRESENTATION

A vector space model (VSM) representation called bag of words is a simplest and widely used document representation. A vector 'd' is set of document terms(unique terms). In VSM the columns represent terms and row indicates document. Each row of a vector is filled with its term frequency (TF). Hence d_{tf} is given by

$$d_{tf} = (tf_1, tf_2, tf_3, ...tf_D)$$

Where tf_i is count of occurrences of term *i* in *d*. *Inverse document frequency* (IDF) is the ratio of total documents(*N*) to the occurrence of term i in documents(df_i). IDF values are low for high frequent terms in dataset and high for less frequent terms in dataset. Log due to large dataset. Thus resulting definition of IDF is

$$IDF_i = \log\left[\frac{N}{df_i}\right] \qquad .....1$$

IDF with TF is known as tf-idf weight.

$$W_{i,j} = tf_{i,j} X\, idf_i \qquad .....2$$

tf–idf of the document d is :

$$d_{tf\text{-}idf} = \left[tf_1 \log(N/df_1), tf_2 \log(N/df_2), .., tf_D \log(N/df_D)\right] \qquad .....3$$

The centroid c_p of a cluster $Clus_p$ is given by

$$c_p = \frac{1}{|C_p|} \sum_{doc_i \in clus_p} doc_i .....4$$

Where $|Clus_p|$ is the size of cluster $Clus_p$and doc_i is a document of $Clus_p$.

3.1 Similarity measures

Good clustering depends on good similarity measure between a pair of points [4]. A variety of measures Cosine, Jaccard, Pearson Correlation and Euclidean distance were proposed and widely applied to text documents.

Links and Neighbors : Two documents are considered to be neighbors if they are similar to each other [6] and the link between the documents represent the number of their common neighbors. Let sim(doc_u,doc_v) calculates pair-wise document similarity and ranges in [0, 1], value one indicates docu, docv are alike and zero indicates that documents doc_u, doc_v are different. If *Sim(*doc_u, doc_v*)* $\geq \Theta$ *with* Θ *value between 0 and 1* then doc_u, doc_v are neighbors, where $\boldsymbol{\theta}$ is specified by the user to indicate the similarity among documents to be neighbors. When Θ is set to 1, then it is a neighbor of another exactly same document and if $\boldsymbol{\theta}$ is set to zero then any document can be its neighbor. Hence $\boldsymbol{\theta}$ value should be set carefully. In this work after performing many experiments with different datasets we have arrived to a conclusion to set automatic value for $\boldsymbol{\theta}$. For a chosen similarity value x, count of entries ≥x in similarity matrix is 2 times N where N is the size of dataset then similarity value for $\boldsymbol{\theta}$ can be set as x.

Neighbor of every document is represented in a matrix called as neighbor matrix. Let NM be an *n x n* matrix of neighbors with *n* being the dataset size and based on doc_u, doc_v being neighbors NM[u,v] is set to 1 or 0 [7]. Let N[doc_u] gives the count of neighbors of doc_u obtained from NM with u^{th} row entries as one.

The links(doc_u, doc_v) is used to find the count of shared neighbors of docu, docv[6] and is calculated as a product of u^{th} row, v^{th} column of NM.

$$link(doc_u, doc_v) = \sum_{m=1}^{n} NM[u,m] \times NM[m,v] \qquad .....5$$

Thus, large value of link(doc_u, doc_v) has high possibility of these documents assigned to one cluster. Since the measures [Cosine/Jaccard/Pearson] measure pair wise similarity between two documents, these measures alone will lead to general or narrow clustering while using link function with these measures can be considered as a specific or comprehensive clustering approach [6], as neighbor data in similarity adds global view to determine documents similarity.

3.2 Similarity Measure with Link

In [8] various similarity measures for text clustering are described. The link function based on neighbors determines similarity of documents. Let a group of documents considered as neighbors of document doc_u has a set of common terms with document doc_u and let document doc_v has another set of terms common with many documents of doc_u, we can consider doc_u and doc_v as similar base on the number of common neighbors doc_u, doc_v share, even though when doc_u, doc_v are not similar by pair wise similarity. Based on these discussions in [7], the authors propose a new similarity measure making use of cosine and link functions. In [9] we extended and experimented with jaccard and pearson measures on k-means and noticed that cosine and jaccard performed better than pearson on neighbor information. The new similarity measures are as follows

$$f(doc_u, cc_v) = \cos(doc_u, cc_v)(1-\alpha) + link\left(\frac{doc_u, cc_v}{HN}\right)\alpha \quad \text{..... } 6a$$

$$f(doc_u, cc_v) = jac(doc_u, cc_v)(1-\alpha) + link\left(\frac{doc_u, cc_v}{HN}\right)\alpha \quad \text{..... } 6b$$

$$f(doc_u, cc_v) = pea(doc_u, cc_v)(1-\alpha) + link\left(\frac{doc_u, cc_v}{HN}\right)\alpha \quad \text{..... } 6c$$

where $0 \le \alpha \le 1$

Table 1: NM Neighbor Matrix, with k = 2 and Θ= 0.4 of Dataset D

	doc_0	doc_1	doc_2	doc_3	doc_4	doc_5
doc_0	1	0	0	0	0	1
doc_1	0	1	0	0	1	0
doc_2	0	0	1	1	1	0
doc_3	0	0	1	1	1	1
$adoc_4$	0	1	1	1	1	1
doc_5	1	0	0	1	1	1

where, *HN* is the highest number of neighbors possible for document doc_u to centroid of cluster v(cc_v) obtained using link(docu, cc_v), and α is a user defined value, that assigns weight to link() or similarity measure generally, based on dataset. *HN*is set to'n', where in clustering process all documents 'n' are involved (like in k-means). If cluster $Clus_v$ is considered *HN* is set with maximum neighbors from a document to the centroid c_j of the cluster j. *HN* is a normalization factor such that count of neighbor with in cluster lies between[0, 1]. If link(doc_u, cc_v) is set to 0, implies doc_u, cc_v has no shared neighbors.(doc_u, cc_v) value ranges [0,1] when $0 \le \alpha \le 1$. Thus, if α is 0, function f(doc_u, cc_v) is simple pair wise similarity cosine/jaccard/pearson thus ignores neighbor information, and if α is 1 the f(doc_u, cc_v) is depends purely neighbor information for clustering and thus ignores similarity measures cosine/jaccard/pearson. A good mix of similarity measure and link function gives better measure of closeness. These new similarity measures are complete and when used in clustering process leads to coherent clustering.

Table 2: DataSet D with k=2 &Θ=0.4 ,Neighbor matrix (NM') with cluster centroids

	doc_0	doc_1	doc_2	doc_3	doc_4	doc_5	cc1	cc2
doc_0	1	0	0	0	0	1	1	0
doc_1	0	1	0	0	1	0	0	1
doc_2	0	0	1	1	1	0	0	0
doc_3	0	0	1	1	1	1	1	0
doc_4	0	1	1	1	1	1	0	1
doc_5	1	0	0	1	1	1	1	0

This new matrix NM' is used to find shared neighbors between centroid of cluster v and document doc_u.Thus k columns are added to neighbor matrix NM. Thus n by (n + k) are the dimensions of new matrix, denoted by NM'. Fig. 2 depicts the new matrix NM'. Link(doc_u,cc_v)

is used to find the count of shared neighbors of doc_u,cc_v[6] and is calculated as a product of u^{th} row,$(n+ v)^{th}$ column of NM'.

$$link\left(doc_u, cc_v\right) = \sum_{m=1}^{n} NM'[u,m] \times NM'[m,v] \qquad .....7$$

4. INITIAL CLUSTER CENTERS SELECTION USING SHARED NEIGHBORS

K-Means and its variants require initial cluster centers (Seed points) to be specified. Selection of seed points is a primary task for clustering as it leads to better cluster results with best seeds. In literature algorithms like random, buckshot [3], and fractionation [3] were proposed. Here we propose a shared neighbor based method to find seed documents. The intension of choosing seed document is to see that the seed should be very similar to a set of documents and most dissimilar to other seed documents, such that these seeds produce well separated clusters. Number of neighbors can be determined from a neighbor matrix NM and the common neighbors is obtained using link(doc_u, doc_v). If a document is a neighbor of two documents then we say it is a shared neighbor of those two documents. Find shared neighbors of doc_u, doc_v represented as SHN(doc_u, doc_v) using neighbor matrix M as follows:

$NB\_List(doc_u)$ where $1 \le p \le n$ is a collection of documents with $NM[u,p] \ge 1$.

SHN(doc_u, doc_v*) = $NB\_List(doc_u) \cap NB\_List(doc_v)$ where $u \neq v$.*

Count the total no. of SHN(doc_u, doc_v*)*

For example from the above table

Link(d3, d4) = 3

SHN(d3, d4) = {d3, d4, d5} -> 3

SHN(d1, d2) = {Ø} -> 0

SHN(d1, d4) = {d6} -> 1

Arrange the Shared Neighbor combinations in the descending order of no. of neighbors. Now find the disjoint or minimum common neighbor combinations. If the combinations have any disjoint sets then consider it into the list otherwise ignore it from top of the sorted combinations. Consider the combination SHN(d3, d4) ∩ SHN(d1, d4) = 0, then consider (d3, d4) and (d1, d4) as possible candidates for initial centroid with k=2. After finding all disjoint combinations find the highest similarity document from the combination of shared neighbor candidates and consider that document as the initial centroid of one cluster. Highest Similarity of candidates (doc_u, doc_v) is calculated as follows

$$HS\left(doc_u, doc_v\right) = \max\left(\sum_{l=1}^{n} sim\left(doc_u, doc_l\right), \sum_{l=1}^{n} sim\left(doc_v, doc_l\right)\right) \qquad .....8$$

Assume the highest similarity document in the combination of (d3, d4) is d3 then select d3 as the initial centroid of cluster 1 and so on repeat the process for k cluster.

Algorithm : Finding Initial Centers with Shared Neighbors

SH_INI_CEN(k, D_t){

For each document docu in D_t

Find NB_List(docu) is a collection of documents from neighbor matrix M with M[i,k] ≥ 1, 1≤ k≤ n.

$SHN(doc_u, doc_v) = NB\_List(doc_u) \cap NB\_List(doc_v)$

// where $u \neq v$, and $1 \leq u,v \leq n$.

Count the total no. of $SHN(doc_u, doc_v)$

Arrange the $SHN(doc_u,doc_v)$ collection in the descending order of neighbors where $u \neq v$, and $1 \leq u,v \leq n$.

// Finds disjoint sets in dataset

For each $SHN(doc_u, doc_v)$ combination from top list find the disjoint sets.

do{

$SHN(doc_u, doc_v) \cap SHN(doc_q,doc_r) = \emptyset$) $1 \leq u,v \leq n$, $1 \leq q,r \leq n$ where $q \neq r$ and $u \neq v$

Candidates_list ← { (doc_u, doc_v), (doc_q,doc_r) }

} while(true)

//SHN(x,y)!= null or no.of candidates selected is k.Where k is the no. of initial centroids to be found

Do{

For each candidate combination find Highest Similarity(HS) using

$$HS(doc_u, doc_v) = \max\left(\sum_{l=1}^{n} sim(doc_u, doc_l), \sum_{l=1}^{n} sim(doc_v, doc_l)\right)$$

And assign the document that has highest similarity as initial centroid of a cluster

}

}while (k centroids are found)}

5. APPROACH TO DETERMINE PARTITIONS IN A DATASET

The family of k-means requires k-value representing the number of partitions to be specified before clustering. Unless the user is well acquainted with the dataset, or user is a domain expert, it is practically impossible to guess the correct value of k for a previously unseen collection of documents. The proposed K find measure is based on clustering method and K_m where K_m is largest possible clusters for a given dataset. Bisecting k-means is employed to determine k dynamically. In each iteration, till K_m clusters are formed it splits the least compact cluster. In our approach we propose a shared neighbor based heuristic measure to determine the compactness of the cluster. Our assumption is that a cohesive cluster is tightly packed around the representative of the cluster and has large number of shared neighbors and on the other hand non-cohesive cluster consists of sub-clusters and has a least number of shared neighbors. We split the cluster that has minimum number of normalized *cluster-confined* neighbors. To measure the compactness of a cluster, we employed the Shared Neighbor split measure.

After each split in bisecting k-means, calculate the Ideal Value of the partition. Let 'Ɗ' be the document with maximum cluster-confined neighbors, let 'μ_i' denote the count of ' cluster-confined shared neighbors of i^{th} partition from '$Ɗ_i$' and 'ç' represent the count of shared neighbors between '$Ɗ'_i$-'$Ɗ'_j$ of two clusters say i and j. Then we define $\mathcal{IV}$ representing ideal_value of clustering at iteration **T,** ratio of Confined-Cluster Coherency obtained by 'μ_i' to maximum of Between-Cluster Coherency obtained by 'ç'. Maximum $\mathcal{IV}(T)$ gives the best value at **T.** This is simplified as follows.

$$\mathcal{IV}(T) = MAX\left(\frac{\mu_i}{MAX(ç)}\right) \quad \text{..... ..... 9}$$

where $1 \le i \le T$

To maximize $\mathcal{IV}$, cluster-confined coherency is in the numerator and to minimize the between cluster coherency, maximum between-cluster coherency is considered in the denominator, when the larger of this neighbor value is minimized, other values are automatically be smaller than this value, minimum between-cluster coherency indicates the separateness between clusters. Thus, maximum ratio specifies the best cluster coherency achieved at iteration **T**. The $\mathcal{IV}$ values at different T can be used to determine best value of k.

Bisecting k-means is considered as the clustering algorithm, initially beginning with whole dataset, the dataset it partitioned into two, and the $\mathcal{IV}$ value at t=2 is calculated. In each iteration(**T**) the cluster with least confined neighbors is split and the ideal_value (**T**) is calculated. The best value of k is for the clustering (**T**) that has maximum $\mathcal{IV}$. With the increase in number of clusters small cohesive clusters should be formed. The process of splitting can be repeated until number of fine tuned clusters are formed or terminate the moment first k value is obtained. Best k-values are determined by comparing $\mathcal{IV}$ values obtained at iterations ***T-1, T, T+1.*** The condition called *local optimum*, which occurs when the relations $\mathcal{IV}(T) > \mathcal{IV}(T-1)$ *and* $\mathcal{IV}(T-1) < \mathcal{IV}(T+1)$, then we can say at iteration ***T*** the compactness of the cluster is maximum, hence best **K** value is at **iteration *T***. To obtain fine tuned clusters this process, may be continued or terminate after K_m clusters are formed.

$$Strength_v = \frac{TSH_v}{\|clus_v\|} \quad \text{..... ..... 10}$$

$$Min\left(Strength_v\right) \quad 1 \le v \le k$$

Identifying the Cluster to be Split

SHNSplit(p, $clusters_p$)

{
for i =1 to p // compute shared neighbors
do
for document d_v in $clusters_i$
$TSH_i = TSHi + \Sigma_{u=1, u \neq v}^{|clusters_i|} link(d_v, d_u)$
$Strength_i = TSH_i / |clusters_i|$
done
return i whose strength is minimum ie minimum($Strength_i$)
}

6. CLUSTERING ALGORITHM

We use Bisecting KMeans partitioning approach, where complete dataset will be considered initially. In each iteration, till required number of clusters formed, it selects a cluster to be bisected based on a heuristic measure. Bisecting KMeans is termed as efficient, because it assign a document to a group by comparing the document in the bisecting cluster with the two new cluster centers for similarity thus ignores the comparison with all the formed cluster centers.

In general, the heuristic measure for splitting a cluster is compactness. In [2], cluster compactness is measured in three ways viz., firstly, largest remaining cluster split, secondly, quality of cluster computed using intra cluster similarity or thirdly, the combination of both. After performing many experiments they concluded that the difference with these two measures is very small and thus, recommended to split the largest remaining cluster. Though, largest remaining cluster split produces better quality than others. The disadvantage of this process is that when there is a large cluster that is tightly coupled, this approach still splits the large cluster, which is going to degrade the quality of clusters formed. The cluster that has small intra cluster similarity is termed to be low quality or loosely coupled cluster, and hence the best cluster to be split. With this approach determining the intra-cluster similarity is a time consuming task.

In [16] we experimented with the above three measures and neighbor information and found that considering neighbor information in clustering process produces better results. The information of shared neighbors, gives us how closely the documents are with in a cluster. Hence, we use the above proposed shared neighbor based heuristic measure to identify the cluster to be split.

The following is the Shared Neighbor Based Text Clustering SHNBTC algorithm.

Text Clustering using Shared Neighbors

SHNBTC Algorithm

Build Similarity and Neighbor Matrix

SM is selected criterion measure for clustering.

D is dataset, k is no. of clusters,

Dt = D;

p= 2

While (p<=k)

do

split the cluster Dt into 2 *//using SM &Kmeans*

call SH_INI_CEN(2, Dt) *// finds 2 initial centers from Dt*

Step (i) for each docu ϵ Dt do

Clusv←docu iff SM(docu, ccv) is maximum

// where 1≤v≤2 ,$cc_v$is thev^{th} centroid and Clusv is j^{th} Cluster

done

re-compute the centroids with the newly assigneddocuments and repeat the above for loop

goto Step (i) till convergence.

// Find the cluster to be split from p clusters using SHNSplit Algorithm

cid = call SHNSplit(p, clustersp)

// cid is the cluster to be split , p is the no. of clusters

// clustersp is p clusters formed

p++

done

7. EXPERIMENTS

7.1 Datasets

We experimented with benchmark datasets "Reuters 21578" and Classic. DT our designed dataset of 200 documents of research papers containing four different categories, collected from web. We have considered acq, trade, money, ship, gun, crude and stock sub-categories of topics category of reuters and formed 3 datasets named reu1, reu2 and reu3 datasets. We selected 800 documents of classic dataset and formed three datasets where CL_1 contains combination of all four categories, CL_2 contains only cisi category and CL_3 a combination of cacm and med.

The Datasets features are presented in Table 3.

Table 3 : Dataset Characteristics

Sno	Data sets	No.of Docs.	No. Of Classes	Min category size	Max category size	Avg.File size	Avg. Pairwise similarity
1	Dt	197	4	46	50	156	0.0601
2	Reu1	180	6	30	30	108	0.1087
3	Reu2	300	4	31	121	99	0.1325
4	Reu3	140	5	25	59	158	0.1728
5	CL_1	800	4	200	200	203	0.0304
6	CL_2	200	4	15	68	163	0.0648
7	CL_3	400	3	35	146	221	0.0329

7.2 Pre-Processing

Before building the datasets for our experiments we eliminated documents with single word file size. For the datasets considered calculated average file size and ignored those documents that are less than average file size. On each category we have applied the file reduction procedure where we considered the documents of a category in the dataset satisfying average file size and eliminated other documents of the category. To achieve this we built a Boolean vector space representation of documents where for each category average file size is determined and pruned

documents that are with length less than average file size, thus forming valid documents. On these documents we applied preprocessing which includes, tokenization of input file, removal of special characters, removal of stop words, applied stemming to derive stem words, identified unique terms and built a vector of term document representation. Then we calculated document frequency of all terms and removed less frequent terms from the vector as the inclusion of these terms form clusters of small sizes. The terms with high frequency of occurrence are also pruned for they will not contribute to clustering process.

7.3 Results Analysis

Firstly, performance of initial centers is considered, following it, automatic determination of number of partitions in a given dataset, are described, next we see the performance of SBTC, RBTC and SNBTC approaches, where SBTC is Simple Bisecting K-means Text Clustering, RBTC is Rank Based Text [9] implemented for kmeans is extended to bisecting k-means in this work and proposed SNBTC, Shared Neighbor Based Text Clustering are analyzed and lastly we compare the effect of applying proposed approaches in clustering algorithm.

Figure 1 depicts the performance of Initial centers methods, where in Sequential, Random, Rank Neighbors and Shared Neighbor based are compared. For each type of initial centers choosen, we have run Bisecting K-means clustering, and the quality of the clusters formed are evaluated. To compare the results we used entropy as the quality measure. The lesser the value of entropy, the better is its quality, and the proposed shared neighbor seed document selection method, has showed significant improvement in the clustering process.

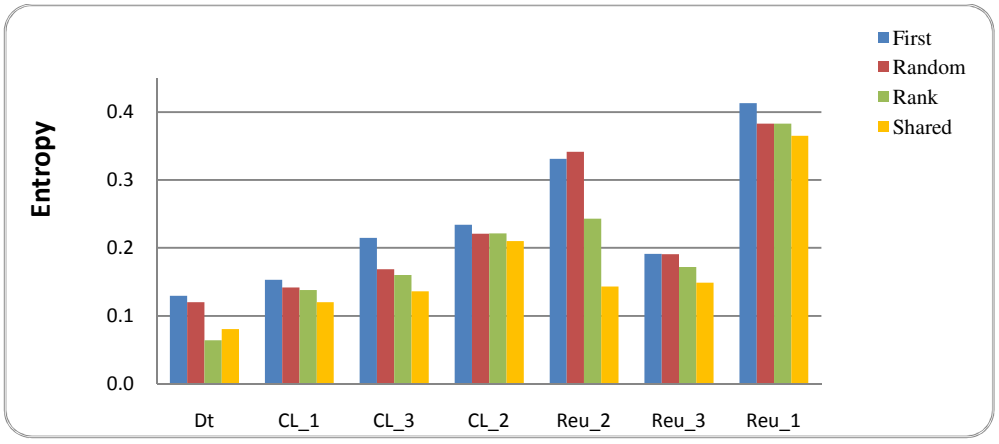

Figure 1: Results of Bisecting K-means with Initial Centers

Table 4 shows different k values, where thematic cohesive clusters are expected to form. At these k-values simple bisecting k-means is applied and observed intra cluster similarity to be maximum at these k's. The experiments showed quite accurate results.

Table 4 : Best k-values

Dataset	K
Dt	4,8,9
CL_1	4,6,9
CL_2	4,6,12
CL_3	5,9,12
Reu_1	4,6,9
Reu_2	3,6,7,9
Reu_3	4,6,9

At these specified k, the clustering produced cohesive clusters.

Figure 2: In bisecting k-means, when a new cluster is formed, we have computed ideal_values for each bisect step (iteration) and the iteration where these ***IV*** values are maximum, the k value is set with that iteration number. The ideal_values obtained at iteration **T** are all normalized in the range [0,1]. Figure 2 depicts the normalized ***IV*** values at different k's on CL_1, CL_2 and CL_3 datasets. As the number of clusters increases, the cohesive small clusters get formed, hence the approach specified can be extended to get small cohesive clusters or can terminate at K_m where the maximum cluster value is specified by user in case of a known dataset or at the first k where this criteria is satisfied.

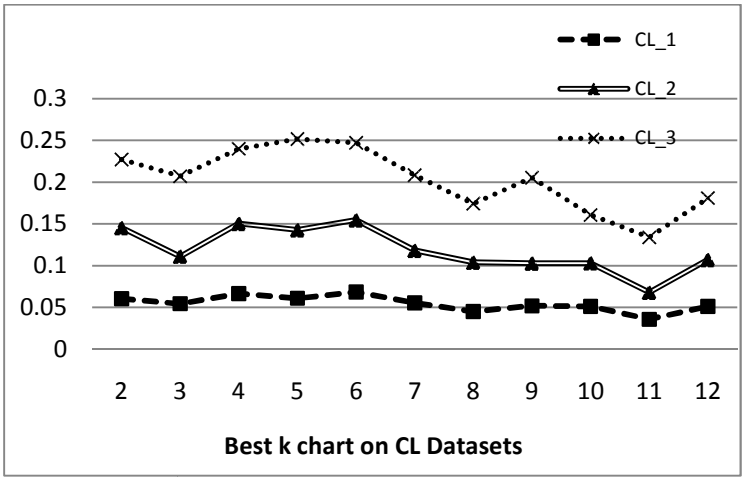

Figure 2 : Automatic determination of number of partitions

The tendency of maximum ***IV(T)*** occurrence is observed when the k value is small. This may be due to the fact that minimum number of common neighbors exists between clusters when number of clusters being 2 or 3 thus maximizing ***IV*** value.

Table 5: Entropy on Reuters Dataset with SNBTC

	Cosine	Jaccard	Pearson
Reu_1	0.404617	0.327961	0.446074
Reu_2	0.259109	0.303612	0.226766
Reu_3	0.321275	0.324401	0.322054

Figure 3 depicts the performance of proposed SHNBTC with all the three similarity measures on Classic datasets. The proposed approach performed well in all the three case, but best clusters for classic dataset is observed with Jaccard measure. The cluster quality varies with similarity measures and the datasets considered.

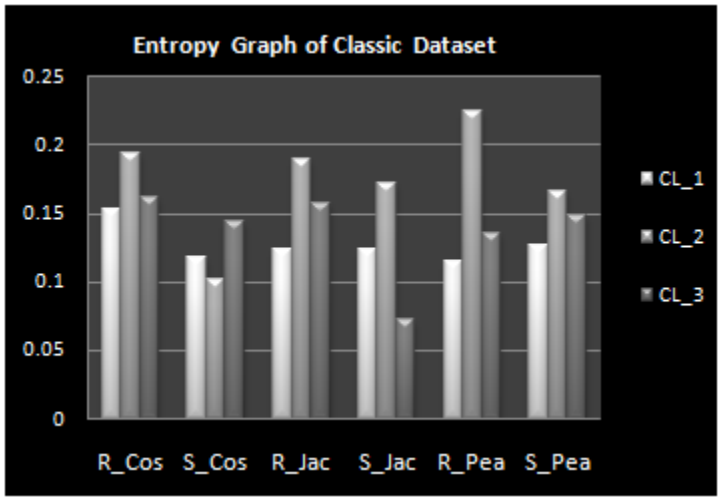

Figure 3: Entropy graph on RBTC and SNBTC on classic dataset

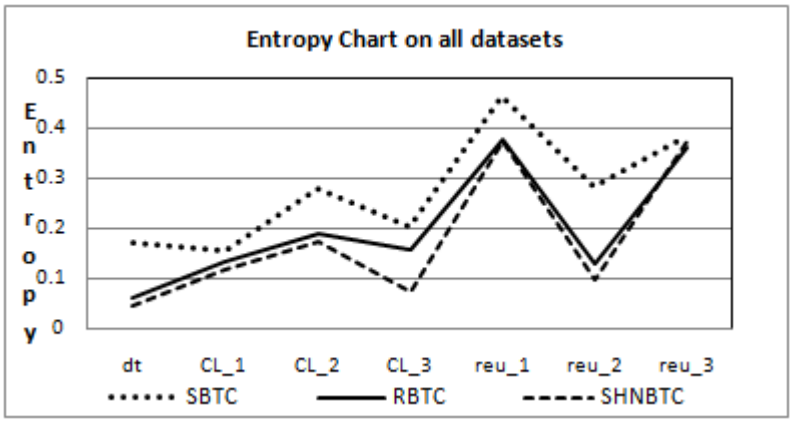

Figure 3 : Comparision of SBTC, RBTC and SHNBTC

Figure 4 compares SBTC,RBTC with proposed SHNBTC on all the datasets considered in our experiments and indicates that proposed shared neighbor measure works better and improves cluster coherency. In case of RBTC and SHNBTC α value chosen ranged between 0.8 -.95 thus

giving importance to neighbor information. From this one can say neighbor data influence cluster quality and our proposed approached formed coherent clusters.

Figure 5 shows the application of proposed heuristic measure, initial centroids seperately in the clustering process, and compared it with the proposed SNBTC algorithm and we observe that the clusters formed with our algorithm are cohesive.

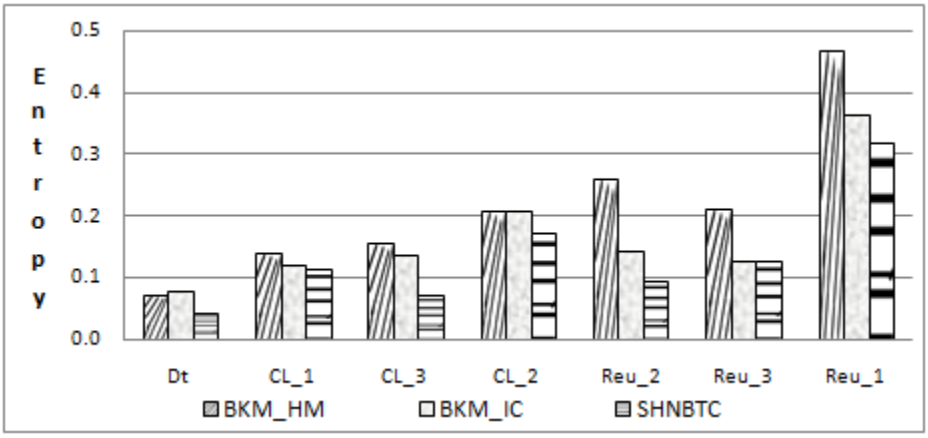

Figure 4 : Result of applying HM, IC and Proposed

8. CONCLUSIONS

In this paper an attempt is made to improve performance of bisecting k-means. This work has given a neighbor based solution to find number of partitions in a dataset. Then, proceeds to give an approach to find k initial centres for a given dataset. The family of k-means require k initial centers and number of clusters to be specified. In this work we have addressed these two issues with neighbor information. Then we proposed a heuristic measure to find the compactness of a cluster and when employed in selecting the cluster to be split in bisecting step has shown improved performance. All the three approaches proposed, when applied to bisecting k-means shown better performance. We have experimented with neighbors and links concept specified in [6],[7] and found that the cluster quality improves with neighbor information combined with text clustering similarity measures. Neighbors are used in determining the compactness of clusters in bisecting k-means. In our previous study we have noticed that Jaccard and Cosine outperforms Pearson coefficient with link function. It is observed that the clusters formed are cohesive. Efficiency of clustering results are based on representation of documents, measure of similarity and clustering technique. In our future work semantics knowledge shall be incorporated in the document representation to establish relations between tokens and study various measures semantic and similarity on these representations with neighbors based clustering approaches for better clustering results.

REFERENCES

[1] Salton G., (1989) "Automatic Text Processing : The Transformation, Analysis and Retrieval of Information by Computer", Addison-Wesley, New York.

[2] Steinbach M.,et.al.,(2000) "A Comparison of Document Clustering Techniques", Workshop of Knowledge Discovery And Data Mining on Text Mining.

[3] Cutting D.R., et.al.,(1992) "Scatter/Gather: A Cluster-Based Approach to Browsing Large Document Collections". In Proceedings of ACM SIGIR, pp.318-329.

[4] Huang A.,(2008),"Similarity Measures For Text Document Clustering", In Proceedings of New Zealand Computer Science Research Student Conference.
[5] Strehl. et.al,(July 2000) "Impact Of Similarity Measures On Web-Page Clustering". In Workshop on Artificial Intelligence for Web Search,pp.58-64
[6] Guha S.,et.al.,(2000) "ROCK: ARobust Clustering Algorithm For Categorical Attributes", in Information Systems, Volume 25 No. 5, pp. 345–366.
[7] Luo, Yanjun Li, Chung Soon M.,(2009) "Text Document Clustering based on neighbors", Data and Knowledge Engineering volume 68 issue 11 pp 1271-1288.
[8] Sandhya N, Sri Lalitha Y, Govardhan A et al. (2011), "Analysis Of Stemming Algorithms For Text Clustering", In International Journal of Computer Science Issues, Vol. 8, Issue 5, No 1 pp. 352-359.
[9] Sri Lalitha Y. Govardhan A. et.al, (2011), "Neighbor based Text Document Clustering", (IJCSIS) International Journal ofComputer Science and Information Security, Vol. 9, No. 8, pp.146-153.
[10] Li A. Y., Chung S.M., (2007)"Parallel Bisecting Kmeans With Prediction Clustering Algorithm", in The Journal of Supercomputing 39 (1) pp. 19–37.
[11] Jarvis R.A., Patrick E.A. (1973), "Clustering Using A Similarity Measure Based On Shared Near Neighbors", IEEE Transactions on Computers C-22 (11).
[12] Gowda K.C., Krishna G., (1978) "Agglomerative Clustering Using the Concept of Mutual Nearest Neighborhood", Pattern Recognition volume 10, No. 1, pp. 105–112.
[13] Ester M., et.al. (1996) "A Density-Based Algorithm For Discovering Clusters In Large Spatial Databases With Noise", In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining.
[14] Ertöz L., Steinbach M., Kumar V., (2004) "Finding Topics In Collections Of Documents: A Shared Nearest Neighbor Approach", In Clustering and Information Retrieval, Kluwer Academic Publishers, pp.83-104.
[15] Hanan A. and Mohamed Kamel,(2003), "Refined Shared Nearest Neighbors Graph for Combining Multiple Data Clustering", In M.R. Berthold et al. (Eds.): LNCS 2810, Springer-Verlag Berlin Heidelberg, pp. 307-318.
[16] Y. Sri Lalitha,Dr. A. Govardhan, (2013) "Analysis of Heuristic Measures for cluster Split in Bisecting K-means", CiiT International Journal of Data Mining and Knowledge Engineering, Vol 5, No 12, pp. 438-443.

AUTHORS

First A. Y. Sri Lalitha, M.Tech from JNTUH, (Ph.D) from ANU, Guntur. Working as Associate Professor, in the Department of CSE, GRIET, Hyderabad, having total 17 years of Teaching Experience. Taught subjects related to programming. Areas of interest include Algorithms, Data Structures, Databases, Information Retrieval, Data Mining and Text Mining. Had seven publications in the field of data mining in international/ National Journals and Conferences including IEEE, Springer.

Second A. Dr. A. Govardhan, M.Tech CSE from Jawaharlal Nehru University, Delhi, Ph.D from Jawaharlal Nehru Technological University Hyderabad. He has 20 years of teaching experience. Presently, Director & Professor of CSE at School of Information Technology in JNTU Hyderabad. He held various key roles in the University, now he is the Chairman of CSI Hyderabad Chapter. His areas of interest are Data Mining, Text Mining and Information Retrieval. He has guided 18 P.hds theses . 125 M.Tech projects and has152, Research Publications in International/ National Journals and Conferences including IEEE, ACM, Springer and Elsevier. He is the recipient of 21 International and National Awards including A.P. State Government Best Teacher Award.