Farthest Neighbor Approach for Finding Initial Centroids in K-Means

N. Sandhya *sandhyanadela@gmail.com*
Professor/CSE
VNR Vignan Jyothi Institute of Engineering &Technology
Hyderabad, 500 090,India

K. Anuradha *kodali.anuradha@yahoo.com*
Professor/CSE
Gokaraju Rangaraju Institute of Engineering and Technology
Hyderabad, 500 090,India

V. Sowmya *sowmyaakiran@gmail.com*
Associate.Prof/CSE
Gokaraju Rangaraju Institute of Engineering and Technology
Hyderabad, 500 090,India

Ch. Vidyadhari *chalasanividyadhari@gmail.com*
Asst.Prof/CSE
Gokaraju Rangaraju Institute of Engineering and Technology
Hyderabad, 500 090,India

Abstract

Text document clustering is gaining popularity in the knowledge discovery field for effectively navigating, browsing and organizing large amounts of textual information into a small number of meaningful clusters. Text mining is a semi-automated process of extracting knowledge from voluminous unstructured data. A widely studied data mining problem in the text domain is clustering. Clustering is an unsupervised learning method that aims to find groups of similar objects in the data with respect to some predefined criterion. In this work we propose a variant method for finding initial centroids. The initial centroids are chosen by using farthest neighbors. For the partitioning based clustering algorithms traditionally the initial centroids are chosen randomly but in the proposed method the initial centroids are chosen by using farthest neighbors. The accuracy of the clusters and efficiency of the partition based clustering algorithms depend on the initial centroids chosen. In the experiment, kmeans algorithm is applied and the initial centroids for kmeans are chosen by using farthest neighbors. Our experimental results shows the accuracy of the clusters and efficiency of the kmeans algorithm is improved compared to the traditional way of choosing initial centroids.

Keywords: Text Clustering, Partitional Approach, Initial Centroids, Similarity Measures, Cluster Accuracy.

1. INTRODUCTION

One of the important techniques of data mining, which is the unsupervised classification of similar data objects into different groups, is data clustering. Document clustering or Text clustering is the organization of a collection of text documents into clusters based on similarity. It is a process of grouping documents with similar content or topics into clusters to improve both availability and reliability of text mining applications such as information retrieval [1], text classification [2], document summarization [3], etc. During document clustering we need to address the issues like

defining the similarity of two documents, deciding the appropriate number of document clusters in a text collection etc.

2. DOCUMENT REPRESENTATION

There are many ways to model a text document. One way of representing a set of documents as vectors in a common vector space is known as the vector space model [4]. A widely used representation technique in information retrieval and text mining is 'bag of words' model.

In the vector space model every document is represented as a vector and the values in the vector represents the weight of each term in that document. Depending on the document encoding technique weight of the term will be varied. For constructing the vector space model, we need to find unique terms in the dataset. Each document d in the vector space model is represented as:

$$d = [wt_1, wt_2, \ldots wt_n] \quad (1)$$

Where, wt_i represents the weight of the term i and n represents the number of unique terms in dataset.

There are three document encoding methods namely, *Boolean, Term Frequency and Term Frequency with Inverse Document Frequency.*

2.1 Boolean

In Boolean representation, the weight of the term will have only one of the two values that is either 1 or 0. If the term exists in the document the weight of the term is 1 otherwise 0.

2.2 Term Frequency

In Term Frequency, the weight of the term is represented as the number of times the term is repeated in that document. In this method weight of the term is equal to frequency of the term in that document. The term frequency (TF) vector for a document is represented as:

$$d_{tf} = [\, tf_1 , tf_2 , \ldots\ldots\ldots tf_n \,] \quad (2)$$

where tf_i is the frequency of term i in the document and n is the total number of unique terms in the text database.

2.3 Term Frequency-Inverse Document Frequency

In Inverse document frequency, the term weight is high for more discriminative words. The IDF is defined via the fraction N/df_i, where, N is the total number of documents in the collection and df_i is the number of documents in which term i occurs.

Thus, the TF-IDF representation of the document d is:

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

To account for the documents of different lengths, each document vector is normalized to a unit vector (i.e., $||d_{tf\text{-}idf|}||=1$).

3. SIMILARITY MEASURES

One of the prerequisite for accurate clustering is the precise definition of the closeness between a pair of objects defined in terms of either the pair-wised similarity or dissimilarity. Similarity is often conceived in terms of dissimilarity or distance as well.

Similar documents are grouped to form a coherent cluster in document clustering. A wide variety of similarity and dissimilarity measures exists. The measures, such as cosine, Jaccard coefficient, Pearson Correlation Coefficient are similarity measures where as the distance measures like

Euclidian, Manhattan, Minkowski are dissimilarity measures. These measures have been proposed and widely applied for document clustering [5].

Similarity measure can be converted into dissimilarity measure:

$$\text{Dissimilarity}=1-\text{Similarity} \quad (4)$$

3.1 Cosine Similarity

The similarity between the two documents d_i , d_j can be calculated using cosine as

$$\cos(d_i, d_j) = \frac{\sum_{k=1}^{n}(d_{i,k} * d_{j,k})}{\sqrt{\sum_{k=1}^{n}(d_{i,k})^2 * \sum_{k=1}^{n}(d_{j,k})^2}} \quad (5)$$

Where n represent the number of terms. When the cosine value is 1 the two documents are identical, and 0 if there is nothing in common between them (i.e., their document vectors are orthogonal to each other).

3.2 Jaccard Similarity

The Cosine Similarity may be extended to yield Jaccard Coefficient

$$jaccard(d_i, d_j) = \frac{\sum_{k=1}^{n}(d_{i,k} * d_{j,k})}{\sum_{k=1}^{n} d_{i,k} + \sum_{k=1}^{n} d_{j,k} - \sqrt{\sum_{k=1}^{n}(d_{i,k} * d_{i,k})}} \quad (6)$$

3.3 Euclidean Distance

This Euclidean distance between the two documents d_i , d_j can be calculated as

$$Euclidean\ Distance\left(d_i, d_j\right) = \sqrt{\sum_{k=1}^{n}(d_{ik} - d_{jk})^2} \quad (7)$$

Where, n is the number of terms present in the vector space model.

Euclidian distance gives the dissimilarity between the two documents. If the distance is smaller it indicates they are more similar else dissimilar.

3.4 Manhattan Distance

The Manhattan distance between the two documents d_i , d_j can be calculated as

$$Manhattan\ Distance\left(d_i, d_j\right) = \sum_{k=1}^{n} | d_{ik} - d_{jk} | \quad (8)$$

4. ACCURACY MEASURE

To find the accuracy of the clusters generated by clustering algorithms, F-measure is used. Each cluster generated by clustering algorithms is considered as the result of a query and the documents in these clusters are treated as set of retrieved documents.[6] The documents in each category are considered as set of relevant documents. The documents in the cluster that are

relevant to that cluster are set of relevant documents retrieved. Precision, Recall and F-measure are calculated for each cluster. The overall accuracy of the clustering algorithm is the average of the accuracy of all the clusters.

Precision, recall and F-measure are calculated as follows:[7]

Precision = relevant documents retrieved /retrieved documents (9)

Recall = relevant documents retrieved / relevant documents (10)

F-Measure = (2 * Precision * Recall)/(Precision + Recall) (11)

5. RELATED WORK

The words that appear in documents often have many morphological variants in Bag of Words representation of documents. Mostly these morphological variants of words have similar semantic interpretations and can be considered as equivalent for the purpose of clustering applications. A number of *stemming Algorithms*, or *stemmers*, have been developed for this reason, which attempt to reduce a word to its *stem* or root form. Thus, the key terms of a document are represented by stems rather than by the original words.

A high precision stemmer is needed for efficient clustering of related documents as a preprocessing step [8].

The most widely cited stemming algorithm was introduced by Porter (1980). The Porter stemmer applies a set of rules to iteratively remove suffixes from a word until none of the rules apply.

Many clustering techniques have been proposed in the literature. Clustering algorithms are mainly categorized into Partitioning and Hierarchical methods [9,10,11,12]. Partitioning clustering method groups similar data objects into specified number of partitions. K-means and its variants [13,14,15] are the most well-known partitioning methods [16]. Hierarchical clustering method works by grouping data objects into a tree of clusters [17]. These methods can further be classified into agglomerative and divisive Hierarchical clustering depending on whether the Hierarchical decomposition is formed in a bottom-up or top-down fashion.

6. FARTHEST NEIGHBORS ALGORITHM

K-means algorithm is used to cluster documents into k number of partitions. In K-means algorithm, initially k-objects are selected randomly as centroids. Then assign all objects to the nearest centroid to form k-clusters. Compute the centroids for each cluster and reassign the objects to form k-clusters by using new centroids. Computing the centorids and reassigning the objects should be repeated until there is no change in the centroids. As the initial k-objects are selected randomly depending on the selection of these k-objects the accuracy and efficiency of the classifier will vary. Instead of selecting the initial centroids randomly, we are proposing to find the best initial centroids. The main intention for choosing the best initial centroids is to decrease the number of iterations for the partitioning based algorithms because the number of iterations to get the final clusters depends on the initial centroids chosen. If the number of iterations decreased then the efficiency of the algorithm will be increased. In the proposed method the initial centroids are chosen by using farthest neighbors. For the experimental purpose the algorithm chosen is k-means which is one of the well known partitioning based clustering algorithms. To increase the efficiency of the k-means algorithm instead of selecting k-objects randomly as initial centroids the k-objects are chosen by using farthest neighbors. After finding the initial centroids by using farthest neighbors apply k-means algorithm to cluster the documents.

The documents need to be preprocessed before applying the algorithm. Removing of stop words, performing stemming, pruning the words that appear with very low frequency etc., are the preprocessing steps. After preprocessing vector space model is built.

The algorithm works with dissimilarity measures. The documents are more similar if the distance between the documents is less else the documents are dissimilar. Algorithm for finding the initial centroids by using farthest neighbors is as follows:

Algorithm:

1. By using the dissimilarity measures construct dissimilarity matrix for the document pairs in the vector space model.
2. Find the maximum value from the dissimilarity matrix
3. Find the document pair with the maximum value found in step 2 and choose them as first two initial (i.e., these two documents are the farthest neighblrs)
4. For finding remaining specified number of initial centroids
    i. Calculate the centroid for already found initial centroids.
    ii. With the centroid calculated in step 4.i generate the dissimilarity matrix between centroid and all the documents except those chosen as initial centroids.
    iii. Find the maximum value from the dissimilarity matrix generated in step 4.ii and choose the corresponding document as next initial centroid.
5. Repeat the step 4 until the specified number of initial centroids are chosen.

Form the vector space model given below construct 3 clusters by using k-means algorithm and use the farthest neighbors for finding the initial centroids.

The algorithm is explained as follows for finding the initial centroids:

Step1: Consider term frequency vector space model for 8 documents and generate dissimilarity matrix using dissimilarity measures. (For the explanation we have considered manhattan distance)

Terms / Documents	1	2
1	2	10
2	2	5
3	8	4
4	5	8
5	7	5
6	6	4
7	1	2
8	4	9

TABLE 1: Vector Space Model.

Documents	1	2	3	4	5	6	7	8
1	0	5	12	5	10	10	9	3
2	5	0	7	6	5	5	4	6
3	12	7	0	7	2	2	9	9
4	5	6	7	0	5	5	10	2
5	10	5	2	5	0	2	9	7
6	10	5	2	5	2	0	7	7
7	9	4	9	10	9	7	0	10
8	3	6	9	2	7	7	7	0

TABLE 2: Dissimilarity Matrix.

Step 2: The maximum value from the dissimilarity matrix is 12.

Step 3: The document pair with the maximum value is (1,3) and there for the first two initial centroids are (2,10) and (8,4) which represents document 1 and 3 respectively.
Step 4: As mentioned in the problem, 3 centroids are required. Already 2 centroids are choosen from step 3. The remaiming 1 centorid need to be found.
Step 4.i: the centroid for (2,10) and (8,4) is (5,7)
Step 4.ii: Generating the dissimilarity matrix between (5,7) and (2,5), (5,8), (7,5), (6,4), (1,2), (4,9) which represents the documents 2,4,5,6,7 and 8 respectively
Step 4.iii: maximum value is 9 and the corresponding document is 7. Therefore the third initial centroid is (1,2)
Step 5: Required number of initial centroids is found so no need of repeating step 4.

Now by applying the k-means algorithm by using the initial centroids obtained above the clusters are formed are shown in fig 5.

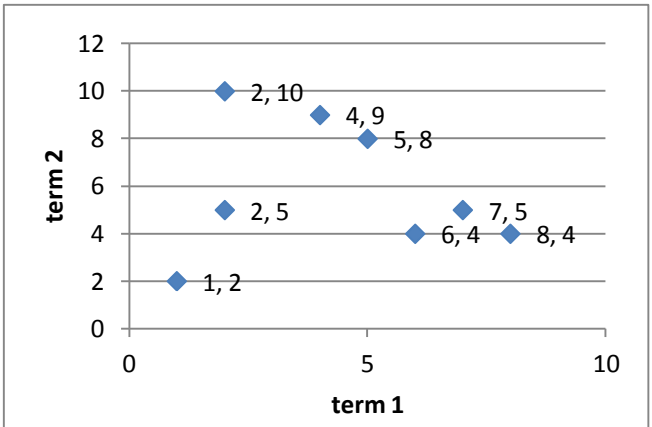

FIGURE 1: D documents plotting

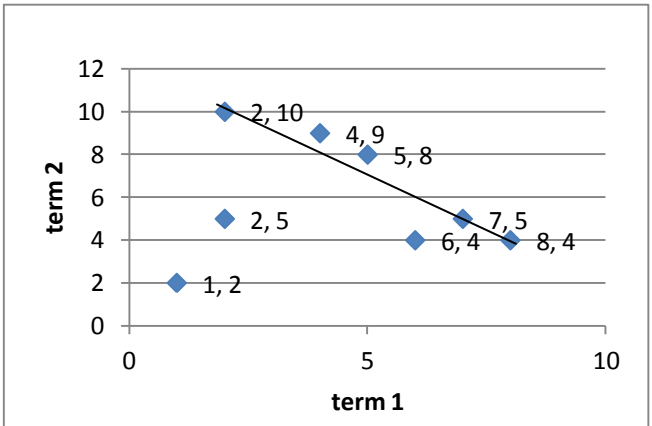

FIGURE 2: Finding the farthest neighbors

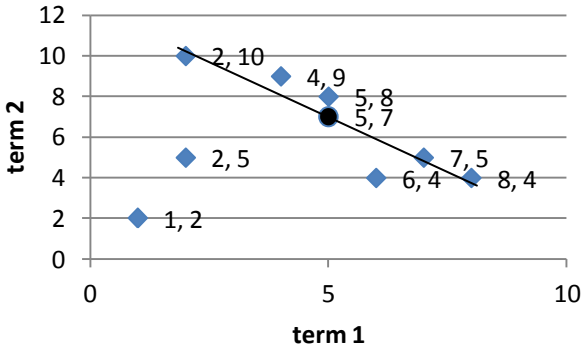

FIGURE 3: find the centroid of the farthest neighbors

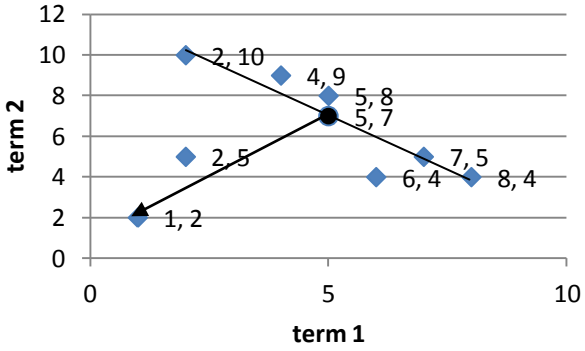

FIGURE 4: find the farthest point from the centroid

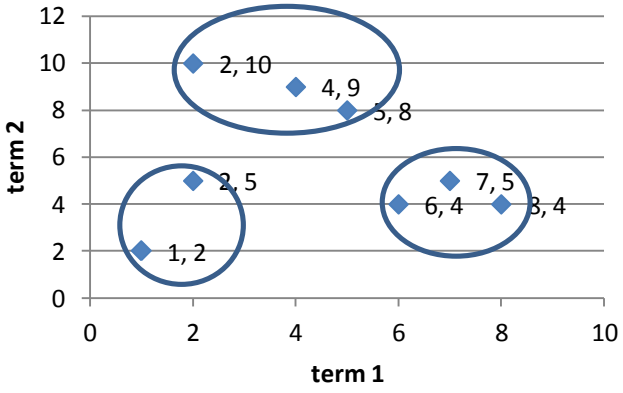

Figure 5: the final three clusters after applying k- means

7. EXPERIMENT

In this section, by using bench mark classic dataset a series of experiments are carried out to categorize the documents into predefined number of categories by using k-means algorithm. The k initial centroids for k-means algorithm is chosen randomly and by using farthest neighbors as explained in section 5. The accuracy of the clusters and efficiency of the algorithm is inspected.

7.1 Dataset

In this work the experiments are carried out on with one of the bench mark dataset i.e., Classic dataset collected from uci.kdd repositories. CACM, CISI, CRAN and MED are four different collections in Classic dataset. For experimenting 800 documents are considered after preprocessing the total 7095 documents.

The documents that consist of single words are meaningless to consider in the dataset. The documents that does not support mean length on each category is eliminated so that the number of files are reduced. For file reduction the Boolean vector space model of all documents is generated category wise. From this matrix the mean length of each category is calculated and the documents from the dataset that doesn't support mean length are considered as invalid documents and they are deleted from the dataset. After this process the left over documents are considered as valid documents. From these valid documents we collected 200 documents from each of the four categories of the classic dataset which are summing to 800 documents.

7.2 Pre-Processing

Preprocessing steps that take as input a plain text document and output a set of tokens (which can be single terms or n-grams) that are to be included in the vector model. In this work we performed removal of stop words and after taking users choice to perform stemming and built vector space model. We have pruned words that appear with very low frequency throughout the corpus with the assumption that these words, even if they had any discriminating power, would form too small clusters to be useful. Words which occur frequently are also removed. We need to preprocess the documents before applying the algorithm. Preprocessing consists of steps like removal of stop words, perform stemming, prune the words that appear with very low frequency etc. and build vector space model.

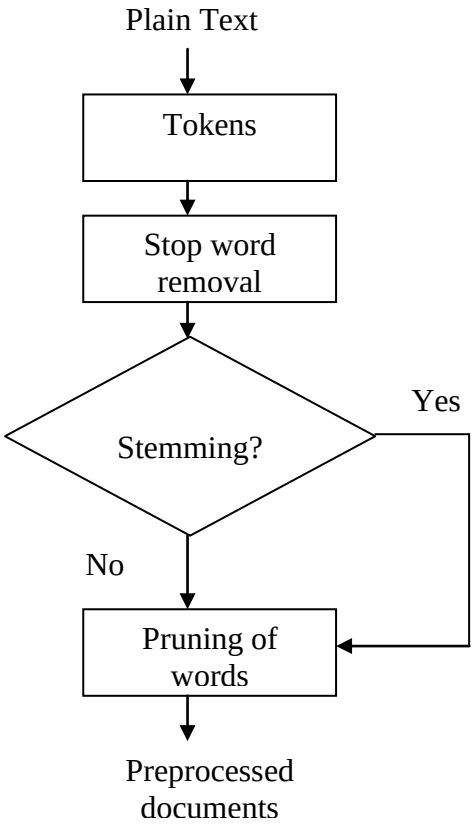

FIGURE 6: Preprocessing.

7.3 Results:

Classes / Clusters	Cis	Cra	Cac	Med	Clustering label	Precision	Recall	F-measure(in %)
Cluster1	5	173	0	2	Cra	0.96	0.865	91
Cluster2	184	23	1	1	Cis	0.88	0.92	89.9
Cluster3	10	3	199	4	Cac	0.921	0.995	95.6
Cluster4	0	1	0	193	Med	0.994	0.965	98.0

TABLE 3: Clustering the documents by using k-means, jaccard similarity measure and random selection of initial centroids.

The overall accuracy=average of all F-measure
i.e., **accuracy=93.60**
One iteration, includes computing the centroids and assigning the objects to the predefined number of clusters. These iterations are repeated until consecutive iterations yield same centroids.
Iterations =18

Classes / Clusters	Cis	Cra	Cac	Med	Clustering label	Precision	Recall	F-measure(%)
Cluster1	179	9	1	2	Cis	0.937	0.895	91.5
Cluster2	6	2	198	13	Cac	0.90	0.99	94.3
Cluster3	15	189	1	1	Cra	0.917	0.945	93
Cluster4	0	0	0	184	Med	1.00	0.92	95.8

TABLE 4: Clustering the documents by using k-means, jaccard similarity measure and farthest neighbors as initial centroids.

Accuracy=93.65
Iterations= 5

Classes / Clusters	Cis	Cra	Cac	med	Clustering label	Precision	Recall	F-measure
Cluster1	1	137	1	61	Cra	0.685	0.685	68
Cluster2	0	0	194	1	Cac	0.994	0.97	98
Cluster3	197	63	5	4	Cis	0.732	0.985	83.9
Cluster4	1	0	0	134	Med	0.99	0.67	79.9

TABLE 5: Clustering the documents by using k-means, Cosine similarity measure and random selection of initial centroids.

Accuracy=82.45
Iterations= 44

Classes / Clusters	Cis	Cra	Cac	Med	Clustering label	Precision	Recall	F-measure(%)
Cluster1	1	152	0	31	Cra	0.826	0.76	79.1
Cluster2	0	0	188	2	Cac	0.989	0.94	96.4
Cluster3	199	48	12	4	Cis	0.75	0.995	85.5
Cluster4	0	0	0	162	Med	I.00	0.81	89.5

TABLE 6: Clustering the documents by using k-means,cosine similarity measure and farthest neighbours as initial centroids.

Accuracy=87.62
Iterations= 7

8. COMPARISON WITH EXISTING METHODS

Harmanpreet singh et al. proposed a new method for finding initial cluster centroids for k-means algorithm[18] and also compared the new method with the existing methods[19,20,21,22,23,24,25,26,27]. In the new method, the required number of clusters should be supplied by the user. The Arithmetic mean of the whole data set represents the first cluster centre. Next the data is divided into two parts and mean of these two parts is calculated and these means act as second and third centres respectively. The process of dividing the dataset into parts and calculating the mean is repeated until k cluster centres are found. In this method the centroids of the clusters will vary depending on the division of the dataset into parts. So, to find the better cluster accuracy the algorithm need to run multiple times.

The farthest neighbor approach presented in this paper, the centroids are not varied. So running the program once is sufficient. The efficiency and accuracy of the k-means algorithm is increased when compared with the random selection of initial centroids. The tables 3,4,5,6 shows the experimental results.

9. CONCLUSIONS AND FUTURE WORK

Here we proposed new method for finding initial centroids by using farthest neighbors. The experimental results showed that accuracy and efficiency of the k-means algorithm is improved when the initial centroids are chosen using farthest neighbors than random selection of initial centroids. As the number of iterations decreased we can tell the efficiency is improved. We intend to apply this algorithm for different similarity measures and study the effect of this algorithm with different benchmark datasets exhaustively. In our future work we also intend to explore the other techniques for choosing the best initial centroids and apply them to partitional and hierarchal clustering algorithms so as to improve the efficiency and accuracy of the clustering algorithms.

10. REFERENCES

[1] O. Zamir, O. Etzioni, O. Madani, R.M. Karp, “Fast and intuitive clustering of web documents”, in: Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining, 1997, pp. 287–290.

[2] C.C. Aggarwal, S.G. Gates, P.S. Yu, “On the merits of building categorization systems by supervised clustering”, in: Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 1999, pp.352–356.

[3] B. Larson, C. Aone, “Fast and effective text mining using linear-time document clustering”, in: Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, vol. 98(463), 1999, pp. 16–22.

[4] Salton, G., Wong, A., Yang, C.S. (1975). “A vector space model for automatic indexing”. Communications of the ACM, 18(11):613-620.

[5] Anna Huang, "Similarity Measures for Text Document Clustering", published in the proceedings of New Zealand Computer Science Research Student Conference 2008.

[6] *Saurabh Sharma, Vishal Gupta. "*Domain Based Punjabi Text Document Clustering". *Proceedings of COLING 2012: Demonstration Papers*, pages 393–400,COLING 2012, Mumbai, December 2012.

[7] D. Manning, Prabhakar Raghavan, Hinrich Schütze, *"An Introduction to Information Retrieval Christopher"*, Cambridge University Press, Cambridge, England

[8] M.F. Porter, "*An algorithm for suffix stripping*", Program, vol.14, no.3, pp. 130−137, 1980.

[9] C.J.Van Rijsbergen,(1989), *"Information Retrieval"*, Buttersworth, London, Second Edition.

[10] G. Kowalski,"*Information Retrieval Systems – Theory and Implementation*", Kluwer Academic Publishers, 1997.

[11] D.R. Cutting, D.R. Karger, J.O. Pedersen, and J.W. Tukey, Scatter/Gather: "*A Cluster-based Approach to Browsing Large Document Collections*", SIGIR '92, Pages 318 – 329, 1992.

[12] O. Zamir, O. Etzioni, O. Madani, R.M. Karp, *Fast and Intuitive Clustering of Web Documents*, KDD '97, Pages 287-290, 1997.

[13] G. Salton, M.J. McGill, "*Introduction to Modern Information Retrieval"*. McGraw-Hill, 1989.

[14] A. Ehrenfeucht and D. Haussler. "*A new distance metric on strings computable in linear time"*. Discrete Applied Math, 1988.

[15] M. Rodeh, V. R. Pratt, and S. Even. "*Linear algorithm for data compression via string matching"*. In Journal of the ACM, pages 28(1):16–24, 1981.

[16] Peter Weiner. "*Linear pattern matching algorithms"*. In SWAT '73: Proceedings of the 14th Annual Symposium on Switching and Automata Theory (swat 1973), pages 1–11, Washington, DC, USA, 1973. IEEE Computer Society.

[17] R. Baeza-Yates, B. Ribeiro-Neto, "*Modern Information Retrieval*", Addison-Wesley, 1999.

[18] Harmanpreet singh**,** Kamaljit Kaur, *"New Method for Finding Initial Cluster Centroids in K-means Algorithm"*,International Journal of Computer Applications (0975 – 8887) Volume 74– No.6, July 2013

[19] Anderberg, M, *"Cluster analysis for applications"* ,Academic Press, New York 1973.

[20] Tou, J., Gonzales, "*Pattern Recognition Principles*" ,Addison-Wesley, Reading, MA, 1974.

[21] Katsavounidis, I., Kuo, C., Zhang, Z., "*A new initialization technique for generalized lloyd iteration*", IEEE Signal Processing Letters 1 (10), 1994, pp. 144-146.

[22] Bradley, P. S., Fayyad, "Refining initial points for K-Means clustering", Proc. 15th International Conf. on Machine Learning, San Francisco, CA, 1998, pp. 91-99.

[23] Koheri Arai and Ali Ridho Barakbah, "*Hierarchical k-means: an algorithm for centroids initialization for k-means*", Reports of The Faculty of Science and Engineering Saga University, vol. 36, No.1, 2007.

[24] Samarjeet Borah, M.K. Ghose, *"Performance Analysis of AIM-K-means & K- means in Quality Cluster Generation"*, Journal of Computing, vol. 1, Issue 1, December 2009.

[25] Ye Yunming, "*Advances in knowledge discovery and data mining",* (Springer, 2006).

[26] K. A. Abdul Nazeer and M. P. Sebastian, " Improving the accuracy and efficiency of the k-means clustering algorithm", Proceedings of the World Congress on Engineering, London, UK, vol. 1, 2009.

[27] Madhu Yedla, S.R. Pathakota, T.M. Srinivasa, "*Enhancing K-means Clustering Algorithm with Improved Initial Centre*", International Journal of Computer Science and Information Technologies, 1 (2) , 2010, pp. 121-125.