Document Topic Generation in Text Mining by Using Cluster Analysis with EROCK

Rizwan Ahmad qazirizwan.ahmad@yahoo.com

Computer Engineering Department
National University of Science & Technology
Rawalpindi, 46000, Pakistan

Dr. Aasia Khanum aasia@ceme.nust.edu.pk

Computer Engineering Department
National University of Science & Technology
Rawalpindi, 46000, Pakistan

Abstract

Clustering is useful technique in the field of textual data mining. Cluster analysis divides objects into meaningful groups based on similarity between objects. Copious material is available from the World Wide Web (WWW) in response to any user-provided query. It becomes tedious for the user to manually extract real required information from this material. This paper proposes a scheme to effectively address this problem with the help of cluster analysis. In particular, the ROCK algorithm is studied with some modifications. ROCK generates better clusters than other clustering algorithms for data with categorical attributes. We present an enhanced version of ROCK called Enhanced ROCK (EROCK) with improved similarity measure as well as storage efficiency. Evaluation of the proposed algorithm done on standard text documents shows improved performance.

Keywords: Text Mining, Cluster Analysis, Document Similarity, Topic Generation.

1. INTRODUCTION

A considerably large portion of information present on the World Wide Web (WWW) today is in the form of un-structured or semi-structured text data bases. The WWW instantaneously delivers huge number of these documents in response to a user query. However, due to lack of structure, the users are at a loss to manage the information contained in these documents efficiently. In this context, the importance of data/text mining and knowledge discovery is increasing in different areas like: telecommunication, credit card services, sales and marketing etc [1]. Text mining is used to gather meaningful information from text and includes tasks like Text Categorization, Text Clustering, Text Analysis and Document Summarization. Text Mining examines unstructured textual information in an attempt to discover structure and implicit meanings within the text.

One main problem in this area of research is regarding organization of document data. This can be achieved by developing nomenclature or topics to identify different documents. However, assigning topics to documents in a large collection manually can prove to be an arduous task. We propose a technique to automatically cluster these documents into the related topics. Clustering is the proven technique for document grouping and categorization based on the similarity between these documents [1]. Documents within one cluster have high similarity with each another, but low similarity with documents in other clusters.

Various techniques for accurate clustering have been proposed, e.g. K-MEAN [3, 8], CURE [11, 12], BIRCH [10], ROCK [1, 2], and many others [1], [3], [10], [11]. K-MEAN clustering algorithm is used to partition objects into clusters while minimizing sum of distance between objects and their nearest center. CURE (Clustering Using Representation) represents clusters by using multiple well scattered points called representatives. A constant number ‘c’ of well scattered points can be chosen from ‘2c’ scattered points for merging two clusters. CURE can detect clusters with non-spherical shapes and works well with outliers [11, 12]. BIRCH (Balance and Iterative Reducing and Clustering using

Hierarchies) is useful algorithm for data represented in vector space. It also works well with outliers like CURE [10]. However, the traditional clustering algorithms fail while dealing with categorical attributes. As they are based on distance measure so their merging processing is not accurate in case of categorical data. ROCK (Robust Clustering Algorithm for Categorical Attributes) gives better quality clusters involving categorical data as compared with other traditional algorithms. Below we first describe the original ROCK approach and then propose our own enhancements to ROCK which we call the Enhanced ROCK or EROCK approach.

2. ROCK ALGORITHM

ROCK is an agglomerative hierarchical clustering algorithm [2]. The original algorithm used the Jaccard coefficient for similarity measure but later on a new technique was introduced according to which two points are considered similar if they share a large enough number of neighbors. The basic steps of ROCK algorithm are:

1. Obtain a random sample of points from the data set ‘S’
2. Compute the link value for each pair of points using the Jaccard coefficient [2]:

$$Sim\ (T_1, T_2) = \frac{|T_1 \cap T_2|}{|T_1 \cup T_2|}$$

3. Maintain a heap (as a sparse matrix) for each cluster’s links
4. Perform an agglomerative hierarchical clustering on data using number of shared objects (as indicated by the Jaccard coefficient) as clustering criterion.
5. Assign the remaining points to the found cluster
6. Repeat steps 1-5 until the required number of clusters has been found.

ROCK algorithm has following advantages over other clustering algorithms [1]:

1. It works well for the categorical data.
2. Once a document has been added to a specific cluster, it will not be re-assigned to another cluster at the same level of hierarchy. In other words, document switching across the clusters is avoided using ROCK.
3. It uses the concept of links instead of using distance formula for measuring similarity resulting in more flexible clustering.
4. It generates better quality clusters than other algorithms.

Limitations of ROCK include the following:

1. ROCK algorithm used sparse matrix for storing cluster links.
2. Sparse matrix takes more space so efficiency suffers adversely.
3. Similarity is calculated by using Jaccard coefficient.
4. Similarity function is dependent on document length.

3. PROPOSED ENHANCEMENTS (EROCK)

EROCK approach includes several enhancements to overcome the limitations of the ROCK algorithm. Here we discuss these enhancements.

First, ROCK algorithm draws random sample from the database. It then calculates links between the points in the sample. The purposed approach (EROCK) makes use of entire data base for clustering. Every point in the database is treated as a separate cluster meaning that every document is treated as a cluster. Then the links between these clusters are calculated. The clusters with the highest number of links are then merged. This process goes on until the specified numbers of clusters are formed. So by decomposing the whole database, linkage and topic generation will become efficient.

Second, ROCK algorithm uses similarity measure based on Jaccard coefficient. We propose cosine measure:

$$CosSim\ (v_1, v_2) = \frac{|v_1 \,.\, v_2|}{|v_1|\,|v_2|}$$

where v_1 and v_2 are the term frequency vectors. v_1. v_2 is the vector dot product defined as:

$$\sum_{i=1}^{k} i = v_{1}i \;\; v_{2}i$$

and $|v_1|$ is defined as:

$$|v_1| = \sqrt{v_1 \,.\, v_2}$$

Cosine similarity is independent of the document length. Due to this property processing becomes efficient. Cosine similarity has advantages over Euclidean distance while applied on large documents (when documents tends of scale up), Euclidean will be preferred otherwise.

Third, ROCK uses sparse matrix for link information. The sparse matrix requires more space and long list of references because of which efficiency suffers adversely. In EROCK adjacency list instead of sparse matrix is proposed for maintaining link information between neighboring clusters. Adjacency list is a preferred data structure when data is large and sparse. Adjacency list keeps track of only neighboring documents and utilizes lesser space as compared to sparse matrix. Besides space efficiency it is easier to find all vertices adjacent to a given vertex in a list.

4. IMPLEMENTATION

4.1 Inputs

The EROCK algorithm requires some initial parameters which are necessary for the whole process. Following are the major inputs to run the algorithm:

- A directory containing text documents (Corpus).
- Threshold for number of clusters to be formed.
- Threshold value for measuring similarity of documents.
- Threshold value for taking top most frequent words for labeling the folders.

4.2 Document Clustering and Topic Generation Using EROCK Algorithm

Basic steps of EROCK are the same as those of ROCK. For document clustering and topic generation, the text files in the corpus are first converted into documents. Following are the steps involved in making the clusters, using EROCK algorithm:

1. Build documents from the text file present in the specified folder.
2. Compute links of every document with every other document using cosine similarity measure [2].
3. Maintain neighbors of each document in an adjacency list structure.
4. After computing links for all documents, each document is treated as a cluster.
5. Extract the best two clusters that will be merged to form one cluster. This decision is made on the basis of goodness measures. In EROCK, goodness measure defined as the two clusters which have maximum number of links between them [2]. Let these two clusters be u and v.
6. Now merge the two clusters u and v. Merging of two clusters involve, merging the names of the two clusters, the documents of two clusters and links of two clusters. This will result in a merged cluster called w.
7. For each cluster x that belongs to the link of w take following steps:

   i. Remove clusters *u* and *v* from the links of *x*.

ii. Calculate the link count for *w* with respect to *x*.

iii. Add cluster *w* to the link of *x*.

iv. Add cluster *x* to the link of *w*.

v. Update cluster *x* in the original cluster list.

vi. Add cluster *x* to the original cluster list

vii. Repeat step (iv.) until the required number of clusters are formed or there are no two clusters found to be merged.

viii. After obtaining the final merged cluster list apply labeling process on each. For labeling, the most frequent word from each document of a cluster is used. Take top most frequent words based on the threshold value.

The word with high frequency will be treated as the topic or label for a cluster. All related documents will be placed under one topic. Physically these documents will be put in folders with topics or label as folder name.

4.3 Output:

- A list of clusters labeled properly.
- Each cluster gets converted into a physical folder/directory on the disk and each folder contains the documents of the respective cluster.

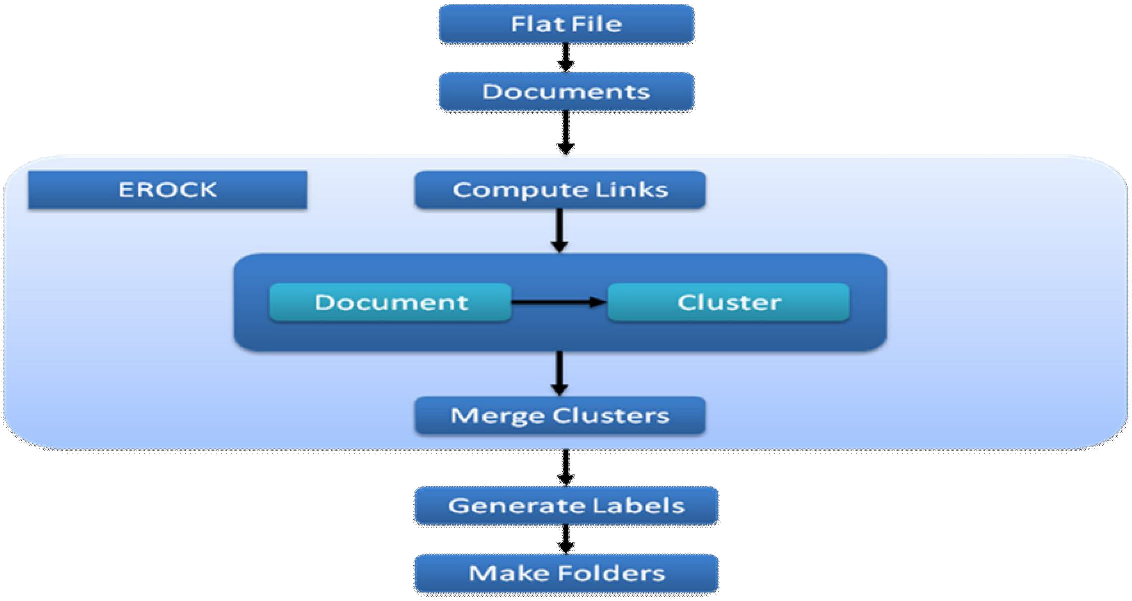

Figure 1: Application Flow

5. EXPERIMENTAL SETUP AND RESULTS

In order to evaluate the effectiveness of our algorithm, we compared the results obtained from ROCK algorithm with EROCK algorithm on similar text documents (corpus). We run both the algorithms on different corpus sizes i.e. 10, 20, 40, 50, 100, 200. For final algorithm comparison, the size of the corpus was four hundred (400) documents. Initially stop words and other useless items were removed from the documents in a pre-processing stage. The first step is to remove common words, for example, *a, the, or,* and *all* using a list of stop words. If a word in a document matches a word in the list, then the word will not be included as part of the query processing [18]. After the generation of intermediate form, clustering algorithm was applied on it. We report results from two types of analysis: Cluster Analysis and Label Analysis.

5.1 Cluster Analysis

We analyzed clustering results by varying the desired number of clusters from one (1) to ten (10). For any target number of clusters, similarity values (threshold) can have the range from 0.1 to 1.0. Figure 2 shows the overall results obtained from this study. According to the figure, clusters to be obtained are dependent on similarity threshold values. The inferences gained from the analysis are given as under:

- If the number of clusters to be obtained is equal to the number of documents then similarity factor has no effect on the clustering results.
- If the number of clusters to be obtained is less than actual number of documents, then the number of clusters to be obtained depends upon the similarity threshold.
- Increase in the threshold of top frequent word(s) of cluster will increase the size of the cluster label.
- For the dataset which we used for analysis, EROCK discovered almost pure clusters containing relevant documents with respect to their topics.

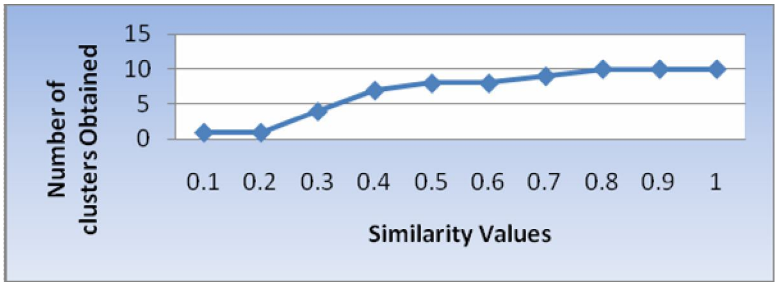

Figure 1: Clusters Obtained w.r.t Similarity values

Table 1 shows cluster labeling by similarity threshold values and number of clusters to be obtained. Here we use ten documents for summarization. If similarity value is 0.1 and number of clusters to be obtained is 1, then only one single label or topic will be generated and the entire document will be put under it. If similarity value is 0.2 and number of clusters to be obtained is 2 then two labels will be generated. If similarity value is 1.0 and numbers of clusters to be obtained are 10 then all the documents will be labeled separately as clusters. It means that labeling or document topics are mainly dependent on both similarity threshold values and number of clusters to be obtained. Other scenarios are very obvious from the table.

Similarity Threshold	Clusters to be Obtained	Folders (Comma Separated)
0.1	1	ALERTS ALARMS WORM AURORA COST DATA CLASSIFIERS HOST
0.2	2	ALERTS ALARMS WORM1 AURORA COST DATA CLASSIFIERS HOST, WORM
0.3	3	ALARMS, ALERTS, CLASSIFIERS, WORM AURORA COST DATA HOST
0.4	4	ALARMS, ALERTS, AURORA, CLASSIFIERS, DATA, HOST, WORM COST
0.5	5	ALARMS, ALERTS, AURORA, CLASSIFIERS, COST, DATA, HOST, WORM
0.6	6	ALARMS, ALERTS, AURORA, CLASSIFIERS, COST, DATA, HOST, WORM
0.7	7	ALARMS, ALERTS, AURORA, CLASSIFIERS, COST, DATA, HOST, WORM, WORM6
0.8	8	ALARMS, ALERTS, AURORA, CLASSIFIERS, COST, COST8, DATA, HOST, WORM, WORM7
0.9	9	ALARMS, ALERTS, AURORA, CLASSIFIERS, COST,COST8 DATA, HOST, WORM, WORM7
1.0	10	ALARMS, ALERTS, AURORA, CLASSIFIERS, COST, COST8, DATA, HOST, WORM, WORM7

Table 1: Cluster analysis results with topics

5.2 Labeling Analysis

Labeling analysis involve the analysis of labels generated for clusters based on similarity threshold values. This analysis is helpful to check whether the process is accurate or not. Label generation varies as per similarity vales as shown in Table 1. Following parameters were used for this analysis:

- Number of Documents: 10

- Number of Cluster to be obtained: 4
- Similarity Threshold: 0.3

Top Label Frequency Threshold	Clusters to be Obtained	Labels (Comma Separated)
0.3	4	ALARMS, ALERTS, CLASSIFIERS, WORM AURORA
0.5	4	ALARMS, ALERTS, CLASSIFIERS, WORM AURORA COST
0.7	4	ALARMS, ALERTS, CLASSIFIERS, WORM AURORA COST DATA
1.0	4	ALARMS, ALERTS, CLASSIFIERS, WORM AURORA COST DATA HOST

Table 2: Cluster Labeling w.r.t Similarity Value

5.3 Comparison of ROCK & EROCK Performance

It is noteworthy that comparison should be performed on same machine and under same environment. We used the same machine for comparisons of ROCK & EROCK. It is also necessary that algorithm should also be implemented in same technology and on same platform. For this we implemented ROCK & EROCK algorithm on same technology and platform.

Performance results of both algorithms are shown in Figure 2 & Figure 3 and with similarity threshold of 0.1 and 0.5 respectively. We compared the both algorithm with varying sizes of the document (we calculated the number of words in a document, after removal of stop words). In Figure 2 & Figure 3 we mentioned the document size (No.of Words) horizontally and time (in second) vertically. It is very clear from both the figures that when document size goes on increasing, EROCK give good performance as compared with ROCK algorithm.

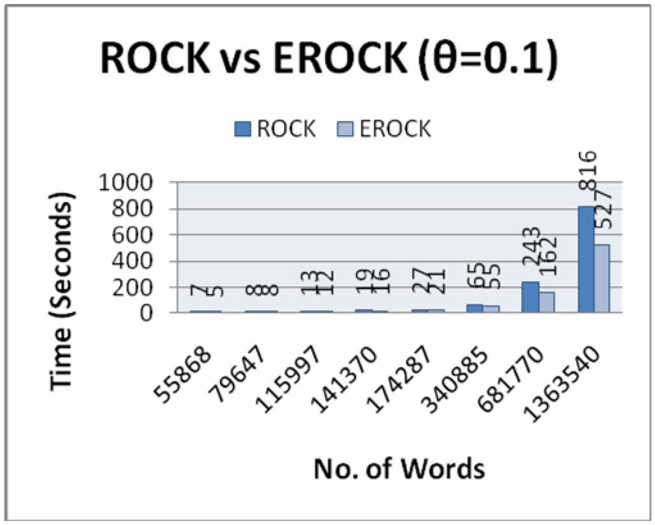

Figure 2: ROCK vs EROCK with SM=0.1

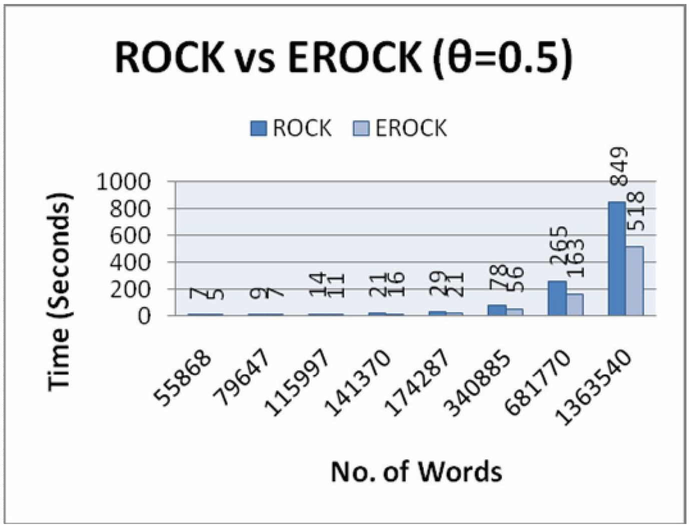

Figure 3: ROCK vs EROCK with SM=0.5

6. CONCLUSION & FUTURE WORK

In this paper we proposed an efficient way of document topic generation with enhanced version of cosine based similarity between the pair of categorical data known as clusters. We also proposed and used efficient document storage technique i.e. adjacency list instead of sparse matrix. By enhancing these two parameters of traditional ROCK algorithm, we get better results (as shown in Figure 2 & Figure 3). The experimental results obtained from the research are very encouraging. The outcome of this research shows that by using proposed approach, the cumbersome task of manually grouping and arranging files becomes very easy. Now user will be able to get relevant information easily without doing tedious manual activity. Huge information is now available in the form of text documents so documents/clusters having related information are grouped together and labeled accordingly. Clusters are merged only if closeness and inter connectivity of items within both clusters are of high significance. Finally it is observed that EROCK gives good performance for large datasets.

There are many areas in text mining; where one may carry on his/her work to enhance those areas. Out of these, the labeling of the clusters is a very daunting challenge of this time. No remarkable effort has been made in this regard to get good result. That is why automatic labeling of the clusters is not so much accurate. A keen and concerted work has been done to remove this hurdle. It will certainly serve as a lime length for future researchers.

7. REFERENCES

[1] Shaoxu Song and Chunping Li, “Improved ROCK for Text Clustering Using Asymmetric Proximity”, SOFSEM 2006, LNCS 3831, pp. 501–510, 2006.

[2] Sudipto Guha, Rajeev Rastogi and Kyuseok Shim, “ROCK: A robust clustering algorithm for categorical attributes”. In: IEEE Internat. Conf. Data Engineering, Sydney, March 1999.

[3] Tapas Kanungo, David M. Mount, Nathan S. Netanyahu, Christine D. Piatko, Ruth Silverman, Angela Y. Wu, “An Efficient k-Means Clustering Algorithm: Analysis and Implementation”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 24, No. 7, July 2002.

[4] Alain Lelu, Martine Cadot, Pascal Cuxac, “Document stream clustering: experimenting an incremental algorithm and AR-based tools for highlighting dynamic trends.”, International Workshop on Webometrics, Informatics and Scientometrics & Seventh COLIENT Meeting, France, 2006.

[5] Jiyeon Choo, Rachsuda Jiamthapthaksin, Chun-sheng Chen, Oner Ulvi Celepcikay, Christian Giusti, and Christoph F. Eick, "MOSAIC: A proximity graph approach for agglomerative clustering," Proceedings 9th International Conference on Data Warehousing and Knowledge Discovery (DaWaK), Regensbug Germany, September 2007.

[6] S. Salvador, P. Chan, Determining the number of clusters/segments in hierarchical clustering/segmentation algorithms, Proceedings of the 16th IEE International Conference on Tools with AI, 2004, pp. 576–584.

[7] Murtagh, F., “A Survey of Recent Advances in Hierarchical Clustering Algorithms”, The Computer Journal, 1983.

[8] Huang, Z. (1998). Extensions to the K-means Algorithm for Clustering Large Datasets with Categorical Values. Data Mining and Knowledge Discovery, 2, p. 283-304.

[9] Huidong Jin , Man-Leung Wong , K. -S. Leung, “Scalable Model-Based Clustering for Large Databases Based on Data Summarization”, IEEE Transactions on Pattern Analysis and Machine Intelligence, v.27 n.11, p.1710-1719, November 2005.

[10] Tian Zhang, Raghu Ramakrishan, Miron Livny, “BIRCH: An Efficent Data Clustering Method for Very Large Databases”.

[11] Linas Baltruns, Juozas Gordevicius, “Implementation of CURE Clustering Algorithm”, February 1, 2005.

[12] Sudipto Guha, Rajeev Rastogi, Kyuseok Shim, “CURE: An Efficient Clustering Algorithm for Large Databases”.

[13] M. Castellano, G. Mastronardi, A. Aprile, and G. Tarricone,”A Web Text Mining Flexible Architecture”, World Academy of Science, Engineering and Technology 32 2007.

[14] Brigitte Mathiak and Silke Eckstein,” Five Steps to Text Mining in Biomedical Literature”, Proceedings of the Second European Workshop on Data Mining and Text Mining in Bioinformatics

[15] Ng, R.T. and Han, J. 1994. Efficient and effective clustering methods for spatial data mining. Proceedings of the 20th VLDB Conference, Santiago, Chile, pp. 144–155.

[16] Stan Salvador and Philip Chan, Determining the Number of Clusters/Segments in Hierarchical Clustering/Segmentation Algorithms, Proc. 16th IEEE Intl. Conf. on Tools with AI, pp. 576–584, 2004.

[17] Sholom Weiss, Brian White, Chid Apte,” Lightweight Document Clustering”, IBM Research Report RC-21684.

[18] Masrah Azrifah Azmi Murad, Trevor Martin,”Similarity-Based Estimation for Document Summarization using Fuzzy Sets”, IJCSS, Volume (1): Issue (4), pp 1-12.

[19] Sohil Dineshkumar Pandya, Paresh V Virparia, “Testing Various Similarity Metrics and their Permutations with Clustering Approach in Context Free Data Cleaning”, IJCSS, Volume (3): Issue (5), pp 344-350.