Document Topic Generation in Text Mining by Using Cluster Analysis With Enhanced ROCK (EROCK)

By
Rizwan Ahmad
[2006-NUST-MS PhD-CSE (E)-04]

Submitted to the Department of Computer Engineering in partial fulfillment of the requirements for the degree of

Master of Science
in
Computer Software Engineering

Advisor
Dr Aasia Khanum

College of Electrical & Mechanical Engineering
National University of Sciences and Technology
2010

بسم الله الرحمن الرحيم

In the name of ALLAH, the most Beneficent, the most Merciful.

ABSTRACT

Clustering is a useful technique in the field of textual data mining. Cluster analysis divides objects into meaningful groups called clusters based on information and relationship between objects. Bunch of material is available related to any topic from internet by just one click. It becomes tedious on user's end to differentiate between data and really required information. This task is very hard as it has to be done manually. This project will explain how to cope with this problem to effectively facilitate the user. We used ROCK algorithm with some modifications. ROCK generates better clusters than other clustering algorithms for data with categorical attributes. We used cosine measure to know the similarity between two documents. Furthermore, we used adjacency list instead of sparse matrix to store the document. The evaluation of algorithm has been done on text documents. Due to these enhancements it is named as Enhanced ROCK or EROCK. These changes affect the time space complexity of the algorithm.

Experimental results on standard test documents show the outcomes of the EROCK algorithm. Similarity threshold, number of clusters to be obtained and text documents (corpus) are the main parameters used for EROCK evaluation.

JAVA with jdk1.6.0 has been used for implementation of the EROCK. NetBeans IDE 6.5.1 has been used as a development editor. Experiments have been carried out on a variety of standard text documents with specific approach.

Dedications

To all those who believe in ….

“No one gets more than what he strives for”

(Al Quran)

ACKNOWLEDGEMENTS

All praise is to the omnipotent and magnanimous Almighty Allah, Who guides us from darkness to the light and helps us in difficulties; Allah enabled me to complete this tough work and bestowed the wisdom and strength to make material contribution to an already existing ocean of knowledge. I offer my humble thanks to Holy Prophet Muhammad (peace be upon him), who is eternal a source of knowledge and guidance for humanity.

It is not a mere formality when I express my full gratitude, high esteem and admiration for my most worthy supervisor Dr. Aasia Khanum. Apart from her skilled supervision and invaluable help in the project work, I shall always be indebted to her for extending all possible help and supervision as and when I was in dire need. At the same time I equally express my respect and regards to the all the committee, members for their time and diligence, with whom I had quite a few good discussions. Their profound inquisitiveness sharpened my concepts and research.

I express special thanks to my friends, especially Mr. Muhammad Ismail and Mr. Faqeer Ali for providing sincere and timely assistance in the hours of complete difficulty. I really want to salute Mr. Faqeer Ali for his continuous support and unconditional help to me during not only my project but also in my office work.

Finally, I want to thank my beloved parents and sister for their consistent moral support and encouragement. Their urge and motivation is commendable. I specially want to thank my father Prof. (R) F.D Arif for his continuous support, love, encouragement and guidance.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

Chapter 1

Introduction

Great focus has been there on structural data such as transactional, relational and data warehouse. Experts are keenly interested in these components. However, in practice, a considerably large portion of information present today is in the form of text data bases or document databases. These text documents are multiplying with the rapid increase of electronic format. Internet, hurridly, provides a number of pages / papers on a certain subject as and when we softly click on it. Thus a great treasure of data on the topic is revealed in a few moments. User will not be able to organize such a treasure efficiently so there should be a mechanism to structure such a treasure.

1.1 Importance of Text Mining

Due to large volume of data, the importance of data/text mining and knowledge discovery is increasing in different areas like: telecommunication, credit card services, sales and marketing etc [1]. To structure large amount of text data in any organization, different text clustering techniques can be used [2]. Text mining is a growing field that is used to gather meaningful information from text. Text mining can be used for different tasks in different organization. Text Mining uses unstructured textual information and examines it in attempt to discover structure and implicit meanings which are there within the text. The general framework of text mining process consists of two main phases: preprocessing phase and discovery phase.

Clustering has been the proven and the best technique for document grouping based on the similarity between these documents [1]. Document clustering is used to distinguish among different document for their categorization. Documents within one cluster have high similarity with each another, but will be different from documents in other cluster. Distinctions between pair of documents are calculated by similarity measure, goodness measure and criterion function. As the data in the documents is well structural text data, special techniques for accurate clustering are required: For example, ROCK [1, 2], KMEAN [3. 8], BIRCH [10], CURE [11, 12] and many other algorithms can be used.

1.2 Text Mining & the Researcher

The main problem in this area of research, which researchers are facing, is regarding organization of data. The organized data should be in meaningful structure. This can be achieved by developing nomenclature. We have a number of research papers available in document format. If we want to find out the topics of each research paper, it will be an arduous task to do it manually. So we shall use a technique to cluster these documents into the related topics. Thus it may become easy for the researchers to find related information. Hence the first and foremost effort should be the exact grouping of papers under relevant topic(s).

Researchers have done lot of efforts to organize huge volume of data in the field of (hyper)text mining [1]. Different clustering methods have been developed and discussed to deal with problems faced during text clustering [3]. There are different tools for textual data mining, used to tune data into useful knowledge. Some useful algorithms are discussed in [1], [3], [10] , and [11].

1.3 Benefit of Document Topic Generation

Document topic generation is very useful to get relevant information with in short time. There are lots of benefits of this technique. It is very much helpful when doing grouping of similar large number of documents based on their topics. It become very difficult when doing the same task manually. Sometimes manual grouping becomes almost impossible.

Cluster analysis is used to divide objects into meaningful groups. These groups or subsets are called clusters. Clusters are obtained based on information and relationship between objects of the documents. So the clusters with maximum similarities will be merged. It is very much helpful for the exact grouping of papers under relevant topic.

So automatic document topic generation and classification (grouping) is very helpful for the end user. It saves the time which is spent in manual document grouping. So this work will elaborate the process of automatic labeling of the document.

1.4 Technology Used

This research has been completely implemented in JAVA programming language. Development kit used is JDK1.6.0 with development editor NetBeans IDE1.6.5. Language has been selected due to its portability and ease of use for such kind of problems. Standardized libraries provided generic way to access host specific features. JAVA has automatic garbage collector to manage memory in the object lifecycle. And it also fits perfectly in the necessities of an analytical processing research due to its inherent characteristics. It is also good for real time application development.

1.5 Scope of Thesis

This work proposes a novel algorithm which is based on simple general parameters which are being used in the field of text mining & clustering for many different purposes. Its strength lies in the correct combination of these parameters which outshines many previously used techniques. There are many algorithms in the market for this purpose. Some are good and some have problems like calculation of similarity measure between pair of documents by using distance measure.

Our focus is on ROCK [1, 2], which belongs to the family of agglomerative hierarchical algorithm. Modified form of ROCK[1, 2] has been used and named as Enhanced ROCK or EROCK. EROCK has been used to cluster the algorithms and label them appropriately. EROCK has been evaluated with test documents and results are presented.

1.6 Thesis Outline

Chapter 1 provided the introduction of text mining and clustering analysis. It discussed the evolutionary path of text mining technology and importance of its applications. Chapter 2 is the literature review and we will discuss text mining and clustering in detail. The chapter presented a general classification of text mining and clustering analysis tasks. We also discussed the basic life cycle of text mining. In chapter one we discussed the cluster analysis in detail with different cluster analysis techniques, technology used for the implementation EROCK and scope of thesis had been discussed.

Chapter 3 will introduce the different clustering algorithms which are used for cluster analysis. Chapter 4 will gives a detailed overview of the clustering algorithm with different important terminologies used for these algorithms. In Chapter 5 we will discuss the implementation methodology for the proposed approach that is EROCK. Chapter 6 presents the experimental results on standard test documents against the recommended metrics and shows the results achieved by the proposed scheme. Chapter 7 narrates conclusion and future work.

1.7 Summary

This chapter illustrated the need for efficient algorithm for text mining. It also gave an overview of text mining & clustering fundamentals and its different techniques and methods. Text mining terminology and complete life cycle had been discussed. Benefits and use of clustering algorithm had also been discussed. Many clustering techniques i.e. hierarchical clustering, partitional clustering, complete clustering, fuzzy clustering etc were introduced in this chapter. Finally, scope and structure of the thesis was outlined. We also discussed about the technology used for the implementation of EROCK algorithm which will be used for clustering.

Chapter 2

Literature Review

2.1 Text Mining

Text mining is a growing field that is used to collect meaningful information from text. It is also known as text data mining. Basically it is derived from the field of data mining. As now-a-days electronic files, like text files, PDF files and doc files etc are growing dramatically, there should be some kind of mechanism to get most of them. The mechanism and technique used is known as text mining. Text mining refers to the process of deriving information from text. Conceptually, text mining is the process of structuring the input text, deriving patterns & interpretation of the output. We can say that text mining is the process of analyzing text to extract meaningful information that is useful for a particular task. There are different tasks of text mining. Some of them are:

1. Text Categorization.
2. Text Clustering.
3. Text Analysis.
4. Document Summarization.

Naturally, the goals of text mining are similar to those of data mining; for example, it also attempts to discover clusters, trends, associations, and deviations in a large set of texts. So the main emphasis of text mining is the clustering and arranging of similar documents after applying some suitable clustering algorithm. Text mining has also adopted techniques and methods of data mining, e.g., statistical techniques and machine learning approaches [6]. Text Mining uses unstructured textual information and examines it in attempt to discover structure and implicit meanings which are there within the text. There are different text mining techniques and approaches which are used to label the documents. Documents can be distinguished by doing the labeling for each document. These labels may be keywords extracted from the document or just a list of words within the document of interest. These can be based on term or keyword frequencies.

The general framework of text mining process consists of two main phases: ***preprocessing phase*** and ***discovery phase*** as shown in Figure 2.1. Preprocessing stage is very important in cluster analysis, at the preprocessing stage, the free-form texts are transformed into some kind of semi-structured representation that allows their analysis. Preprocessing stage can included the tasks of stop word removal and other useless items like alphanumeric characters or punctuations etc.

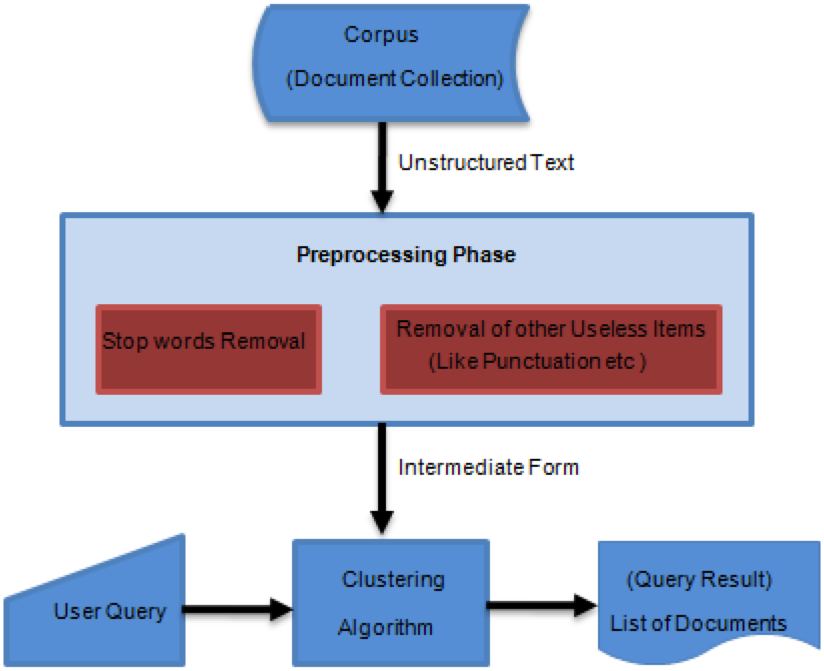
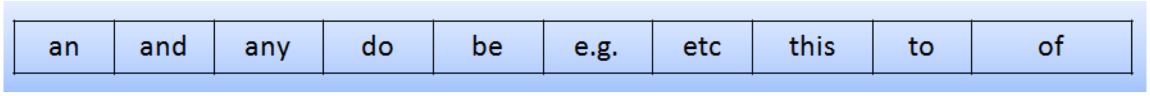


Figure 2.1: Text Mining with Preprocessing

Corpus is the collection of document for statistical analysis. In our case it consists of different documents whose topics will be generated for the ease of end user. After preprocessing stage, specific algorithm will be applied to the intermediate form for the generation of clusters and document labeling. Finally documents can be placed under specific label after clustering.

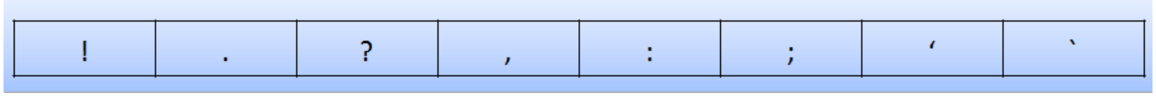
Stop word removal include those words that do not carry any useful information and hence are ignored during indexing and searching. Some of the stop words are shown in Figure 2.2.



an	and	any	do	be	e.g.	etc	this	to	of

Figure 2.1: Stop Words

Other useless items can also be removed from the text for accurate analysis & cluster generation. These useless items can be alpha-numeric words or punctuations as shown in Figure 2.3.



!	.	?	,	:	;	‘	`

Figure 2.2: Punctuations

Words and punctuations shown in Figure 2.2 and Figure 2.3 play no role in clustering so must be removed from the text. After removal of these useless items we get intermediate form which will be used in further processing.

2.2 Text Mining Life Cycle

Text mining life cycle or architecture consists of different steps. These steps are helpful to implement text mining. The process consists of the following phases [13, 14]:

2.2.1 Text Pre-Processing

Text Refining or Text Preprocessing phase is used to transform input text in to structured information. It is important for the exact analysis of relations between words, as it is needed in the extraction of relations between documents. Text Refining input is unstructured text data. The datum is provided to a suitable process that consists of:

1. Documents recovery.
2. Cleansing of the text documents.
3. Extraction of the useful information to the text mining phase.

Text refining output is stored in a structured form that is called Intermediate Form. Text Mining techniques are applied to the Intermediate Form. These techniques can also be very different. Text mining process doesn’t consider only the presence or the frequency of a word or a concept inside a document but also aims at finding the relationship between these words or concepts and other inside documents. In this way its purpose is to find information contained in text.

2.2.2 Document Clustering

Document clustering is the process of entity assignment to a few categories like classes and groups which were not previously defined. The purpose of this activity is to collect similarities among them. It is necessary to divide them on the basis of classification, the measure of the proximities among the entities, the number of groups or clusters and finally to pass them in integration phase.

The clustering techniques are distinguished on the basis of their type like hierarchical or non hierarchical type. Hierarchical analysis is further divided by agglomerative and divisive techniques. Non hierarchical analysis is like the possibility of subdivision in mutually exclusive classes (partitions) or subdivision in overlapped classes or clusters [17]. In such type of analysis the fundamental concept is to plan them on the basis of number of cluster to be produced and to get solutions which predict a varying number of clusters. Textual clustering is used as a process to divide a document collection in groups known as clusters. Inside clusters, documents similarities are measured on the basis of selected characteristics like keywords, term frequencies etc which are common or similar among them.

Textual clustering can be used to remove the extra words or characters from the contents of documents collection or corpus for identifying similarities between documents to find correlated or similar information. It is helpful for labeling the clusters accordingly. If it works with keywords or with features that represents the semantics of the documents, the individualized clusters will be distinguished on the basis of different themes treated in the corpus. Algorithms of hierarchical clustering are used for textual data. The algorithms used are known as ROCK [1, 2], KMEAN [3, 8], BIRCH [10] and CURE [11, 12].

2.2.3 Document Categorization

In Document categorization, the documents are assigned to one or more categories or clusters. This technique of mining is known as Text Categorization (TC). Text Categorization is often developed through algorithms of Machine Learning.

Machine Learning techniques are used to do some AI related activities. Classification is the process in which meaningful correlations among frequent data are found. This is known as term based frequencies. This is a typical process of Data Mining. There are association rules for Text Categorization.

All algorithms which are used to produce association among documents or clusters have two phases. These two phases are keywords collection from document collection and the extraction of associations. First phase is the collection of frequent terms. After getting frequent terms and keywords from whole dataset, all the association rules are calculated that can be derived from the produced frequent set and it satisfies the given threshold level. After this process, documents are divided in different categories.

2.2.4 Pattern Extraction

In Pattern Extraction some patterns are identified. These patterns are generated by following the analysis of associations. The discovery of the associations is the process in which meaningful correlations among frequent whole data are found that are collected from document collection. So the pattern extraction will only be performed on structured data instead of unstructured one.

Pattern Extraction is the base line in text mining and needs attention in Text Mining process. It is carried out through following techniques:

1. **Predictive Text Mining** which is used for identifying the tendencies in collected documents in a time period.
2. **Association Analysis** which identifies the relationships among the attributes, like: if the presence of a pattern implicates the presence of another pattern in documents.

2.2.5 Presentation of Results

Finally, presented techniques are used to introduce the patterns and to visualize the results. These results are helpful in examining the effects of applying a particular technique on collected documents. These results can be based on the following parameters for the judgment of algorithm:

1. Number of clusters to be obtained.
2. Similarity functions.
3. Criterion function.
4. Goodness Measure.
5. Neighbors and link analysis.

The whole life cycle is depicted in Figure: 2.4. The figure shows the overall process of text mining, which starts from Text Pre-Processing step and ends at Result generation step.

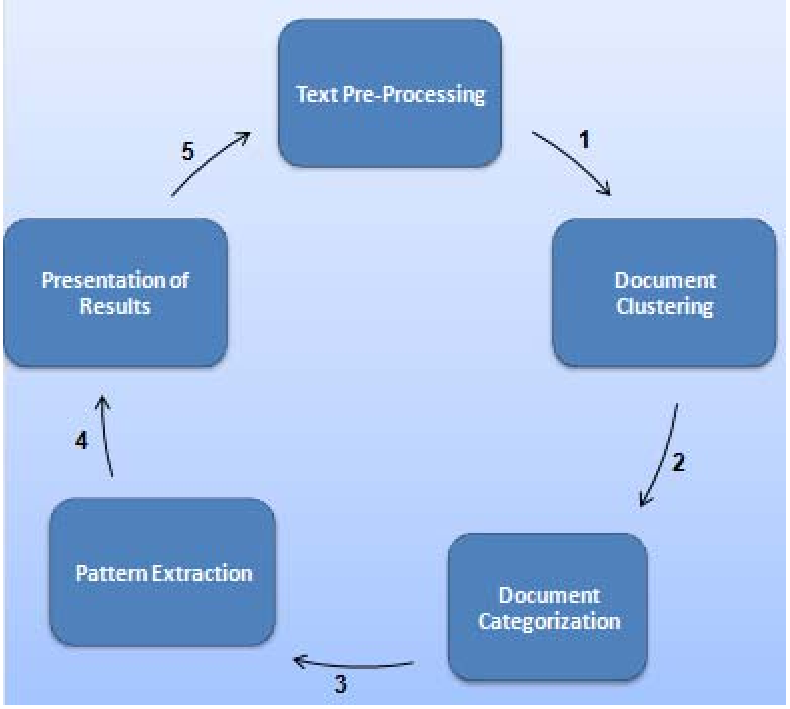


Figure 2.3: Text Mining Life Cycle

2.3 Clustering

Clustering in text mining means a group of documents having features and attributes, which are more similar to each other than to the attributes of any other group. In other words, documents from one cluster share some common features, which distinguish them from the other documents. Technically it is known as inter-document similarities and intra document similarities. So clustering is used to distinguish among different document for their categorization. For this different clustering techniques and methods are used. Distinctions between pair of documents are calculated by similarity measure function [16].

2.3.1 Types of Clustering

There are different techniques of clustering text documents which are helpful in document categorization. Clustering techniques are separated on the basis of their parameters to generate clusters. Different types of clustering which are found in literature are as follow: -

2.3.1.1 Hierarchical Clustering

A hierarchical clustering is the set of nested clusters that are organized as a tree. The root of the tree is a cluster containing all the objects. Node of the tree represents a cluster while children of a node represent sub-cluster. Often leaves of the tree are single clusters of individual data objects. Due to this nature and shape it is named as hierarchical clustering. This technique is helpful when clusters are formed like tree.

2.3.1.2 Partitional Clustering

A partitional clustering is the division of set of data objects into non-overlapping subsets known as clusters, such as each data object is exactly in one subset or cluster. Clusters are partitioned on the basis of their patterns and term frequencies. Clusters are partitioned in small chunks called sub-clusters. These sub-clusters are then grouped together. The basic theme is that it starts with an initial partition and move objects so that criterion function improves up to certain level.

2.3.1.3 Exclusive Clustering

In exclusive clustering each object is assigned to a single cluster. So there is no conflict among objects. Each object is placed in distinct groups or clusters. In this case there is no partitioning of cluster, single whole cluster is used.

2.3.1.4 Overlapping or Non-Exclusive Clustering

When an object or point is placed in more than one cluster, then clustering is known as overlapping clustering. So it reflects that an object can simultaneously belong to more than one group.

2.3.1.5 Fuzzy Clustering

In this clustering, every object belongs to every cluster with a membership weight between 0 (does not belong to the cluster) and 1 (belongs to the cluster), so clusters are treated as fuzzy sets. So similar objects belong to the same clusters and dissimilar objects belong to different clusters. Fuzzy clustering is the best for data analysis, pattern recognition and image segmentation. The fuzzy clustering algorithms include:

1. Fuzzy C-Shell algorithm.
2. Fuzzy C-means algorithm.
3. Fuzzy C-Ring algorithm.
4. Fuzzy C-Verities algorithm.

2.3.1.6 Complete Clustering

Complete clustering assigns every object to a cluster. It is used to find specific objects from a dataset which are tightly related by a common theme.

2.3.1.7 Partial Clustering

Partial clustering does not assign every object to a cluster. Partial clustering is useful when some objects in a dataset may not belong to well-defined groups.

2.3.2 Use of Clustering

Clustering can be useful in many ways. First of all, it is useful in providing an overview of the contents of a corpus documents that means it can predict the document themes stored in the corpus. The next task is identification of basic structures within groups of objects or clusters. So the use of clustering is helpful in finding related information. Trend analysis can also be done by using clustering and after applying different clustering techniques. So discovery of the documents become easy. Contents of the documents can be used for further classification. Finally duplicate documents in a collection can be detected on the basis document contents.

2.3.3 Goal of Clustering

Clusters are not just collection of entities with numerical similarity. Clusters are not merely a document it consists of groups of objects. This object collectively represents a concept. So clustering techniques not only produce clusters but it is also used to describe the relationship among concepts. Clustering does not need any predefined categories in order to group the documents. Thus, the main aim of cluster analysis is to produce a set of clusters in which the internal document similarities are maximized and the external similarities are minimized. By doing this, we can easily determine the grouping among clusters in set of unlabeled data. So it is also useful in structuring the document. Similarity measure produces the best clusters which are based on similarity threshold values.

In our case we consider a document database (textual database) containing some technical terms per document. These data can be used to cluster the documents such as documents with similar referring patterns (terms/frequencies) are in a single cluster. The clusters then can be used to characterize the different documents groups, and these characterizations can be used in other data mining tasks, such as relevance analysis, classification etc.

2.3.4 Problems with Clustering

Clustering is a very useful technique in text mining. But still there are numbers of problems with clustering. Among them are: -

1. Some of the clustering techniques do not address all the requirements sufficiently e.g. KMEAN [3, 8] does not give best results with categorical data.
2. Dealing with large number of dataset objects can be problematic because of time complexity.

3. For distance based clustering, the effectiveness of the method depends on the definition of distance between the objects; if good distance measure is not possible then we must define it, which is cumbersome.
4. The result of the clustering algorithm can be evaluated in different ways depending on the requirement, dataset and clustering method used.

2.4 Cluster Analysis

Cluster analysis classifies some observation (documents in our case) into two or more mutually exclusive *unknown* groups known as clusters. The basic purpose of cluster analysis is to convert observations into related groups or clusters. These members of the groups or cluster, known as objects, can share properties in common. Thus cluster analysis is also used to divide data object or entities into meaningful & useful groups. These groups are formed on the bases of information found in the data that describes the objects and the relationship between them. So the goal of cluster analysis is to measure the similarity between objects located in different groups. Greater the similarity within group and greater the difference between groups better the clustering results.

2.4.1 Simple Example of Cluster Analysis

Clustering is useful in forming clustering depending on different parameters. Once clusters have been formed we can present the final results. We can depict these results in the form of frequency polygon or in any other form. These graphs can provide better and clear picture of the groupings.

Following is the data from the student homework assignments marking. Different students have been given different marks according to their performance. There are total ten students picked on the basis of their grade or marks.

Suleman	Jabar	Saad	Imran	Shazad	Irfan	Noman	Rafique	Assad	Omer
11	11	13	18	18	21	23	23	25	32

Table 2.1: Home Work Assignment

For grouping purpose we will apply the clustering on these students. This grouping will help us to analyse the overall performance of the group of students. Students having close marks with respect to each other will be placed in one group (same group). Figure 2.5 shows the grouping results for the students. Two parameters have been used, relative frequency and the score. Relative frequency shows the number of students having similar or same marks for example Suleman and Jabar both obtained eleven (11) marks. So the relative frequency will be 0.2 as depicted in Figure 2.5. As there is only one student having thirty two (32) marks so this student (Omer) is different from others and depicted differently from all other students (Runt).

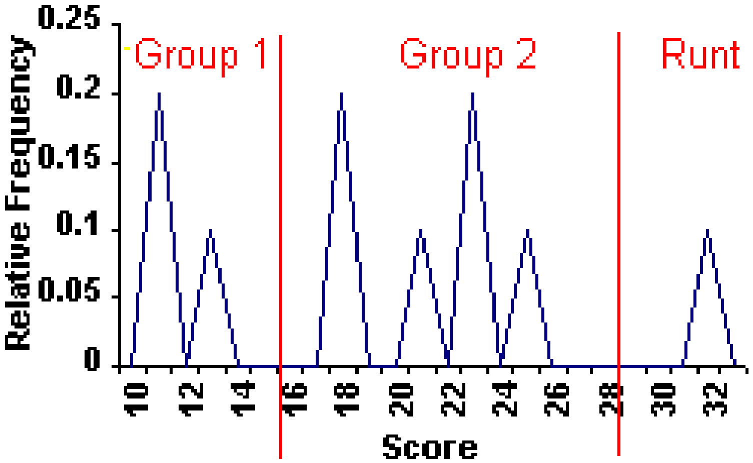


Figure 2.4: Cluster Visualization

2.4.2 Steps of Cluster Analysis

For clustering analysis, the common way to do this task is first to create a table of similarities or differences between all objects. This table is helpful in determining the closeness of the clusters. The second step is to use the gathered information to combine the objects into groups or clusters.

The table of similarities is known as proximities matrix. The entries in the cell of proximities matrix shows the similarities between the objects. It can be used in different ways to measure the similarities between different objects. After the creation of proximities matrix the next step is the combination of objects into groups. This process can be achieved through clustering algorithms. The basic idea is to combine objects that are similar to one another into separate groups of interest.

2.4.2.1 The Proximities Matrix

The analysis of different clusters starts with a data matrix. In data matrix objects are rows (r) and observations are columns (c). To achieve this, a table is created with rows and columns. In data matrix both rows and columns represents the data. The values or numbers in the table are the similarity measures or difference between the data or observations.

For example, Table 2.2 shows the data matrix with rows (O1, O2, O3, O4) and columns (X1, X2, X3, X4, X5). The similarity or difference values will be placed in the intersection of rows and columns, called cell. In Table 2.2 column represents the data and rows represent the observations.

OBSERVATIONS

		X1	X2	X3	X4	X5
OBJECTS	**O1**					
	O2					
	O3					
	O4					

Table 2.1: Data Matrix

A proximities matrix would appear as shown in Table: 2.3 with rows (O1, O2, O3) and columns (O1, O2, O3, O4) and corresponding similarity values will be placed in the cell. Here both rows and columns represent the observations instead of data. This table can be generated on the basis of data matrix.

OBSERVATIONS

		O1	O2	O3	O4
OBSERVATIONS	O1				
	O2				
	O3				
	O4				

Table 2.2: Proximities Matrix

The difference between a proximities matrix in cluster analysis and a correlation matrix is that a correlation matrix contains similarities between variables (X1, X2) while the proximities matrix contains similarities between observations (O1, O2).

To overcome the problem of how to combine multiple measures into a single number, the similarity or difference between the two observations, two techniques are used. These techniques are known as ***univariate*** and ***multivariate*** cluster analysis.

Univariate cluster analysis groups are based on a single measure or variable at a time. So it is used to analyse the data on single variable. ***Multivariate*** cluster analysis is based on multiple measures. So it is used to analyse the data on the basis of multiple variables. There are different types of variables used for this analysis, these are, Nominal variables, Ordinal variables, Interval variables.

2.4.2.2 Grouping Objects using Distance

After the calculation of distances between objects and derivation of data and proximity matrix, the next step in the cluster analysis process is to breakdown the objects into clusters. These groups are based on the distances. Different options and methods are available to do this.

1. **Single Linkage**: This technique is used to calculate the distance between two sub-groups. The basic theme is the minimum distance between any two members of opposite groups or cluster.

2. **Complete Linkage**: It is used to measure the distance between subgroups in each step as the maximum distance between them. The two members of the different groups can be used for this calculation.
3. **Average Linkage**: It is used to get the distance between subgroups at each step as the average of the distances between them.

2.5 Summary

This chapter illustrates the need for efficient algorithm for text mining. It also gives an overview of text mining and clustering fundamentals with different techniques and methods. Text mining terminology and complete life cycle is discussed. Benefits and use of clustering algorithm are explained. Many clustering techniques i.e. hierarchical clustering, partitional clustering, complete clustering, fuzzy clustering etc have been introduced in this chapter. Finally, scope and structure of the thesis is also outlined. The technology used for the implementation of EROCK algorithm for clustering is described in easily comprehensive manner. Thus the readers are facilitated as far as possible to understand efficient algorithm for text mining.

Chapter 3

Prominent Clustering Algorithms

3.1 Clustering Algorithm

We apply clustering algorithms on the resulting dataset, technically known as "Corpus' in text mining field, in order to discover structure within a document collection. Many algorithms have been proposed depending on the data collection and the task to be accomplished. Hierarchical Clustering and the Binary Relational Clustering are of the well known ones. In the former approach, clusters are arranged in cluster trees, where related clusters appear in the same branch of the tree, while the latter creates a flat cluster structure.

3.2 Clustering Methods

There are different algorithms which can be used for clustering. Each of them has its own pros and cons according to their use in different scenarios. Following are the most commonly used clustering methods / algorithms:

3.2.1 KMEAN

KMEAN clustering algorithm is used to divide objects into clusters while minimizing sum of distance, distance between objects and their nearest mean (center). KMEAN algorithm is an iterative scheme. KMEAN is very useful algorithm for statistical analysis. KMEAN uses the partitional clustering technique to create many small clusters. To find the centroid different steps have been performed on these clusters. The basic steps for K-MEAN algorithm are as follow:-

1. K points have been selected as an initial centroids.
2. All the points are assigned to closest centroid.
3. Centroid for each cluster is recalculated.
4. Repeat step 2 & 3 until centroid do not change.

Figure: 3.1 show the complete flow chart of K-MEAN algorithm. It starts by selecting number of cluster K. In second step initial centroid is selected and distance of an object is calculated from the centroid. Objects are grouped on the basis of minimum distance from centroid. After the task of grouping, new centroid is again calculated and new centroid is chosen. The process continues until centroid does not change any more, means it remain the same for new group as well. This condition will stop the recalculation of new centroid and algorithm reached the end [3, 8].

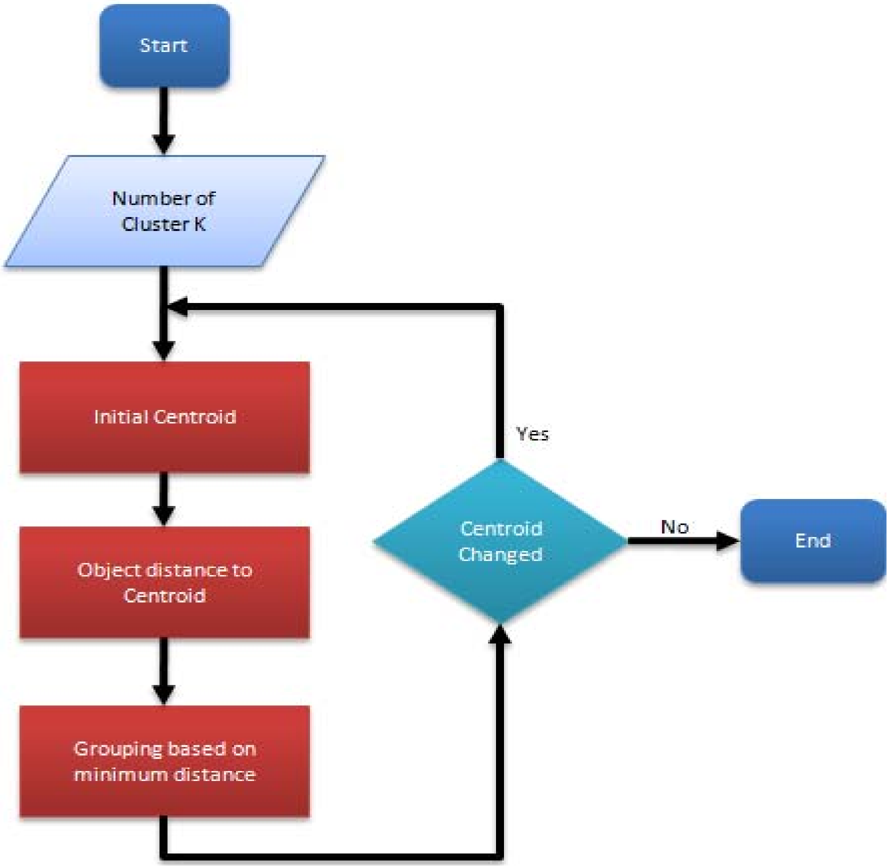


Figure 3.1: K-Mean Flow Chart

3.3.2 CURE

CURE (Clustering Using Representation) lies between centroid base and all point approach. It represents clusters by using multiple well scattered points called representative. A constant number 'c' of well scattered points can be chosen from '2c' scattered points for two merged clusters. It can detect clusters with non-spherical shape.

It works well when with outliers. So CURE represent clusters by fix number of representative objects or point and have the tendency to move towards cenroid.

The running time of CURE is O(n2 log n) and space complexity is O(n). The algorithm cannot be applied to the direct large dataset. To apply it random sampling can be selected from large dataset. The selected random sample space is further partitioned. CURE algorithm can be presented as follows:-

1. For every cluster, the mean of the points in the cluster and a set of c representative points of the cluster have been chosen.
2. Initially c = 1.
3. Closest clusters have been chosen.
4. All the input points are inserted into a tree T
5. Each input point is treated as separate cluster.
6. Distance for each cluster from centroid is chosen and closest one is inserted into the heap with increasing order with respect to the distance from centroid.
7. Remove the top element of heap and merge it with its closest cluster and calculate the new representative points for the merged cluster.
8. Remove old cluster (clusters before merging) from tree and heap.
9. Also for all the clusters in the heap, update closest cluster by calculating its distance from centroid and re-allocate it.
10. Insert the newly calculated closest cluster in the heap.
11. Repeat steps 7 to 10 till heap > number of point.

Figure 3.2 depicts the overview of CURE algorithm. The sequence in the diagram shows that CURE algorithm draws random sample from data or corpus. As CURE cannot directly apply to the data. This sample will further divided into more partitions or subsets. Clusters are made from partitioning sample. After cluster generation next step is the elimination of the outliners with spherical shapes. After applying outliners, these partial clusters are further divided into suitable cluster. The final step is the labeling of generated cluster for further processing.

So from the figure it is obvious that CURE algorithm works well outliners. Also it cannot be applied on direct data. So random samples are drawn from the dataset and partitioned accordingly for further processing to achieve the desired results. This process continues till sufficient clusters are obtained [11, 12].

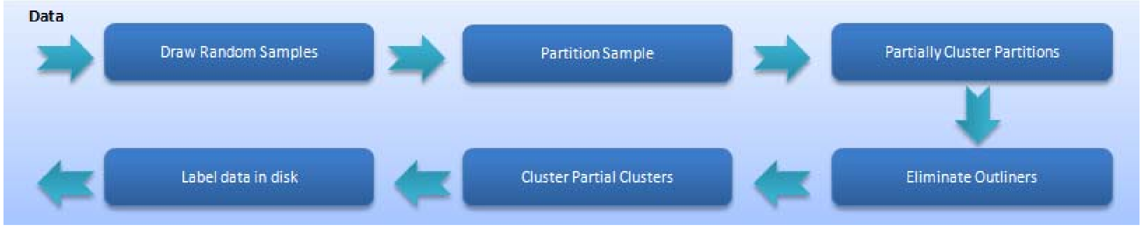


Figure 3.2: CURE Overview

3.3.3 BIRCH

BIRCH (Balance and Iterative Reducing and Clustering using Hierarchies) is useful algorithm for data in vector space model. It can also work well with outliers as that of CURE [11, 12]. It works with CF (clustering feature) and CF tree, a height balanced tree. Other notions used are number of points in cluster ‘N’, linear sum of the points ‘LS’ and sum of square of points ‘SS’.

The main innovation of Birch algorithm is to build a clustering feature tree. Clustering feature tree (CF tree) is used to scan the whole data set. CF tree summaries and represent clusters which improves the speed of cluster generation. Each entry in the CF tree represents a cluster of objects and is characterized by a triple: (N, LS, SS). N, LS and SS are as follows:

1. ‘N’ is the number of items in the sub-cluster.
2. ‘LS’ is linear sum of the points.
3. ‘SS’ is Sum of squared points.

Clustering feature tree is a height balance tree. CF tree can be built on the bases of branching factor (B), non leaf node (L) and threshold (T). So CF tree is used to summarize the information about the cluster. The tree size depends on the threshold T. Larger the values of threshold smaller will be the tree. The concepts related to CF tree are as follow:

1. Each non-leaf node has at most C entries of clusters.
2. Each leaf node has at most L CF entries.
3. The diameter of leaf node entries has to be less than the threshold value T.
4. Node size is determined by dimensionality of data space and input parameter.
5. New objects are inserted dynamically in CF tree.

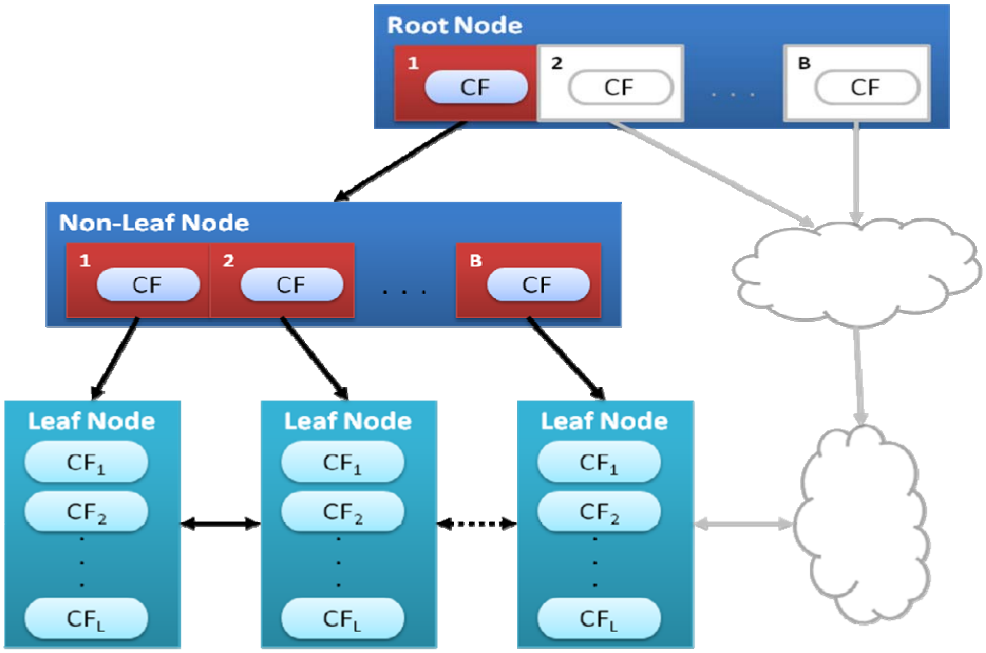


Figure 3.3: CF Tree

Figure: 3.3 shows the basic structure of clustering feature tree. There is different terminology related to CF tree. The tree shows:-

1. Root Node at the top
2. Non-Leaf Nodes
3. Leaf Nodes

For insertion into CF tree appropriate leaf node is selected. Closest child now is selected for the position of new leaf node. White boxes and clouds represent the insertion of new entries in CF tree.

Figure 3.4 depicts the main flow chart of BIRCH approach. It shows that there are four (4) phase of BIRCH approach for clustering. Phase two (2) and four (4) are optional but their use is good for better cluster generation. Phase-1 scans the data set and build CF tree in memory. Insertion in CF tree will be done at leaf. Leaves are modified to absorb the new data point by finding the closest leaf node. If leaf node is full it will be split into two leaf node and new entry will be added in parent node as well. Phase-2 is optional whose main purpose is to shrink the CF tree. Phase-2 mostly depends on the threshold value. Larger threshold value will be considered and CF entries are reinserted into new tree. Reinsertion takes place by modifying the path or without any modification depending on the situation. Phase-3 is responsible for the generation of cluster or global clustering. For Phase-3 we only consider the leaf nodes and centroid is considered as the representative of the cluster. So CF will be clustered instead of data points. Phase-4 is optional and used for cluster refining which is done by scanning the dataset again. Clusters found in Phase-3 will be considered as the seed for this activity. Data points are also redistributed and new clusters will be formed. Outliners will be removed as well.

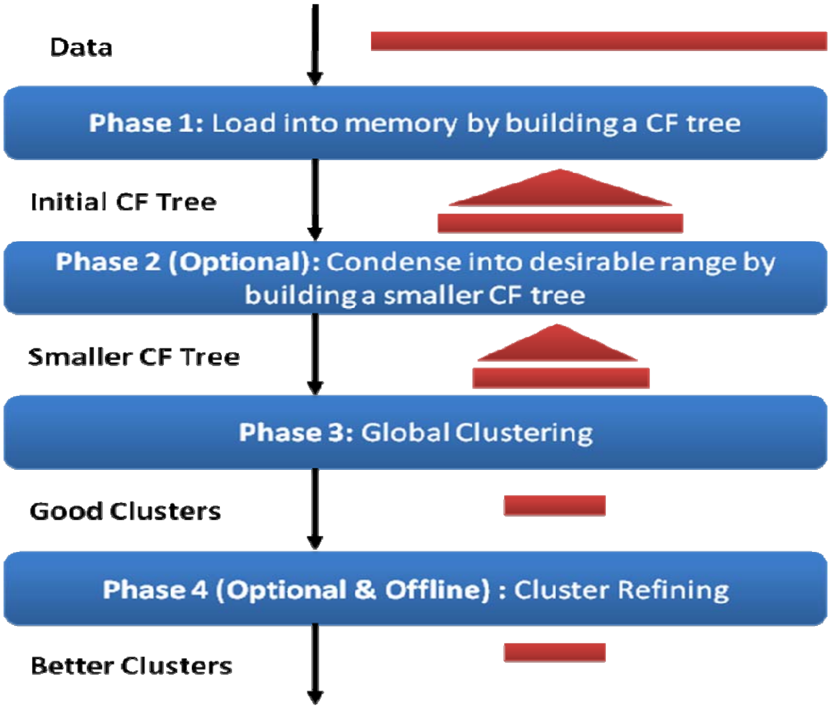

Data
Phase 1: Load into memory by building a CF tree
Initial CF Tree
Phase 2 (Optional): Condense into desirable range by building a smaller CF tree
Smaller CF Tree
Phase 3: Global Clustering
Good Clusters
Phase 4 (Optional & Offline) : Cluster Refining
Better Clusters

Figure 3.4: BIRCH Flow Chart

As it is obvious from the above discussion, each node in a CF tree can hold only a limited number of entries due to the size depending on threshold values. So a CF tree node doesn't always correspond to actual cluster. So it summarises the natural cluster to be represented in CF tree model. BIRCH can only give better result if clusters are of spherical shape. As BIRCH uses the concept of radius or diameter so it does not perform well for other shapes instead of spherical [10].

3.4 Distance Measure

Distance measure is an important component of clustering algorithms. It is useful to measure the closeness of objects. Distance measure is mainly used to measure distance between data points. Objects with smallest distance will be considered as the best candidate for merging with each other. To measure the distance between objects Euclidean formulae is used. Euclidean distance will be sufficient to group similar data instances if the physical units are the same. However Euclidean distance can sometimes be misleading. So domain knowledge must be used to guide the formulation of a suitable distance measure for each particular application.

3.5 Problem with Distance Measure

Distance measure is good for checking the differences and similarities between objects. Objects with smaller differences and with similarities will be merged for further processing. But sometimes it leads to error and merge the objects or clusters with smaller distance without having any similarity between them.

For examining the drawback of distance measure we consider the following four transactions.

- **T1= {1, 2, 3, 4}**
- **T2= {1, 2, 4}**
- **T3= {3}**
- **T4= {4}**

First we convert these transactions to Boolean points as it is necessary to convert the transactions in Boolean points to apply the Euclidean formula on them.

- **P1= (1, 1, 1, 1)**
- **P2= (1, 1, 0, 1)**
- **P3= (0, 0, 1, 0)**
- **P4= (0, 0, 0, 1)**

We use Euclidean distance to measure the difference between all pairs of points. After applying the formula we find that Dist(P1,P2) is the smallest one. Points P1 and P2 will be merged by applying centroid-based hierarchical algorithm. We get a new cluster (P12) with (1, 1, 0.5, 1) as a centroid. Then, using Euclidean distance again, we find:

- **Dist(P12,P3)= $\sqrt{3.25}$**
- **Dist(P12,P4)= $\sqrt{2.25}$**
- **Dist(P3,P4)= $\sqrt{2}$**

From above values it is clear that point P3 and point P4 has the smallest values. So, P3 & P4 has the shortest distance and should be merged. However, there is no common item or similarity between transactions T3 and T4.

The final result from above discussion is that the use of distance measure as similarity measure is not appropriate for categorical data. So, there must be some other metric for this purpose which should be used along-with distance measure to get better results.

3.6 Summary

This chapter gives an overview of what clustering algorithms are and how they operate on dataset. Here we discussed three major algorithms of text clustering including KMEAN [3, 8], BIRCH [10] and CURE [11,12] algorithms. Some salient advantages of clustering algorithms are outlined and also mentioned. Each method has its own effects and influences on other approach. We also discussed about the similarity and distance measure. By an example we proved that distance measure is not appropriate in all scenarios. It may lead to wrong results when applied to data points, so there must be some other metrics to get better results. However, the chapter provides ample knowledge and guidance about clustering algorithms and their practical operation on dataset.

Chapter 4

ROCK ALGORITHM

To overcome the problems in traditional clustering algorithm ROCK [1, 2] is a good method to generate better results by using different metrics for distance measure and similarity measure. In this chapter we will discuss different terminologies related to ROCK [1, 2] algorithm. ROCK [1, 2] is suitable for datasets containing categorical attributes. ROCK [1, 2] produces good results for agglomerative hierarchical clustering.

4.1 Overview

There are many algorithms available for clustering data like; ROCK [1, 2], KMEAN [3, 8], BIRCH [10], CURE [11, 12] etc, but here, ROCK [1, 2] (A Robust Clustering Algorithm for Categorical Attributes) has been chosen with some modifications. It belongs to the agglomerative hierarchical clustering algorithms [3]. Concept of links has been introduced in this algorithm to measure the similarities between the data points. Links play the vital and an important role in ROCK [1, 2] for the generation of clusters. It uses links instead of distance measure to merge the clusters. ROCK [1, 2] belongs to the category of agglomerative clustering algorithm.

4.2 Agglomerative Clustering Algorithm

The agglomerative clustering algorithms are built in a bottom-up manner. A top down clustering algorithms are not commonly used. The basic theme of this technique is that it starts with a single cluster which contains all objects. Then it splits the cluster until only individual objects remain. Steps of building such algorithms are as follows:-

1. Put each data point in its own cluster.
2. Select the clusters with the highest similarity.
3. Replace the selected clusters with new cluster, produced by merging the two clusters selected in step 2.
4. Repeat the above two steps until there remains only one cluster.

After performing all these steps, we will get binary cluster tree. The generated tree will contain single data point as its leaf nodes. The root node will contain all the data points. Agglomerative clustering is useful for data display as it can produce ordering for objects. It also generates smaller clusters, which are helpful in cluster discovery.

4.3 ROCK Terminology

First of all we discuss some of the terminologies related to ROCK [1, 2] algorithm. These will be helpful in understanding the theme of algorithm [1].

4.3.1Similarity function

Let us have two data points Pi and Pj. Then similarity function between two points Pi and Pj can be represented as Sim(Pi, Pj). It is used to measure the closeness between two points Pi and Pj. In ROCK [1, 2] similarity function is normalized which means that it will produce values in the range between 0 and 1. Similarity function is based on Jaccard coefficient and is defined as in Equation (1). It is calculated with two main characteristic that is, either it is present or absent.

$$Sim(P_i,P_j) = \frac{|P_i \cap P_j|}{|P_i \cup P_j|} \tag{1}$$

Where $\cap$ means the intersection between Pi and Pj and $\cup$ means the union between Pi and Pj. Suppose two documents (P1 and P2) contain the following subjects:

P1 = {Alert, Network, Security, Cost}

P2 = {Software, Cost, Alert}

Sim (P1, P2) = | P1$\cap$ P2| / | P1$\cup$P2| = 2 / 5 = 0.40

4.3.2 Neighbors and Links

There is no concept of neighbors and links in traditional algorithms. They only deal with two points. And that is why focused on just distance measure, which fails in certain cases as discussed in chapter 3.

To overcome this problem ROCK [1, 2] introduced the concept of neighbors and links. Similarity measure and closeness between two data points can be measured on the basis of links. A cluster merging is also done on the basis of links. Similarity between two points depends on the threshold value (θ). If θ exceeds some specified value then the points are considered as neighbors. The link between two points depends on the number of common neighbors between them. Link describes the global information about other points when they are in the neighborhood of those points. It is necessary for the pair of points to be in the same cluster so that link must have larger value.

We elaborate the concept of neighbors and links by the following example. Here we suppose that we have three points P1, P2 and P3 as shown in Figure 4.1 which depicts the neighbor graph with links. All the three points are distinct. The neighbor of all the tree points are described as follow:-

- **Neighbor(P1)={P1,P2}**
- **Neighbor(P2)={P1,P2,P3}**
- **Neighbor(P3)={P3,P2}**

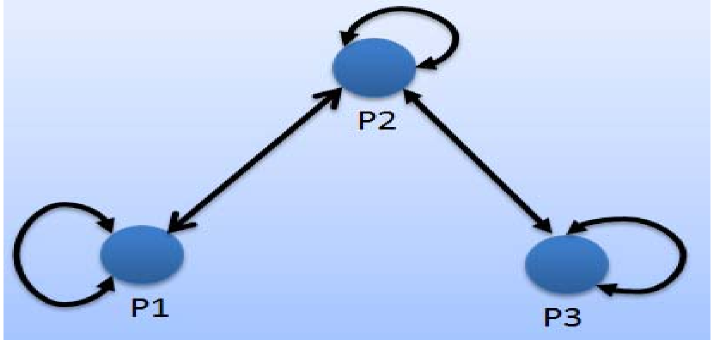


Figure 4.1: Neighboring Graph

We use neighbor graph to display the concept of link, neighbor and the relationship between the two. To calculate the number of links between any two points, it is necessary to find out first, the common neighbors between them. So link between two points can be calculated by linkage function. Let us suppose that two points P1 and P3, the linkage function will be as follow where LF is linkage function:

LF (P1, P3) = Neighbor (P1) ∩ Neighbor (P3) = {P2}

Or

LF (P1, P3) = 1

Let us say we have four data points, which are P1, P2, P3, P4 and similarity threshold value is equal to 1. We can define similarity threshold so that we can find the neighbors of any point. So two points are considered to be neighbor if and only if Sim(Pi,Pj) >= 1. It means points are neighbors only if they are identical. So they are neighbors to themselves. To find the link between two points we will use Linkage Function. Let us have two points P1 and P2. To calculate the link between P1 and P2, first we have to calculate the neighbor for both the points. The neighbors will be calculated as follow:

Neighbor (P1) = {P1}

Neighbor (P2) = {P2}

Now, linkage function between P1 and P2 will be calculated as follow. Linkage function shows the number of links between pair of points. For linkage function it is necessary to find out the neighbor for each point like above (here we calculated the neighbors for points P1 and P2). Linkage function will be the intersection of neighbors for the pair of points.

LF (P1,P2) = Neighbor (P1) ∩ Neighbor (P2) = 0

We can build a neighboring table for each point as shown in Table 4.1. Rows and Columns are the points. Here we have four points P1, P2, P3 and P4. After neighbor calculation we will calculate the linkage function between points. The table 4.1 shows the number of links (common neighbors) between the four points:

	P1	P2	P3	P4
P1	1	0	0	0
P2	0	1	0	0
P3	0	0	1	0
P4	0	0	0	1

Table 4.1: Neighboring Table with Links

Neighboring table can be transformed into neighboring graph. The above table can be transformed as neighboring graph Figure 4.2. The graph shows that every point is the neighbor of itself.

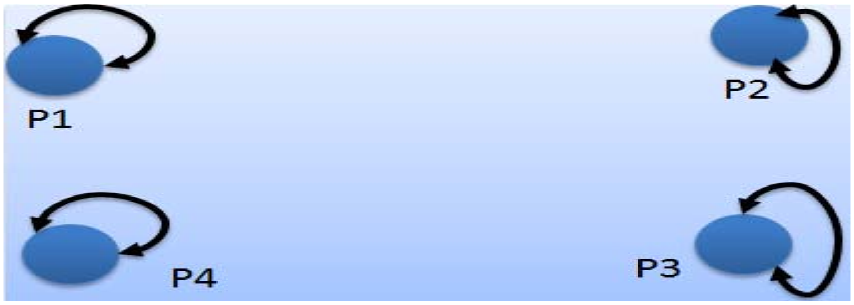


Figure 4.2: Neighboring Graph based on Links

If we want four (4) links between pair of points P1 and P3 then from the previous example, we have:

Neighbor (P1) = {P1, P2, P3, P4}

Neighbor (P3) = {P1, P2, P3, P4}

LF (P1, P3) = Neighbor (P1) ∩ Neighbor (P3) =4

We can depict these four different links in a neighboring graph. It is clear that P1 and P3 will be linked with all four points. Figure 4.3 depicts this scenario as a neighbor graph:

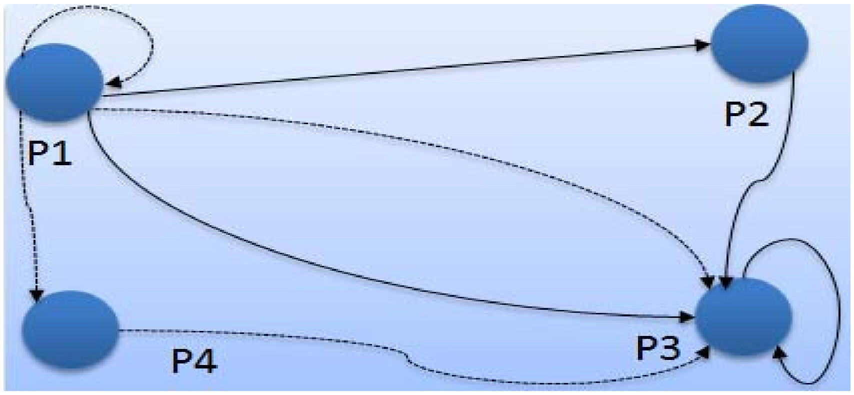


Figure 4.3: Neighbor Graph with θ=0

We can select four points P1, P2, P3, P4 from data set and similarity threshold value is equal to 0. In this case two points are neighbors if similarity value between them is greater than or equal to zero so Sim(Pi,Pj) >= 0. It means any pair of points are neighbors to any other points.

To find the linkage function, first we calculate the neighboring factor for each point. Let suppose we select two points P1 and P2, then Neighbor (P1) and Neighbor (P2) and LF (P1, P2) will be calculated as under:

Neighbor (P1) = {P1, P2, P3, P4}

Neighbor (P2) = {P1, P2, P3, P4}

LF (P1, P2) = Neighbor (P1) ∩ Neighbor (P2) = 4

After the calculation of neighbor for each point and linkage function between any pair of point, we will build a neighbor table as shown in Table 4.2. Both rows and columns of the table represent the points selected from data set. Values shown against each cell means that pair of points has that number of link between them. Here in this case the values are four for all the points, which means each point has four neighbors. Table 4.2 shows the number of links which are common neighbors, between the four points:

	P1	P2	P3	P4
P1	4	4	4	4
P2	4	4	4	4
P3	4	4	4	4
P4	4	4	4	4

Table 4.2: Common Neighbors

The above common neighbor Table 4.2 can further be explained as the neighboring graph shown in Figure 4.4. Figure 4.4 shows that each point has four links, three with other three points and one with itself. For example P1 has four links, three with P2, P3 and P4 and one link with itself.

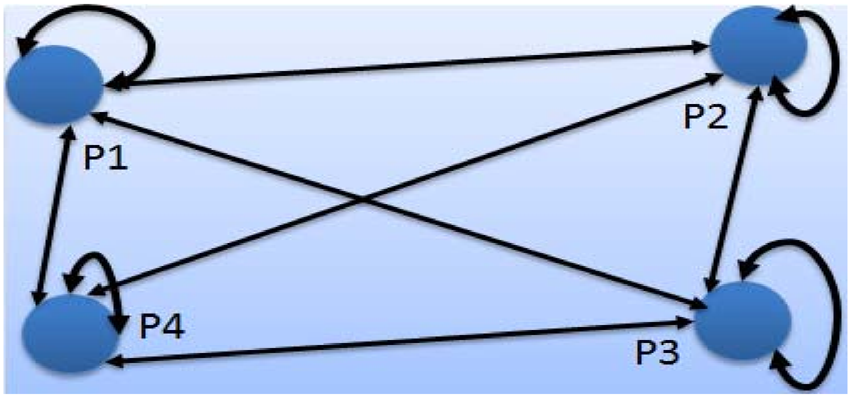


Figure 4.4: Neighboring Graph

4.3.3 Criterion function

In traditional algorithms, criterion function was defined as the distance from the mean that is center point. Lager the distance values smaller will be the value for criterion function. Criterion function is used to get the best cluster from the data set. The best cluster will be those which maximize the criterion function value. The criterion function is shown in Equation (2).

$$E1 = \sum_{i=1}^{k} ni \ X \sum_{Pq,\Pr \in Ci} \frac{link(P_q, \mathrm{P_r})}{n_i^{1+2f(\theta)}} \qquad (2)$$

- **C$_i$** denotes cluster **i**.
- **n$_i$** is the number of points in **C$_i$**.

- **k** is the number of clusters.
- θ is the similarity threshold.
- ∑ shows the summation e.g. sum of all the points.

For high degree of connectivity between points, sum of the linkage function for both points must be maximized and will be minimized for outer point, not belonging to that pair. So maximizing the criterion function value means that the sum of links of intra cluster pairs of point are maximized and it will minimize the sum of links among pairs of points belonging to different clusters we can mark them as inter cluster points, which are not belonging to that cluster.

4.3.4 Goodness measure

Goodness measure function is used to select the best pair of cluster. The resulted clusters with highest goodness measure will be merged. First link between pair of cluster will be calculated. This link is considered as the number of cross links between pair of clusters. Equation (3) shows the goodness measure function for Cluster Ci and Cj

$$g(C_i,C_j) = \frac{link[C_i,C_j]}{(n_i+n_j)^{1+2f(\theta)} - n_i^{1+2f(\theta)} - n_j^{1+2f(\theta)}} \qquad (3)$$

1. $link[C_i,C_j]$ is the linkage function.
2. n_i and n_j represents number of points in cluster C_i and C_j respectively.
3. θ is similarity threshold.

This function is the best way to maximize the criterion function. This is helpful in the identification of best pair of clusters which will be merged at each step. So the pair of clusters with identical characteristics will be merged. It means goodness measure is used to measure the quality of clusters.

4.4 ROCK Methodology

ROCK [1, 2] is a clustering algorithm. For similarity measure initially it uses Jaccard coefficient. Later on a new technique was introduced according to which two points will be similar if they share large number of neighbors. It uses the concept of neighbor, links, similarity measure and sparsification. Clusters will be generated on the basis of similarity function. We can merge the clusters if similarity measure is maximum for the pair of points. Instead of using distance measure, concept of links, neighbors, similarity measure and goodness measure is used. Following are the main steps in building ROCK [1, 2] algorithm.

1. Obtain a sample of points from the data set ‘S’.
2. Compute the link value for each set of points (Pairs).
3. Maintain a heap for each cluster ‘*i*’.
4. Perform an agglomerative hierarchical clustering on data using number of shared objects.
5. Using bottom up technique split the cluster (obtained in point 4) further until we get desired results.
6. Assign the remaining points to the found cluster.

4.5 Advantages of ROCK Algorithm

There are many clustering algorithm techniques available. Some have advantages over the other and perform well in some situations but fail in others. ROCK [1, 2] algorithm is preferable in many situations especially for categorical data. It has the following advantages over other algorithms [1]: -

1. It works well for the categorical data.
2. Once a document is added in a cluster, it will not go into another at the same level.
3. Document switching across the clusters is avoided using ROCK [1, 2] because it is an agglomerative hierarchical algorithm.
4. It uses the concept of links instead of using distance formula for measuring similarity.
5. Links are good for measuring goodness factor while distance measure fails in some situation (as discussed in chapter 3).

6. It generates better quality clusters than other algorithms.
7. Can generate better cluster labeling.
8. Provides better results than other algorithms because of goodness measure.

4.6 Summary

This chapter deals with an agglomerative clustering algorithm named as ROCK. In this method clusters are formed on the basis of criterion function, similarity function, goodness measure etc. This chapter also describes the methodology of ROCK and its advantages over other methods. By this chapter it is obvious that traditional algorithms fail in some situation for example, for categorical data, so in such situation ROCK is the best choice for applying clustering on categorical data. The chapter ends by giving advantages of ROCK algorithm.

Chapter 5

Proposed Approach (EROCK) and its Implementation

This chapter deals with the design and implementation of ROCK algorithm with modification. The modified algorithm is named as Enhance ROCK or EROCK. JAVA® with development kit JDK1.6.5 has been chosen as a development tool. NetBeans 6.5.1 has been used as an IDE for the development of this algorithm. The algorithm is developed, compiled and tested in NetBeans IDE 6.5.1. Text documents have been used for experimental evaluation.

5.1 PROPOSED APPROACH

5.1.1 Inputs

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

1. A directory containing text documents known Corpus.
2. Threshold for number of clusters to be formed.
3. Threshold value for measuring similarity of documents.
4. Threshold value for taking top most frequent words for labeling folders.

5.1.2 Basic Steps of EROCK Algorithm

1. Obtain a sample of points from the data set ‘S’.
2. Compute the link value for each set of points (Pairs).
3. Maintain a heap for each cluster ‘i’.
4. Perform an agglomerative hierarchical clustering on data using number of shared objects.
5. Assign the remaining points to the found cluster.

5.1.3 Algorithm for Clustering

Clusters play an important role in EROCK algorithm. Corpus contains the text documents. Preprocessing activity will be done on that document and intermediate form will be generated. Clustering algorithm will be applied on the intermediate form for further processing. Following are the steps involved in making the clusters, using EROCK algorithm: -

1. Build intermediate text documents (clusters) from the text file(s) present in the corpus.
2. Compute links of every document with every other document. An adjacency list of neighbors for each document has been maintained to achieve links for that document. To calculate links of a document neighboring documents are required. This process needs similarity measures to be used to calculate similarity of two documents. For similarity measure, cosine measure is used to obtain similarity between two documents.
3. After link computation, each document is now represented as a cluster. Each cluster has associated cluster link with it, which contains the information about its neighboring clusters.
4. Extract the closest clusters, to be merged for forming one cluster. This decision is made on the bases of goodness measures. Goodness measure defined as the two clusters which have maximum number of links between them. Let these two clusters are u and v.
5. Now merge the two clusters u and v. Merging of two clusters involves merging the name of two clusters, documents of two clusters and links of two clusters. This will result in a merged cluster called w.
6. For each new 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. And use them for naming the clusters.

5.1.4 Output

1. A list of clusters labeled properly.
2. Similar documents under common label.

Each cluster gets converted into a physical directory on the disk and each folder contains the documents of the respective cluster.

5.2 Design of the System

Figure 5.1 shows the overall of structure of the approach. The basic theme of the approach is to generate clusters and merge them to get best result. The main input is the Corpus. Corpus is the collection of text documents. Corpus can also be known as text corpus. It is basically used for statistical and other type of analysis. There are some basic steps which are helpful in the generation of final result.

Steps involved are getting data (documents) from the Corpus ‘D’, where ‘D’ represents the documents. Next step is the removal of dummy characters and words from the text documents. This process is known as stop word removal. These unnecessary words or characters include those parts of the document that have no impact on the structure of the data or cluster like alpha-numeric words, punctuations etc. An intermediate form will be used for further processing. This intermediate form will be used for next step.

Clusters will be generated from intermediate form of the text produced after the removal of stop words. Cluster generation includes the similarity measure, goodness measure and other parameters to choose best clusters. After getting the best clusters, they will be merged to get the resultant cluster. After this, clusters will be labeled for identification purpose. This labeling is done on the bases of similar characteristics of the cluster or term-based frequencies. Most frequent words or terms will be considered for cluster or document labeling.

This process will continue until desired numbers of clusters are generated. Cluster generation depends on similarity threshold value, number of clusters to be obtained and goodness measure. Linkage measure plays an important role for the calculation of similarity measure and cluster merging.

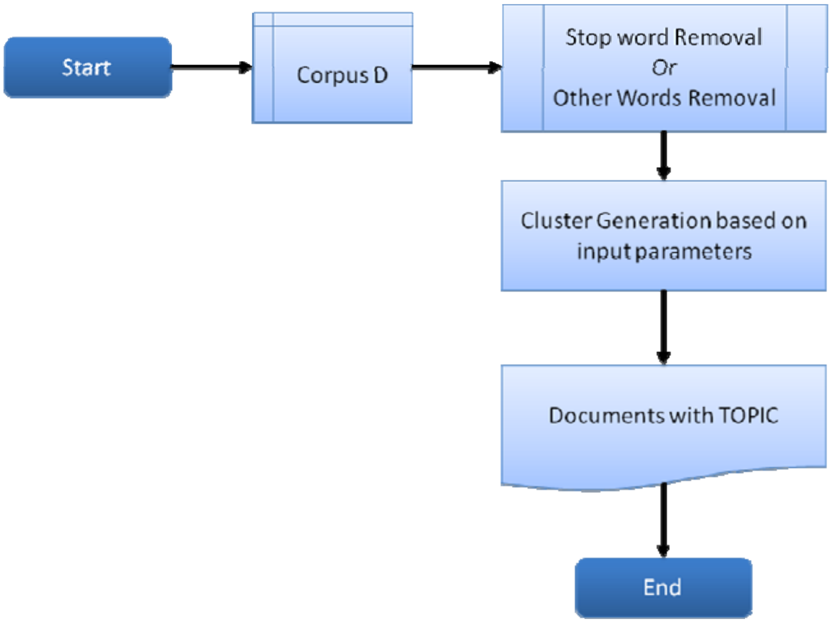


Figure 5.1: Overall Application Structure

The whole application flow of EROCK is shown in Figure 5.2. Flat text file(s) has been converted into documents. Flat text file is a plain text or mixed text. It usually contains one record per line. It means records or text has some delimiter for separation. A line can be ended by some line breaker e.g. dot (.) sign. These text files will be used for algorithm evaluation. These are the main input for EROCK. These file can also be termed as documents. Dummy character or stop words will be removed from these documents for further processing.

EROCK algorithm will be applied on these generated documents. Clusters will be created from these documents. Cluster generation activity can be done by applying some primary parameters, which are necessary for generating best results. Each document will be placed in respective clusters. Merging of clusters will be done on all those clusters having high closeness. Closeness is calculated by linkage function. Linkage function describes the link between pair of documents. Best clusters are chosen for merging. Best pair of clusters can be selected by applying goodness measure.

Documents within one cluster have high similarity as compared to documents in other clusters. Similarity between these clusters has been calculated by cosine measure. Adjacency list has been used to store all these clusters in memory. Adjacency list is the most appropriate way to store the cluster. This technique is efficient as compared to sparse matrix, used in traditional ROCK algorithm.

After this process, label generation process will be invoked to generate labels (topics) for merged clusters. Topics will be picked by most frequent word occurrences in a document. 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. Relevancy is calculated by linkage function and criterion measure. The entire relevant document with respect to word frequency and neighborhood will be the best candidate for merging process. Physically these documents will be put in folders with topics or label as folder name. Each folder will contain the document with high relevancy. The accuracy depends on the threshold values.

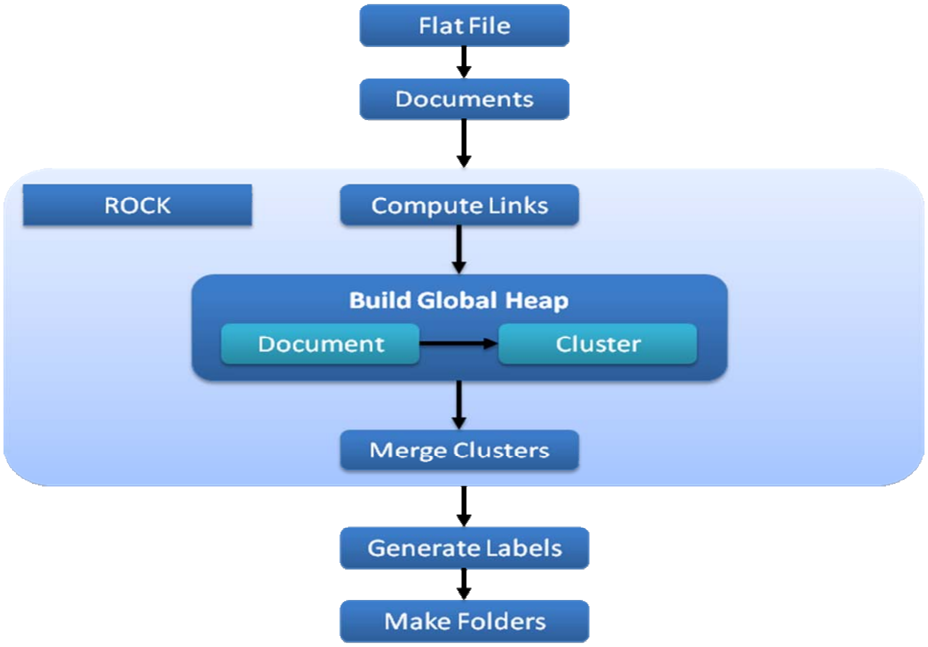


Figure 5.2: Application Flow

5.3 Contributions & Modifications

We have done some modification in existing ROCK algorithm. Contribution and modifications done in ROCK algorithms are as follow.

5.3.1 Similarity Function

ROCK algorithm uses similarity measure based on Jaccard coefficient for transactions T1 and T2 [1], as shown in Equation (4).

$$Sim(T_1, T_2) = \frac{|T_1 \cap T_2|}{|T_1 \cup T_2|} \tag{4}$$

To measure similarity, instead of using equation in Figure4.3, cosine measure [2] has been applied in this project. It is used to measure the similarity between two vectors by finding cosine angle between them. Let suppose v1 and v2 are two vectors, cosine similarity between v1 and v2 will be measured as in Equation (5). Vectors v1 and v2 are also known as term frequency vectors. Similarity measure in Equation (4) is dependent on document length. So it will change with respect to the document length. Cosine similarity is independent of the document length. The reason is that it uses the concept of vectors and cosine as the angle between pair of vector. If COSINE angle is one (1) then two vectors are getting closer and similarity increases. Due to this property processing with respect to time becomes efficient.

$$CosSim(v_1,v_2) = \frac{|v_1 . v_2|}{|v_1|\,|v_2|} \tag{5}$$

Equation (5) shows the similarity measure based on cosine measure. Where CosSim (v1,v2). means Cosine Similarity between vector v1 and v2. |v1. v2| is the vector dot product and it is defined as in Equation (6):

$$\sum_{i\ =\ 1}^{k} i = V_1 i\ V_2 i \tag{6}$$

In Equation (6) we will consider v1 as an absolute value and |v1| is defined as shown in Equation (7):

$$|V_1| = \sqrt{V_1 . V_2} \tag{7}$$

5.3.2 Graph Representation

Adjacency list, instead of sparse matrix, has been used. It represents edges in a graph as a list. It is because the sparse matrix requires more space and long list of references. Thus efficiency suffers adversely. Whereas adjacency list keeps track of only neighboring documents [6] and utilizes lesser space. It enhances efficiency of the algorithm as well. So time-space complexity will be affected by doing these changes.

Besides space trade-off, it is easy to find all vertices adjacent to a given vertex in a list. Scanning an entire row in a list requires O(n) time. Adjacency list is preferred when data are large and sparse.

ROCK algorithm draws random sample from the database. It then calculates links between the points in the sample. Instead of that the purposed approach EROCK in this thesis makes use of entire corpus for clustering. Every point in the corpus is treated as a separate cluster which means 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. Best cluster from the pair of clusters are chosen on the bases of goodness measure. This process goes on until the specified numbers of clusters are formed. So instead of sampling, EROCK considers whole corpus and every point is considered as a cluster.

5.3.3 Goodness Measure

Goodness measure has been used for merging of clusters [5]. It is defined as the maximum number of cross links between cluster points Pi, Pj. It can be represented as g(Pi,Pj). Goodness measure will be calculated on the basis of: Size of Points Pi, Pj, Input Parameters, Cross Link Count between clusters points Pi, Pj. Clusters with high cross link count are the best candidate for merging. If cross link is high, it means that this cluster is linked with others and is appropriate to be chosen as the best cluster for merging process. More cross links also show that this cluster is the neighbor for other and every other cluster is directory linked with it. So goodness measure is also good for linkage function which tells about the attachment of the cluster with other cluster in the same document. It means cross links have importance and play vital role for cluster merging activity. Goodness measure is shown in Equation (8) where Link [Pi, Pj] shows the link calculation between pair of points. Theta (θ) is the similarity threshold value.

$$g(P_i, P_j) = \frac{link\ [P_i, P_j]}{(n+m)^{1+2f(\theta)} - n^{1+2f(\theta)} - m^{1+2f(\theta)}} \tag{8}$$

5.4 EROCK with Example

In this section we will try to elaborate the algorithm with an example. For this we select some frequent terms from the corpus. Terms are selected on the bases of their occurrence in the document.

Suppose we have four terms or words that contain some subjects. These four terms or data points chosen for this purpose are P1, P2, P3 and P4 as follows:

P1 = {Virus, Network, Worms, Alert}

P2 = {Cost, Network, Worms}

P3 = {Alert, Cost, Network}

P4 = {Cost, Worms, Arora}

For this example, the similarity threshold value used will be 0.3, and number of required cluster is equal to two (2). Using Jaccard coefficient with cosine measure as a similarity measure, we can obtain the similarity table as shown in Table 5.1. This table shows the closeness of the pair of points after applying similarity measure formula. Table 5.1 shows the data points in rows and columns with similarity values in corresponding cell.

	P1	**P2**	**P3**	**P4**
P1	1	0.4	0.4	0.17
P2		1	0.5	0.5
P3			1	0.2
P4				1

Table 5.1: Similarity Table

Since we have a similarity threshold equal to 0.3, we derive the adjacency table. Adjacency table has been shown in Table 5.2 which is being built from similarity table as shown in Table 5.1. All the pair of points having value greater than or equal to 0.3 will be considered as 1 and other values less than 0.3 will be considered as 0.

	P1	**P2**	**P3**	**P4**
P1	1	1	1	0
P2		1	1	1
P3			1	0
P4				1

Table 5.2: Adjacency Table

By multiplying the adjacency table with itself, we derive the following table which shows the number of links (or common neighbors). Common neighbors can also be calculated by checking the similar terms in each pair of point. Neighboring table has been shown in Table 5.3.

	P1	P2	P3	P4
P1	-	2	2	1
P2		-	2	2
P3			-	1
P4				-

Table 5.3: Common Neighbor with Links

We compute the goodness measure for all adjacent points. We assume that **f(θ) =1-θ / 1+θ**. Where θ is the similarity threshold value and it will be used for the calculation of goodness measure. Equation (8) shows the goodness measure equation.

$$g(P_i,P_j) = \frac{link[P_i,P_j]}{(n+m)^{1+2f(\theta)} - n^{1+2f(\theta)} - m^{1+2f(\theta)}} \qquad (8)$$

We can obtain the pair wise goodness measure table by applying the equation shown in Figure 5.7. Pair wise goodness measure table is shown in Table 5.4.

Pair	Goodness Measure
P1,P2	1.35
P1,P3	1.35
P1,P4	0.45
P2,P3	1.35
P2,P4	0.90
P3,P4	0.45

Table 5.4: Pair wise Goodness Measure

We have an equal goodness measure for merging (P1, P2), (P2, P1), (P3, P1). Now, we start the hierarchical algorithm by merging, say P1 and P2. A new cluster (let's call it C(P1,P2)) is formed. It should be noted that for some other hierarchical clustering techniques, we will not start the clustering process by merging P1 and P2, since Sim(P1,P2) = 0.4,which is not the highest. But, EROCK uses the number of links as the similarity measure rather than distance.

Now, after merging P1 and P2, we have only three clusters. The following table shows the number of common neighbors for these clusters:

	C(P1,P2)	**P3**	**P4**
C(P1,P2)	**-**	**3+3**	**2+1**
P3		**-**	**1**
P4			**-**

Table 5.5: Common Neighbors after Cluster Merging

The next calculation is for goodness measure. We can obtain the following goodness measures for all adjacent clusters:

Pair	**Goodness Measure**
C(P1,P2),P3	1.31
C(P1,P3),P4	0.66
P3,P4	0.22

Table 5.6:Pair-wise Goodness Measure

Since the number of required clusters is 2, then we finish the clustering algorithm by merging C(P1,P2) and P3, obtaining a new cluster C(P1,P2,P3) which contains {P1,P2,P3} leaving P4 alone in a separate cluster.

5.5 Summary

This chapter deals with the entire design of the proposed approach along with its implementation in JAVA. There are many retrieval models/ algorithms/ systems, which one is the best? How far down the ranked list will a user need to look to find some/all relevant documents? Effectiveness is related to the relevancy of retrieved items. The main hurdle was the understanding of the ROCK algorithm with respect to its implementation design and then its modification. The major issue was the understating and use of adjacency list instead of adjacency matrix to remove sparsity. Labeling the clusters was another major issue to finalise the process of EROCK algorithm.

Besides all these problem ROCK algorithm had been modified. We changed its similarity measure function from Jaccard coefficient to cosine measure. By doing this we get desired change in processing with respect to the time it takes to complete the job. We also used adjacency list instead of sparse matrix for efficient memory utilization.

Chapter 6

Results and Discussion

Results for this study are mainly based on two types of analysis, Cluster Analysis and Label Analysis. The EROCK algorithm was applied to the text documents. Initially stop words and other useless items were removed from the document, known as pre-processing stage. After the generation of intermediate form, EROCK clustering algorithm will be applied on it. We also compared both of the algorithms to check the performance.

6.1 Cluster Analysis

There are total four hundred (400) documents in a specified directory. Number of clusters to be obtained varies from one (1) to ten (10). In the same way similarity values (threshold) can have the range from 0.1 to 1.0.

Some of the results generated from these cases are depicted here. Table 6.1 shows the final output based on similarity values and clusters to be obtained. These are some of the scenarios which have been applied on the EROCK algorithm for resulted cluster generation with labels.

Figure 6.1 shows the overall results obtained from this study. According to the figure, clusters to be obtained are dependent on similarity threshold values. The figure shows the number of clusters to be obtained along with the threshold values. The inferences gained from Figure 6.1 are given as under: -

- If the number of clusters to be obtained is equal to the number of documents then similarity factor has no affect on the clustering.
- If the number of clusters to be obtained is less than actual documents, then the number of clusters to be obtained depends upon the similarity threshold.

- Increase in the Threshold of Top Frequent Word of Cluster will increase the Size of the Cluster Label.
- For the dataset which we used for analysis, EROCK discovered almost pure clusters containing documents with respect to their topics.
- EROCK generates good quality of clusters that means the documents were organized almost perfectly with respect to their topics.

Similarity Values	Clusters Obtained
0.1	1
0.2	2
0.3	3
0.4	4
0.5	5
0.6	6
0.7	7
0.8	8
0.9	9
1	10

Table 6.1: Parameters for EROCK

Table 6.1 shows cluster labeling by similarity threshold values and number of clusters to be obtained. Documents with high similarity will be placed under one label. 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 this label. If similarity value is 0.2 and numbers of clusters to be obtained are 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 clusters will be labeled separately. It means that labeling or document topics are mainly dependent on both similarity threshold values and number of clusters to be obtained.

Labeling analysis involves the analysis of label generated for clusters based on similarity threshold values. This analysis is helpful to check whether process is accurate or not. Label generation varies as per similarity values as shown in Table 6.1.

5.2 Clustering Analysis

To conduct any type of analysis, some of the parameters are mandatory to check it. In our case the following parameters are required: -

- Number of Documents: 10
- Number of Cluster to be obtained: 3
- Similarity Threshold: 0.3

There are total four hundred (400) documents in a specified directory. The number of clusters to be obtained varies from one (1) to ten (see Table 6.1). In the same way similarity values (threshold) can have the range from 0.1 to 1.0 (see Table-6.1).

Some of the results generated are shown in this section. These results are taken by keeping some scenarios into account. These results are mainly based on similarity threshold value and number of clusters to be obtained as discussed in above section.

Similarity Values	Clusters Obtained
0.1	1
0.2	1
0.3	4
0.4	7
0.5	8
0.6	8
0.7	9
0.8	10
0.9	10
1	10

Table 6.2: Clusters with Similarity Measure

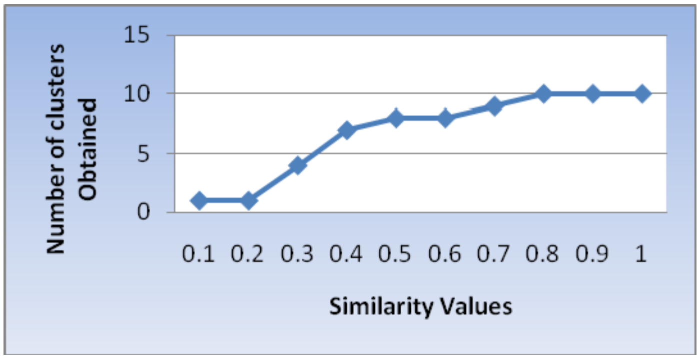


Figure 6.1: Graph (Similarity Measure vs Clusters)

Similarity Measure	Clusters Obtained
0.1	6
0.2	6
0.3	6
0.4	7
0.5	8
0.6	8
0.7	9
0.8	10
0.9	10
1	10

Table 6.3: Cluster with Similarity Measure

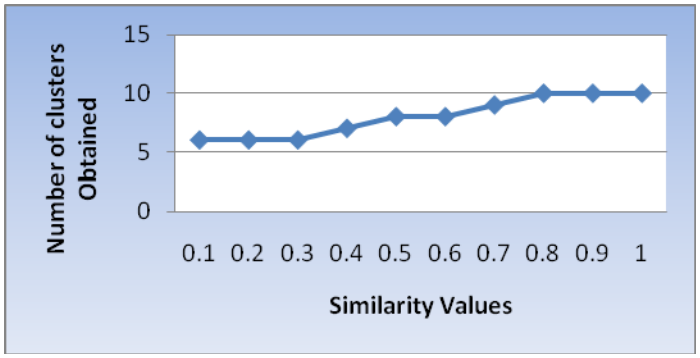


Figure 6.2: Graph (Similarity Measure vs Clusters)

Similarity Values	Clusters Obtained
0.1	10
0.2	10
0.3	10
0.4	10
0.5	10
0.6	10
0.7	10
0.8	10
0.9	10
1	10

Table 6.4: Cluster with Similarity Measure vs Clusters

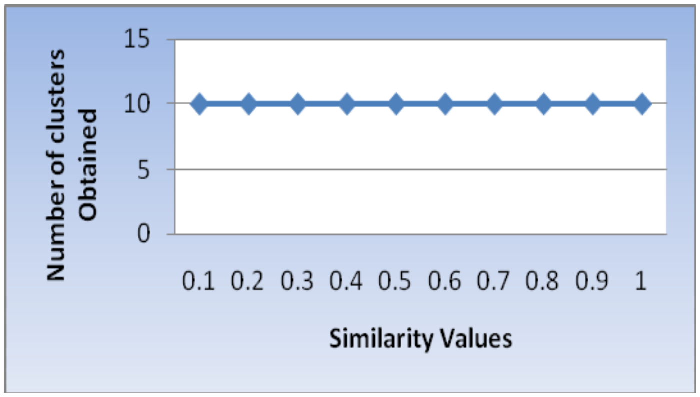


Figure 6.3: Graph (Similarity Measure vs Clusters)

The above analysis shows that grouping of the data is based on similarity value, cluster(s) to be obtained and the corpus size (number of documents) used.

6.3 Labeling Analysis:

This analysis includes the topic generation against documents. All the documents whose term frequencies, link and neighbors are more in numbers and have less cosine distance, will be merged / collected against one topic or label. This process tries to select descriptive labels for the clusters based on ROCK algorithm. Typically, the labels are obtained by examining the contents of the documents in a cluster. A good label not only summarizes the central concept of a cluster but also uniquely differentiates it from other clusters in the collection. So labeling is basically used to differentiate among clusters.

Here is the result obtained against following parameter values for labeling analysis:

- Number of Documents are 10.
- Number of Cluster to be Obtained are 3.
- Similarity Threshold is 0.3.

Top Label Cluster %age	Clusters 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 6.5: Label Analysis

6.4 How ROCK & EROCK are compared

For comparing any sort of technique some factors are necessary. To compare ROCK and EROCK algorithms we are focusing on following factors: -

- Size of dataset, number of documents used
- Similarity Threshold Value from 0.1 to 1.0
- Number of clusters to be obtained from 1 to 10

It is also noteworthy that comparison should be performed on the same machine and under the same environment. For comparisons of ROCK & EROCK we used a machine with the following specifications: -

- Windows Vista ™ Home Premium
- Hewlett & Packard
- HP Pavilion dv2700 Notebook PC
- Intel® Core ™ 2 Duo CPU T5750 @ 2.00GHz 2.00GHz Processor

- 2.00GB Memory (RAM)
- 32-bit Operating System

It is also necessary that algorithm should also be implemented in same the technology and on the same platform. For this we implemented ROCK & EROCK algorithm on the same technology and platform. Implementation Technology:

- Windows Platform
- Java with JDK 1.6
- NetBeans IDE 6.5.1
- Console Application

Performance results of both algorithms (ROCK & EROCK) are shown in following table with similarity threshold is 0.1.

# of Documents	Size of Documents	# of Words	# of Characters	Similarity Threshold	Performance	
					ROCK	EROCK
10	590 KB	55868	374031	0.1	7 seconds	5 seconds
20	852 KB	79647	536774	0.1	8 seconds	8 seconds
30	1137 KB	115997	769525	0.1	13 seconds	12 seconds
40	1403 KB	141370	943278	0.1	19 seconds	16 seconds
50	1679 KB	174287	1160453	0.1	27 seconds	21 seconds
100	3369 KB	340885	2270742	0.1	1 minute 15 seconds	55 seconds
200	6810 KB	681770	4541484	0.1	4 minutes 3 seconds	2 minutes 42 seconds
400	13619 KB	1363540	9082968	0.1	13 minutes 36 seconds	8 minutes 47 seconds

Table 6.6: ROCK vs EROCK with θ 0.1

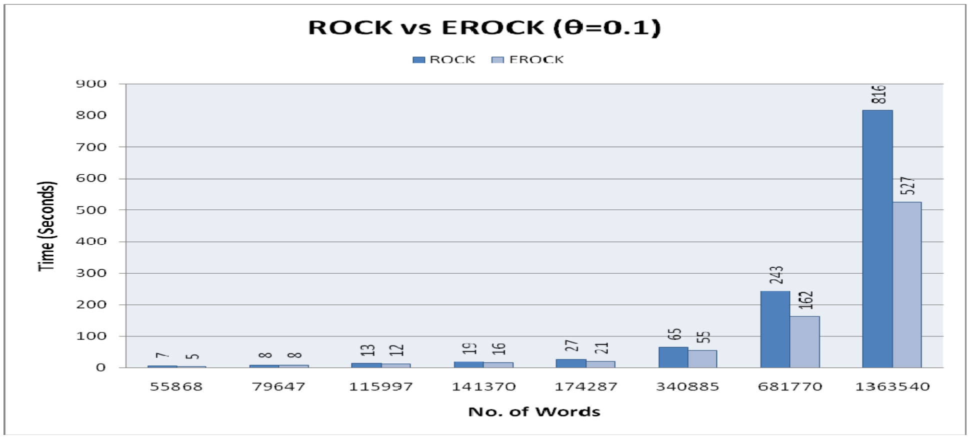


Figure 5.4: ROCK VS EROCK (θ=0.1)

Other performance results of both algorithms (ROCK & EROCK) are shown in the following table with similarity threshold is 0.5.

# of Documents	Size of Documents	# of Words	# of Characters	Similarity Threshold	Performance	
					ROCK	EROCK
10	590 KB	55868	374031	0.5	7 seconds	5 seconds
20	852 KB	79647	536774	0.5	9 seconds	7 seconds
30	1137 KB	115997	769525	0.5	14 seconds	11 seconds
40	1403 KB	141370	943278	0.5	21 seconds	16 seconds
50	1679 KB	174287	1160453	0.5	29 seconds	21 seconds
100	3369 KB	340885	2270742	0.5	1 minute 18 seconds	56 seconds
200	6810 KB	681770	4541484	0.5	4 minutes 25 seconds	2 minutes 43 seconds
400	13619 KB	1363540	9082968	0.5	14 minutes 9 seconds	8 minutes 38 seconds

Table 6.7: ROCK vs EROCK with θ 0.5

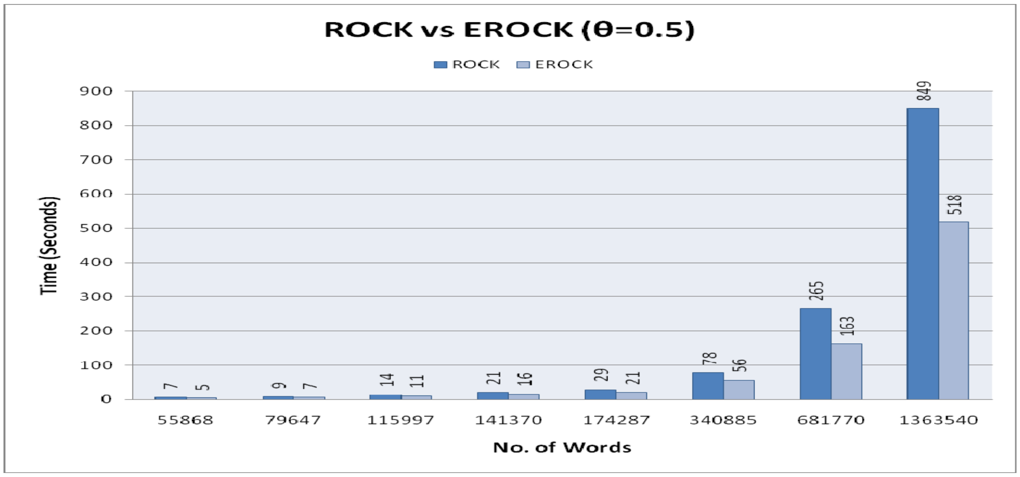


Figure 6.5: ROCK VS EROCK (θ=0.5)

Performance results of both algorithms (ROCK & EROCK) are shown in the following table when similarity threshold is 0.8.

# of Documents	Size of Documents	# of Words	# of Characters	Similarity Threshold	Performance	
					ROCK	EROCK
10	590 KB	55868	374031	0.8	6sec	4sec
20	852 KB	79647	536774	0.8	8sec	8sec
30	1137 KB	115997	769525	0.8	13sec	11sec
40	1403 KB	141370	943278	0.8	19sec	16sec
50	1679 KB	174287	1160453	0.8	29sec	23sec
100	3369 KB	340885	2270742	0.8	1 minute 19 seconds	57 seconds
200	6810 KB	681770	4541484	0.8	3 minutes 58 seconds	2 minutes 44 seconds
400	13619 KB	1363540	9082968	0.8	13 minutes 52 seconds	9 minutes 8 seconds

Table 6.8: ROCK vs EROCK with θ 0.8

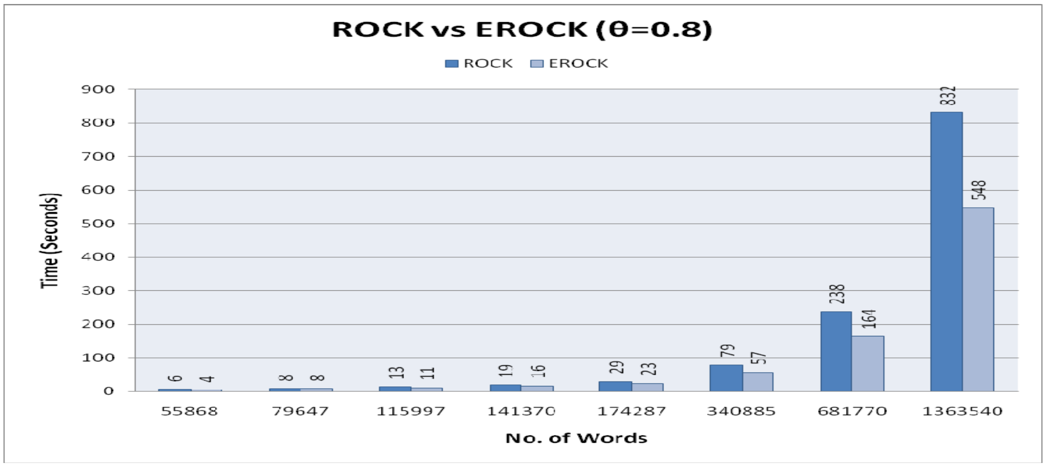


Figure 6.6: ROCK VS EROCK (θ =0.8)

From the above results, it is quite obvious that when corpus size increases ROCK takes much greater time as compared to EROCK. So we can say that performance of EROCK increases with the increase in corpus size while the performance of ROCK decreases.

6.5 Final Results (Document with Topic)

The following table describes the final result in which topics against documents have been generated based on similarity threshold. Documents with similarities were put under one topic. Last column of Table 6.9 shows the folder name, which are frequent terms taken from the documents by using EROCK algorithm.

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 6.9: Final Results

6.6 Summary

In this chapter analysis of proposed scheme has been done for standard test documents and results are shown based on EROCK techniques. There are two types of analyses discussed here. One is clustering analysis and the other is labeling analysis. The results provide arranged documents under common label or topic.

Performance of both the algorithms has also been compared. From the comparison it is clear enough that EROCK has much less time (processing time) as that of ROCK algorithm. EROCK performance is 30% to 40% better than that of ROCK algorithm. Said comparisons have been done under the same conditions and on the same machine.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

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 can only be merged if closeness and inter connectivity of items within both clusters are of high significance.

Our innovation is that we label the documents, in order to cluster them, by using the terms (sequence of words) and events (set of terms) mentioned within the documents. Additionally we have altered a clustering algorithm appropriate for categorical data. The algorithm implemented here depends mainly on three factors:

- **Similarity threshold**: Similarity threshold is used to measure the closeness of the clusters. For this purpose cosine measure is used. The values vary from 0.1 to 1.0.
- **Number of Clusters to be obtained**: Used for getting desired number of clusters. In our case the values varies from one (1) to ten (10).
- **Corpus Size**: Input directory containing text document used for the analysis.

The study described in this paper relates to the field of text mining. Much work has been done in this area. Researcher developed good algorithm for cluster analysis. ROCK [1, 2], KMEAN [3, 8], BIRCH [10] and CURE [11, 12] are some of the well known algorithm used for cluster analysis.

During this study it is observed that clustering has some limitations. First is dimensionality, the second is scalability, third is accuracy, and forth is meaningful cluster description and domain knowledge.

7.2 Future Work

There are many areas in text mining where one may carry on his/her work to enhance the scope of those areas. Out of these, the labeling of the clusters is a very daunting challenge of the modern age. No remarkable effort has so far been made in this regard to get good result. That is why automatic labeling of the clusters is not so accurate. A keen and concerted hard work has been done to remove this hurdle with the view meet the challenge of time.

Similar technique (EROCK) can be applied on the verses of the Holy Quran where goal will be to cluster the verses of The Holy Quran based on the meaning and aim of a verse. A single verse usually deals with many subjects. So verses treasure data will be arranged. Verses of the Holy Quran will be treated as records while the related subject will be treated as the attributes of the record. The attributes will be treated as Boolean T/F or 1/0.

Hence such efforts can be diving deep in the sea of the Holy Quran and elaborate the profound meaning of verses. Thus making the Holy Book easily understandable for not only the muslims but the humanity at large. In fact, it is but natural that each and every step forward in the field of Science and Technology leads the humanity towards the destination of knowing the true and natural religion i.e. Islam.

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.