

**SIMILARITY DETERMINATION AND DOCUMENT  
CLUSTERING IN VECTOR SPACE MODEL**

*Dissertation submitted to the Jawaharlal Nehru University in partial  
fulfillment of the requirements for the award of degree of*

**MASTER OF TECHNOLOGY**  
*in*  
**Computer Science and Technology**

*by*

**Rajesh Singh**



**SCHOOL OF COMPUTER AND SYSTEMS SCIENCES  
JAWAHARLAL NEHRU UNIVERSITY  
NEW DELHI – 110067**

**July, 2007**



# जवाहरलाल नॅहरू विश्वविद्यालय

SCHOOL OF COMPUTER & SYSTEMS SCIENCE

Jawaharlal Nehru University

New Delhi-110067

## DECLARATION

I hereby declare that this dissertation entitled “**Similarity Determination And Document Clustering In Vector Space Model**” submitted by me to the **School of Computer and Systems Sciences, Jawaharlal Nehru University, New Delhi** for the award of degree of **MASTER OF TECHNOLOGY in COMPUTER SCIENCE AND TECHNOLOGY** is a bonafide work carried out by me under the supervision of **Dr. Aditi Sharan.**

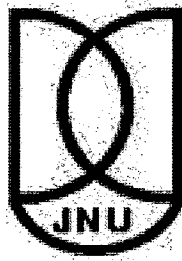
The matter embodied in the dissertation has not been submitted to any other University or Institution for the award of any other degree or diploma.

**Rajesh Singh**

School of Computer & Systems Sciences,

Jawaharlal Nehru University,

New Delhi-110067



# जवाहरलाल नॅहरू विश्वविद्यालय

SCHOOL OF COMPUTER & SYSTEMS SCIENCE

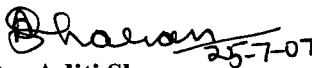
Jawaharlal Nehru University

New Delhi-110067

## CERTIFICATE

This is to certify that this dissertation entitled “**Similarity Determination And Document Clustering In Vector Space Model**” submitted by **Rajesh Singh** to the **School of Computer and Systems Sciences, Jawaharlal Nehru University, New Delhi** for the award of degree of **MASTER OF TECHNOLOGY in COMPUTER SCIENCE AND TECHNOLOGY** is a bonafide work carried out by him under my supervision.

The matter embodied in the dissertation has not been submitted to any other University or Institution for the award of any other degree or diploma.


  
25-7-07  
**Dr. Aditi Sharan**

(Supervisor)

Assistant Professor, SC & SS,

Jawaharlal Nehru University,

New Delhi-110067

  
**Prof. Parimala N:**  
Dean

School of Computer & Systems Sciences  
JAWAHARLAL NEHRU UNIVERSITY  
Dean, NEW DELHI-110067

SC & SS,

Jawaharlal Nehru University,

New Delhi-110067

*To My Parents.....*

## **Acknowledgments**

This dissertation would not have been possible without the whole-hearted support of a lot of individuals. This is an attempt to acknowledge their help, support and guidance, any omissions are involuntary.

This dissertation work has been done under the supervision of Dr. Aditi Sharan. I am immensely grateful to her for her valuable suggestions and continuous guidance throughout the course of this study.

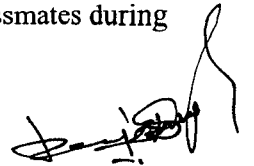
I owe my heartfelt gratitude to Prof. S. Balasundaram, Dean, School of Computer & Systems Sciences, Jawaharlal Nehru University, New Delhi, for his kind and active cooperation during the course of study.

I am also grateful to Dr. R. K. Agrawal for his ungrudging help and valuable support through out the course of the study. I wish to express my sincere thanks to Prof. R. G. Gupta, Prof. K. K. Bhardwaj, Dr. D.K. Lobiyal and Dr. T. V. Vijay Kumar too for their constant support and encouragement during the course of study.

I would also like to thank to entire faculty and staff of SC&SS for their cooperation during the course of study.

I wish to give special thanks to my parent for encouragement and moral support.

I also extend my thanks to my friends Mrs. Hazra Imran, Yamini Rathaur, Prashant Chaudhary, Girish Kumar Singh, Piyush Kr. Shrivastava and Ashish for their care and morale support. I cannot forget the moments shared with my friends and classmates during the course of study.



**Rajesh Singh**

## **ABSTRACT**

Document clustering has been investigated for use in a number of different areas of text mining and information retrieval (IR). It has gained more importance for web based data. In addition to increasing precision or recall in retrieval system, recently it has been proposed for use in browsing a collection of documents or in organizing results returned by search engine in response to user's query. This dissertation addresses the issue of document clustering.

Text documents can not be clustered as such instead they have to be represented using some model. Based on this model similarity measure and hence clustering method is selected. The purpose of this dissertation is to address the issue of document clustering using Vector space model (VSM).

In this work we first explain the representation of documents in vector space model. Then we have tried to identify, analyze and compare various similarity measures that have been used to find similarity between documents. We have made a survey of various clustering algorithm that have been used for clustering documents. Further we have performed some experiments based on our study to see empirically effect of various similarity measures on document clustering.

# Contents

---

<b>DECLARATION</b>	<b>i</b>
<b>CERTIFICATE</b>	<b>ii</b>
<b>ACKNOWLEDGEMENTS</b>	<b>iv</b>
<b>ABSTRACT</b>	<b>v</b>
<b>Chapter 1: Introduction</b>	<b>1</b>
<b>1.1 Vector Space Model</b>	<b>3</b>
1.1.1 Preprocessing	6
<b>1.2 Similarity determination</b>	<b>7</b>
<b>1.3 Document clustering</b>	<b>10</b>
<b>1.4 The Measure of Cluster Quality</b>	<b>13</b>
<b>Chapter 2: Approaches for document Clustering</b>	<b>15</b>
<b>2.1 Traditional Approaches</b>	<b>16</b>
2.1.1 Partitional	16
2.1.1.1 K-Means Clustering Algorithm	17
2.1.1.2 Bisecting K-Means Clustering Algorithm	24
2.1.2 Hierarchical	25
<b>2.2 Approaches for Clustering Large Data Sets</b>	<b>28</b>
<b>2.3 Phrase Based Clustering</b>	<b>28</b>
2.3.1 Suffix Array	29
2.3.2 STC	30
2.3.3 Advantages Phrase Based Document Clustering	33

2.4 Genetic algorithm Based Clustering Approaches	34
2.5 Neural Network	35
2.6 Cluster Parameter	36
2.7 Cluster Based Search	37
2.8 Applications of clustering Approaches	38
<b>Chapter 3: Proposed Work</b>	<b>39</b>
3.1 Motivation	39
3.2 Proposed Work	39
<b>Chapter 4: Implementation of Proposed Work</b>	<b>41</b>
4.1 First Data Set	41
4.2 Second Data Set	48
4.3 Final Results on the Basis of Four Data Sets	52
<b>Chapter 5: Conclusion</b>	<b>55</b>
<b>Reference</b>	<b>56</b>



### Introduction

---

Ever since the advent of computer systems and in particular the Internet the amount of information at our disposal has been increasing exponentially. This phenomenal growth in data is not always a blessing: the more information is available, the more difficult it becomes to find one's way to the particular piece of information of interest. As a consequence investigations into old and new techniques for dealing with the extraordinary flood of data remain topical for information science. Document Clustering is one of the important techniques in this direction, which groups the documents based on their content so as to reduce the search space as well as increase search efficiency [6, 20].

Clustering has a rich and independent history of its own. Clustering is an unsupervised method where the labels of classes are not known a priori; instead the classification depends on similarity between data sets. Relatively recently it has acquired its new application in the field of Information Retrieval [29]. Clustering is used here to divide large unstructured document corpora into groups of more or less closely related documents. The clusters can then be used as a well-arranged interface to a potentially huge and overwhelming number of documents, allowing a prospective user to home in quickly on his specific requirements [29].

Most cluster methods do not work directly with text of documents. Instead documents have to be represented into some model. On the basis of model they may use a matrix of similarity computations between all pairs of documents or compute similarities between documents as needed to build clusters. These similarity computations are one of the most computation-intensive parts of the clustering process. Even for small collection of few hundred documents, the number of similarity computations is in thousands. Matrix methods begin with a precomputed set of all pairs of similarities [6]. Alternate cluster methods attempt to minimize number of computations needed by computing similarities as needed. Therefore document clustering consists of two fundamental stages: the

transformation of documents as linear strings of words into suitable data structures/models and the algorithmic grouping of these representations based on some similarity measure.

Various models have been proposed for retrieving textual information. Three classical models being: Boolean, Vector, Probabilistic models [29]. In the Boolean model, documents and queries are represented as set of index terms. Thus, this model is *set theoretic*. In the vector model, documents and queries are represented as vector in an n-dimensional space. Thus this model is *algebraic*. In the probabilistic model, the frame work for modeling document and query representation is based on probability theory. Thus, this model is probabilistic [29].

Vector space model is the most popularly used model despite its curse for dimensionality. The theoretical foundation of search engines is the vector space model. Vector Space model is very flexible since each term can be individually weighted, allowing that term to become more or less important within a document or entire document collections as a whole. By applying the different similarity measure to compare queries to term and documents properties of the document collection can be emphasizes or deemphasized [6].

As clustering depends on similarity between data sets, the basic question is what similarity is? In this case, the similarity is distance or closeness between two vectors. For this different similarity measures have been defined. Some of the important measures are: Euclidian distance, dot product, cosine, Jaccard coefficient etc.

#### ❖ **Scope of Dissertation:**

My work is divided in three parts:

- I have tried to identify and compare various similarity measures that have been used to cluster text documents in vector space model.
- I have tried to study and analyze various clustering algorithms that have been used for text document clustering.
- Finally I have done some experiments on document clustering to see the effect of various similarity measures.

In this chapter we discuss briefly about Vector space model in Section 1.2, similarity determination in section 1.3, document clustering in section 1.4 and cluster quality in section 1.5.

## 1.1 Vector Space Model

Vector space is basic object of linear algebra. It is a collection of objects. It is called linear space also. Vector space model (VSM) is the collection vector spaces [6].

The vector space model is used here for document representation. It involves constructing a vector which represents terms in the document. In the vector space model, a document is conceptually represented by a vector of key words extracted from the documents with associated weights representing the importance of the key words in the documents and within the whole document collection [6]. Likewise a query (vector) is model is a list of key words with associated weights representing the importance of the key word in the query. The weight of a term in a document vector can be determined in many ways. Once the terms weight are determined a similarity measure can compare similarity between query and document. Based on the similarity score, a ranking function can return the ranked list of relevant documents.

In the vector space model each document is considered to be a vector. To understand the representation of documents in vector space model consider the following definitions:

- 'n' = number of distinct terms.  
The 'orthogonal' terms form a vector with dimensions 'n'.  
'n' term means an n-dimensional vector will be formed.
- $tf_{ij}$  = number of occurrences of term  $t_j$  in document  $D_i$ .
- $df_j$  = number of documents that contain term  $t_j$ .
- $idf_j = \log\left(\frac{d}{df_j}\right)$  where  $d$  is total number of documents [inverse document frequency of term  $t_j$ ] [6].
- Each term,  $t_j$ , in document  $D_i$  is given a real-valued weight,  $w_{ij}$  as follows :

$$w_{ij} = 1 \times tf_{ij} \times idf_j$$

The weight  $w_{ij}$  for a term  $t_j$  in document  $D_i$  is the combination of the term frequency ( $tf$ ) and inverse term frequency ( $idf$ ).

- Further each document  $D_i$  is represented as collection of term weights  $D_i = (w_{i1}, w_{i2}, w_{in})$ .
- Queries are treated like documents.

There can be two ways of viewing vector space model for given document set. One is document vector space (each document is represented by weights of terms), other is term document space (each term is represented as the weight of terms in each document also called inverted index).

**Let now explain the vector space model by an example.** In this example we explain representation of the documents in vector space model and formation of similarity matrix with the explanation of term frequency, inverse term frequency and term weight calculation.

There are three documents and one query.

$D_1$  : “ Computer Science And Engineering”.

$D_2$  : “Computer Science And System Science”.

$D_3$  : “Information Technology”.

Q : “Science And Technology”.

The number of documents  $n = 3$ ,

If the term appears in only one of the three documents, its *idf* is  $\log\left(\frac{d}{df}\right) = \log\left(\frac{3}{1}\right) = 0.477$ .

Similarly if a term appears in two documents then *idf* is  $\log\left(\frac{3}{2}\right) = 0.176$  and if a term appear in the three document then *idf* is  $\log\left(\frac{3}{3}\right) = 0$ .

Arranging the terms alphabetically inverse term frequency of each term (t) in the documents is given

$idf_1$  (And) = 0.176  $\rightarrow t_1$ ,

$idf_2$  (Computer) = 0.176  $\rightarrow t_2$ ,

$idf_3$  (Engineering) = 0.477  $\rightarrow t_3$ ,

$idf_4$  (Information) = 0.477  $\rightarrow t_4$ ,

$idf_5$  (Science) = 0.176  $\rightarrow t_5$ ,

$idf_6$  (System) = 0.477  $\rightarrow t_6$ ,

$idf_7$  (Technology) = 0.477  $\rightarrow t_7$ ,

The terms frequency ( $tf$ ) in the documents is given,

Doc id	$tf_{1j}$	$tf_{2j}$	$tf_{3j}$	$tf_{4j}$	$tf_{5j}$	$tf_{6j}$	$tf_{7j}$
D <sub>1</sub>	1	1	1	0	1	0	0
D <sub>2</sub>	1	1	0	0	2	1	0
D <sub>3</sub>	0	0	0	1	0	0	1
Q	1	0	0	0	1	0	1

**Table 1.1:** Term Frequency matrix for documents and query.

The weight for term  $i$  in vector  $j$  is computed as the  $W_{ij} = tf_i \times tf_{ij}$ .

The document vector is constructed .There are seven terms. The seven dimension vector is constructed. The alphabetical order of terms is considered the document vector.

Doc id	$w_{1j}$	$w_{2j}$	$w_{3j}$	$w_{4j}$	$w_{5j}$	$w_{6j}$	$w_{7j}$
D <sub>1</sub>	0.176	0.176	0.477	0	0.176	0	0
D <sub>2</sub>	0.176	0.176	0	0	0.352	0.477	0
D <sub>3</sub>	0	0	0	0.477	0	0	0.477
Q	0.176	0	0	0	0.176	0	0.477

**Table 1.2:** Weights of terms in Vector space model.

If we consider the above matrix row wise we get representation in document term vector space , whereas if we consider the matrix column wise we get representation in term document vector space. The selection of one of the representation depends on the need of the required application. For example for

matching of query and documents term document vector space representation is preferred as it is easy to find similarity between query and documents based on terms. Instead if consider document clustering application then document space representation will be preferred as it will be easier to find similarity between the documents.

### **1.1.1. Preprocessing**

In specific document some words carry more meaning than others. Usually noun words are the ones which are most representative of document content. Therefore it is usually considered worthwhile to preprocess the text of the documents in the collection to determine the terms to be used as index terms. The preprocessing of the documents in the collection might be viewed as a process of controlling the size of the vocabulary along with ignoring non discriminating words [20]. It is expected that the use of a controlled vocabulary also leads to an improvement in retrieval performance.

Document preprocessing is a procedure which can be divided mainly in to five text operations or transformation.

- Lexical analysis of the text with the objective of treating digits, hyphens, punctuation marks, and the case of letters.
- Elimination of stop words with the objective of filtering out words with very low discrimination values for retrieval purpose.
- Stemming of the remaining words with the objective of removing affixes(i.e. prefixes and suffixes) and allowing the retrieval of documents containing syntactic variations of query terms (e.g. connect, connecting, connected, connection etc will be stemmed to give single word connect).
- Selecting of index terms to determine which words/stems or groups of words will be used as an indexing element. In fact noun words frequently carry more semantics than adjectives, adverbs, and verbs.
- Construction of term categorization structures such as a thesaurus.

## 1.2 Similarity Determination

The **vector space models**, by basing their rankings on similarity measure between the query and terms or documents in the space, are able to automatically guide the user to documents that might be more conceptually similar and of greater use than other documents [6].

The similarity measures that are accurate and provide good retrieval performance are more used by the search engines. The traditional method for determining closeness (similarity) of two vectors is to use the size of the angle between them. This angle is computed by the use of the inner product. However, it is not necessary to use the actual angle. Several different means of comparing a query vector  $Q$  with a document vector  $D_i$  and finding out the Similarity Coefficient (SC) have been implemented. Before discussing similarity measure let us define some variables used:

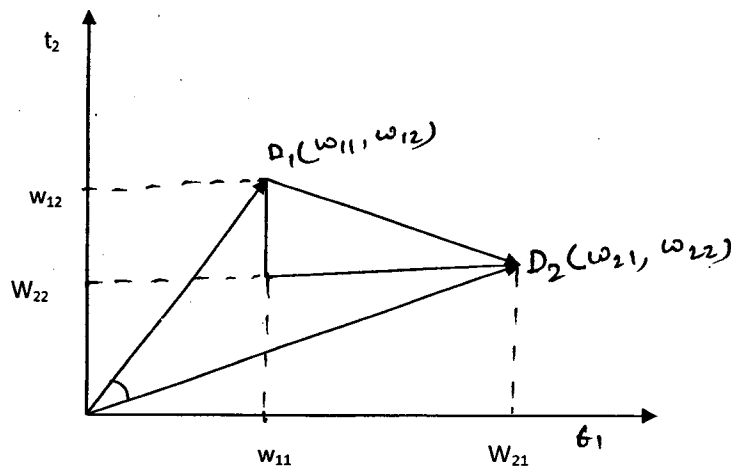


Fig. 1.1: Two Documents with Two Dimensions [20].

$t$  : Number of terms in vector space.

$w_{pj}$  : Weight of the term  $j$  in the document  $p$ .

$w_{qj}$  : Weight of term  $j$  the document  $q$ .

The representation is shown graphically in Fig. 1.1. Here we have considered two documents with two dimensions corresponding to two terms  $t_1$  and  $t_2$ .

Based on the above definitions now we discuss some of the important similarity measures used with VSM.

- **Inner Product or Dot Product**

Simply Dot product of two vectors  $D_p$  and  $D_q$  calculated as :

$$SC(D_p, D_q) = \sum_{j=1}^t w_{qj} \times w_{pj}$$

Measures how many terms are matched but don't measure how many terms are not matched [6, 20].

This similarity measure is not normalized and favors long documents with large number of unique terms.

The dot product can easily be explained by considering the previous example that is discussed in 1.1 section. The three documents are there  $D_1, D_2, D_3$ , and query document  $Q$ . The similarity of each document is calculated with the query relevance rank is decided on the basis of similarity.

The similarity:

$$SC(Q, D_1) = (0.176)(0.176) + (0)(0.176) + (0)(0.477) + ((0)(0) + (0.176)(0.176) + (0)(0) + (0.477)(0) = 2 \times (0.176)^2 \approx 0.062.$$

For  $D_2$

$$SC(Q, D_2) = (0.176)(0.176) + (0)(0.176) + (0)(0) + (0)(0) + (0.176)(0.352) + (0)(0.477) + (0.477)(0) = (0.176)^2 + (0.176)(0.352) \approx 0.0928.$$

Similarly

$$SC(Q, D_3) = (0.477)(0.477) = (0.477)^2 = 0.228.$$

Hence the ranking of the documents will be  $D_1, D_2$ , and  $D_3$ .

- **Cosine Measure**

The **cosine** measure gives the cosine of the angle between the query and document vector. It is the most commonly used similarity measure [6]. It divides the dot product by the length of the document vector.

$$SC(D_p, D_q) = \frac{\sum_{j=1}^t w_{qj} \times w_{pj}}{\sqrt{\sum_{j=1}^t (w_{qj})^2 \sum_{j=1}^t (w_{pj})^2}}$$

The cosine measure provides a similarity measure between 0 and 1. The cosine measure captures a scale invariant understanding of similarity. Also, a stronger



property is that it does not depend on the length of the documents (as every keyword vector is normalized). This allows documents with the same composition, but different totals to be treated identically. These properties make Cosine the most popular similarity measure for text documents [6].

The denominator in this equation, called the normalization factor, discards the effect of document lengths on document scores. Thus, a document containing {x, y, z} will have the same score as another document containing {x, x, y, y, z, z} because these two document vectors have the same unit.

- **Dice Coefficient**

**Dice coefficient** is defined as twice the number of terms common to compared entities divided by the total number of terms in both tested entities.

Dice Coefficient = (2\* Common Terms) / (Number of terms in String1 + Number of terms in String2)

$$SC(D_p, D_q) = \frac{2 \sum_{j=1}^t w_{pj} \times w_{qj}}{\sum_{j=1}^t (w_{qj})^2 + \sum_{j=1}^t (w_{pj})^2}$$

The coefficient result of 1 indicates identical vector where as a 0 equals orthogonal vectors.

- **Jaccard Coefficient**

**Jaccard** similarity uses word sets from the comparison instances to evaluate similarity.

$$SC(D_p, D_q) = \frac{\sum_{j=1}^t w_{pj} w_{qj}}{\sum_{j=1}^t (w_{qj})^2 + \sum_{j=1}^t (w_{pj})^2 - \sum_{j=1}^t w_{pj} w_{qj}}$$

It measures the degree of overlap between two sets. The Jaccard similarity penalizes a small number of shared entries (as apportion of all non-zero entries) more than the Dice Coefficient.

- **Matching Coefficient**

The Matching Coefficient is a very simple vector based approach which simply counts the number of terms (Dimensions), on which both vectors are non zero. So for vector set X and set Y:

$$\text{Matching Coefficient} = |X \& Y|$$

This can be seen as the vector based count of referent terms. This is similar to the vector version of the simple hamming distance although position is not taken into account.

- **Euclidean Distance**

This measure finds the dissimilarity or the distance measure between any two documents. The direct Euclidean distance between the vector inputs is measured as:

$$SC (D_p, D_q) = \sqrt{\sum_{j=1}^t (wp_j - wq_j)^2}$$

It is called L2 Distance measure also. This approach again works in vector space similar to the matching coefficient and the dice coefficient, however the similarity measure is not judged from the angle as in cosine rule [6].

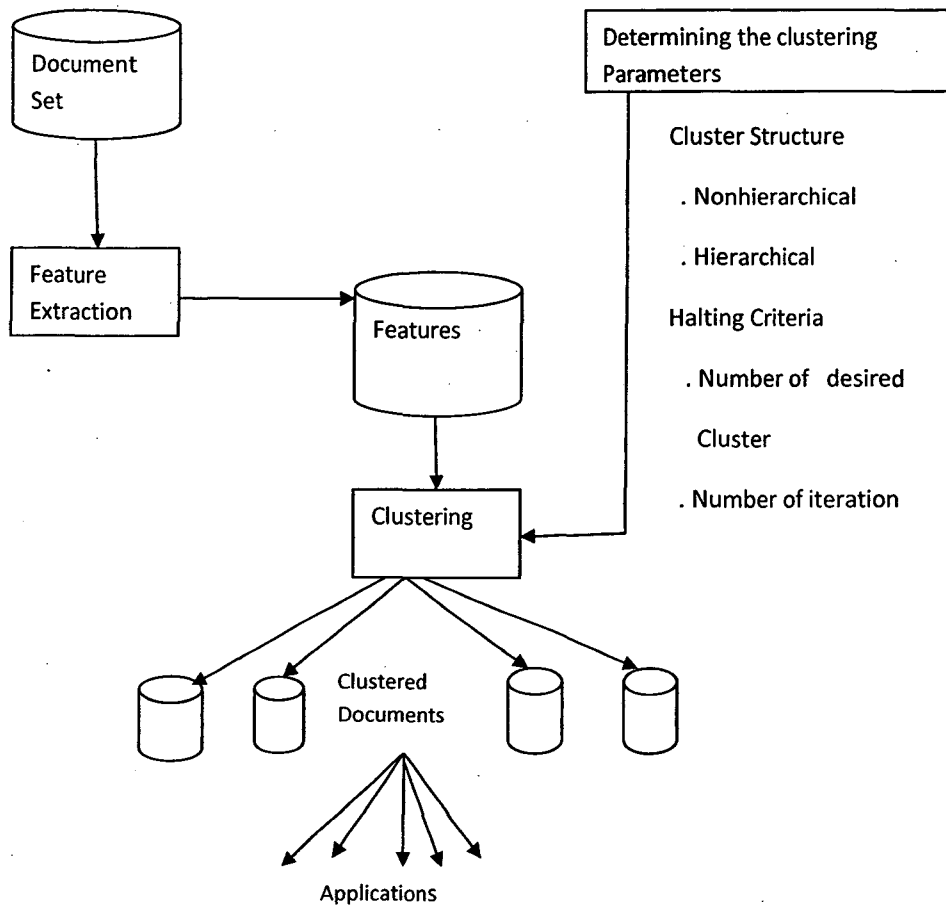
It has been observed that cosine is the most popularly used Euclidean is not preferred as a similarity measure for comparing documents.

### 1.3 Document Clustering

**Clustering** is generally described as the task of segmenting a heterogeneous population into a set of homogeneous subgroups [20, 22]. Clustering approaches are commonly used for segmentation. Clustering algorithms allow entities described by a large number of attributes to be partitioned into a few distinct groups or “segments”.

Clustering segments the population into classes on the basis of the similarity between the class members. It also produces a high level description of the population applying distance measures between its elements. Document clustering has been investigated as process for improving document search or organization, information retrieval and automatic key extraction [7].

**Block Diagram of document clustering:** The document clustering can easily be explained by the block diagram. The diagram explains the steps included in the document clustering.



**Fig. 1.2:** Block diagram of document clustering.

Example of document clustering:

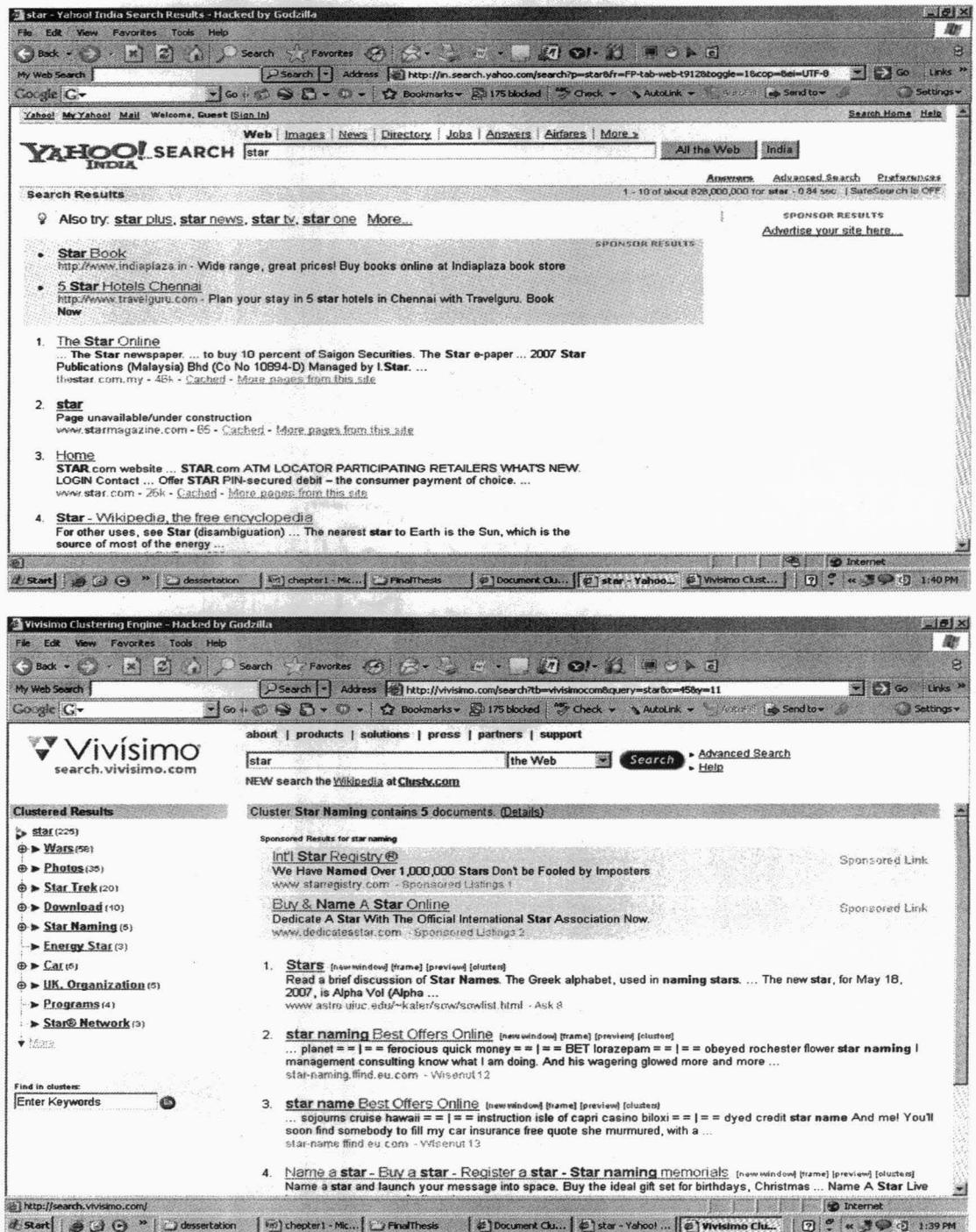


Fig. 1.3: Screens shot of yahoo and Vivisimo search Engine for the query of “star”.

To appreciate the role of clustering let us consider a practical example from results of two popular search engines. Here there are two search engines yahoo and vivisimo as shown in Fig. 1.3. The query is “star”. The yahoo search engine gives single results related to the star query. Vivisimo search engine gives clustered result with the different meanings of query giving different clusters. The results in vivisimo are grouped according the meaning of the query. As the “star” query have different meaning, with each meaning it has different cluster. The user may select the most relevant cluster. The vivisimo initially shows 225 documents (results) related to star and categorized according the different meaning of star and forms 10 clusters. As can be seen easily it is difficult to find the relevant documents in yahoo search engine result, in comparison the vivisimo search engine. Thus we have seen how clustering reduce the search speed as well as increase efficiency.

#### 1.4 The Measure of Cluster Quality

The goodness and cluster quality can be decided by the two measures, internal quality measure and external quality measure. To compare the different set of clusters without any external information is known as internal quality measure, like “over all similarity” measure [1, 22]. External measure allows us to evaluate how well clustering is working by comparing groups produced by clustering techniques to known classes. Popular external measures are: entropy and F-measure.

- **Entropy:** It is an external measure. It measures the “goodness” for un-nested cluster. The cluster contains the single data point, the best entropy obtained. Here the class distribution of the data is calculated first, by using this class distribution the entropy of each cluster is calculated. The entropy of cluster  $j$  :

$$E_j = - \sum_i p_{ij} \log(p_{ij})$$

Where  $p_{ij}$  is the probability that a member of cluster  $j$  belongs to class  $i$ .

The total entropy for a set of cluster is calculated as the sum of the entropies of each cluster weighted by the size of each clusters:

$$E_{cs} = \sum_{j=1}^m \frac{n_j \times E_j}{n}$$

Where  $n_j$  is the size of cluster  $j$ ,  $m$  is the number of clusters and  $n$  is the total number of data points.

- **Overall Similarity:** When the class level information is not available, the cohesiveness of cluster can be used as a measure of cluster similarity. The

cohesiveness of the cluster is measure by the weighted similarity of the internal cluster similarity [21]. As

$$\frac{1}{|S|} \sum_{d \in S} \sum_{d' \in S} \text{Cosine}(d', d) = \frac{1}{|S|} \sum_{d \in S} d \cdot \frac{1}{|S|} \sum_{d \in S} d = \text{c.c} \|c\|^2$$

- **F-measure:** It measures the effectiveness of the cluster. It is joint venture of recall and precision. For f-measure the recall and precision are calculated of the cluster for a class. Then find out the f-measure of the cluster. The F-measure of cluster j and class I is the given by:

$$F(i, j) = (2 * \text{Recall}(I, j) * \text{precision}(I, j)) / ((\text{precision}(I, j) + \text{Recall}(I, j)))$$

For entire hierarchical clustering the F measure of any class is the maximum value it attains at any node in the tree. An overall value for the F measure is computed by taking the weighted average of all values for the F measure as given by the following.

$$F = \sum_i \frac{n_i}{n} \max \{F(I, j)\}$$

Where max is taken over all clusters at all levels, and n is the number of documents.

## Approaches of Document Clustering

---

One of the main purposes of clustering documents is to quickly locate relevant documents. In an ordered collection, the end user must scan individual documents for relevance. For large collections this is time consuming and tedious. When a collection is organized into clusters it is easier to search a smaller set of clusters for relevance. After a relevant cluster has been identified, all its member documents are likely to be relevant as well.

Document clustering has been investigated for use in a number of different areas of text mining and information retrieval. Initially, document clustering was investigated for improving the precision or recall in information retrieval system and as an efficient way of finding the nearest neighbors of document. More recently, clustering has been proposed for use in browsing a collection of documents or in organizing the results returned by a search engine in response to a user's query [23, 25]. Document clustering has also been used to automatically generating hierarchical clusters of documents.

Documents are identified by its keywords. A keyword may be found in one or more documents. In Keyword based clustering a cluster of documents share a set of keywords that co-occur in document text [7]. Documents in a cluster may possibly be related to a common topic and are represented by a sequence of keywords and their associated weights.

When a document is modeled using document-term frequency matrix representation or its variants (as in VSM) the relative ordering of words in text gets lost. Thereby syntactic information of formation of text, such as grammar for sentence structure also disappears. In spite of this, the term frequency modeling has been found to be very effective in text or document retrieval such as query processing, clustering document analysis etc [4, 20].

Once a document is represented in document-term frequency matrix model it can be represented in vector space. We can apply a vector based similarity measure such as

dot product, cosine, Euclidean, Jaccard coefficient etc. to find similarity of two documents. Document can then be clustered based on these similarity measures.

When we use VSM the dimensionality of vector space is equal to number of unique keywords in document set. Thus the dimensionality of representation is very large [6]. By using preprocessing techniques such as stemming and stop word removal dimensionality is reduced to a certain extent. Further Latent Semantic Indexing can also base to decrease dimensionality.

Many of the traditional clustering algorithms such as K-means, Hierarchical agglomerative clustering methods have been used for document clustering. However the algorithms defined for working on large data sets may give better results [1, 22, 30]. The approaches which use clustering based on phrases rather than words generally give better results [12]. Further machine learning techniques can be used to develop efficient document clustering algorithms [35]. In this chapter we firstly explain the working of traditional clustering algorithms and discuss there application for document clustering. In later sections we discuss some other methods that have been applied for document clustering.

## **2.1 Traditional Clustering Approaches**

Document clustering was initially used to improve information retrieval performance. The documents retrieved by IR systems were grouped based on similarity between documents [3]. In the coming subsection we discuss Partitional clustering algorithms specifically K-means and Bisecting K-means algorithm and its application in document clustering.

### **2.1.1 Partitional Clustering algorithms**

Partitioning algorithms had been popular clustering algorithms long before the emergence of data mining. Given a set  $D$  of  $n$  objects in a  $d$ -dimensional space and an input parameter  $k$ , a partitioning algorithm organizes the objects into  $k$  clusters such that the total deviation of each object from its cluster center or from a *cluster distribution* is minimized. The deviation of an object from the cluster center is commonly computed using a *similarity function*. There are many partitioning methods such as  $k$ -mean algorithm [1, 6, 19, 34, 35], Bisecting  $k$ -means, EM



(Expectation Maximization) [35] algorithm [28], PAM (Partition around Medoid,  $k$ -medoid) algorithm [19], CLARA [19], CLARANS [34] etc.

The partitioning algorithms generally start with an initial partition of the data set and then uses an iterative control strategy to optimize an objective function. Each cluster is represented by the center of gravity of the cluster ( $k$ -mean algorithms) or by one of the objects of the cluster located near its center ( $k$ -medoid algorithms) [6, 26, 34, 35]. Partitioning algorithms use a two-step procedure. First, determine  $k$  representatives minimizing the objective function. Second assign each object to the cluster with its representative “closest” to the considered object. The second step implies that a partition is equivalent to a Voronoi diagram and each cluster is contained in one of the Voronoi cells [3]. Thus the shape of all clusters found by a partitioning algorithm is convex which is very restrictive. Partitions can be found two ways. The process can start with each document of its own grouped together until the number of partitions is suitable, this is called *bottom-up* clustering [1, 5, 9, 22]. A process where the number of portions is assigned prior, that is called *top-down* clustering. This is an iterative process the repeats until the defined terminating condition is true.

#### 2.1.1.1 K-Means Clustering Algorithm

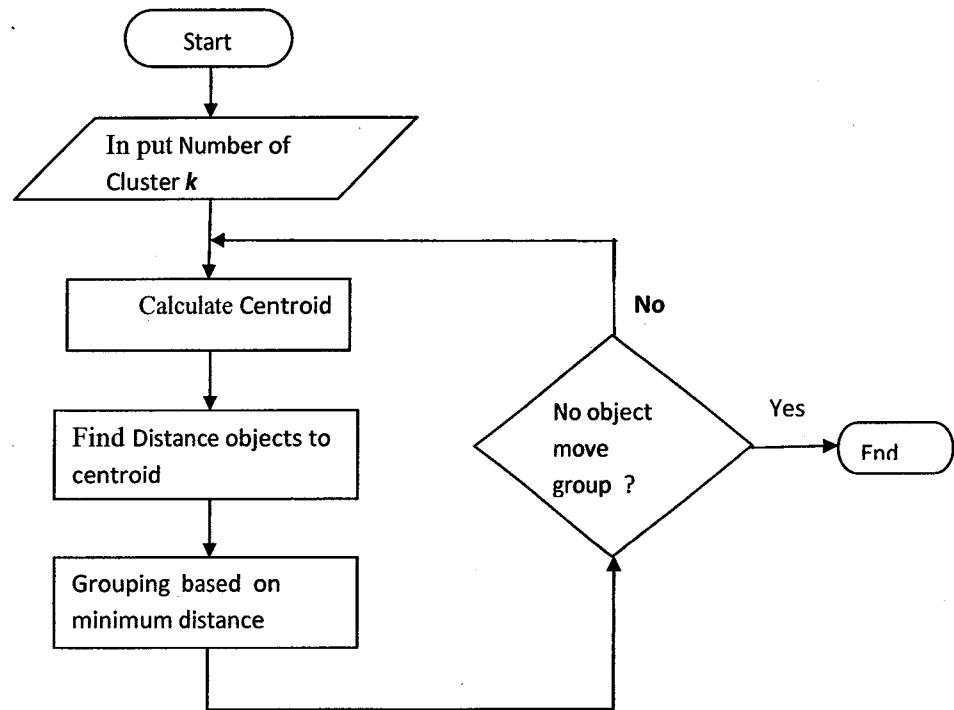
**K-means** is partitional clustering algorithm. It classifies a given data set into  $k$  number of clusters, fixed initially. Where  $k$  is a positive integer number during each partition, the centroids or means of clusters are computed [35]. The steps and flow chart of basic  $k$ -means clustering algorithm for finding  $k$  clusters are as follows.

1. Select  $k$  points as the initial centroids. Called  $C_i$
2. Assign all points  $X_k$  to the cluster that has the closest centroid  $U_i$ .
3. Recalculate the centroid of each cluster using

$$C_i = \frac{\sum_{X \in U_i} Xk}{n}$$

Where  $n$  is the number of objects in cluster  $U_i$ .

4. Repeat step 2 and step 3 until the centroids do not change that is there are no more assignments.



**Fig. 2.1:** Flow Chart of K-means.

The k-means algorithm is discussed by the numerical example.

**Example:** Suppose there are four jobs each object has two attribute feature as shown in table. My goal is to cluster these objects into 2 clusters. Let the table

Object	Attribute 1(x): weight index	Attribute 2(y): ph
Medicine A	1	1
Medicine B	2	1
Medicine C	4	3
Medicine D	5	4

Each medicine represents the point with tow attributes (x, y).

The data can be represented in matrix form as follows:

$$\begin{array}{cccc} & \text{A} & \text{B} & \text{C} & \text{D} \\ \left[ \begin{array}{cccc} 1 & 2 & 4 & 5 \\ 1 & 1 & 3 & 4 \end{array} \right] & & & & \begin{array}{l} \text{X} \\ \text{Y} \end{array} \end{array}$$

Initial value of centroid, suppose medicine A medicine B as the first centroids  $C_1$  and  $C_2$  have the coordinates  $C_1 = (1, 1)$  and  $C_2 = (2, 1)$

Object centroid distance is calculated by using Euclidean distance. The column in the distance matrix represents the objects. First row of the distance matrix show the of the each object to the first centroid and second corresponding to second centroid.

Distance  $C_1$  to C (4, 3)

$$= \sqrt{(4-1)^2 + (3-1)^2} = 3.61$$

Distance  $C_1$  to D (5, 4)

$$\begin{aligned} &= \sqrt{(5-1)^2 + (4-1)^2} \\ &= 5 \end{aligned}$$

Distance  $C_2$  to C(4,3)

$$= \sqrt{(4-2)^2 + (3-1)^2} = 2.83$$

Distance  $C_2$  to D (5, 4)

$$\begin{aligned} &= \sqrt{(5-2)^2 + (4-1)^2} \\ &= 4.24 \end{aligned}$$

The distance matrix  $D^0$  is given

$$D^0 = \begin{bmatrix} 0 & 1 & 3.61 & 5 \\ 1 & 0 & 2.83 & 4.24 \end{bmatrix} \begin{array}{l} C_1 = (1, 1) \text{ group - 1} \\ C_2 = (2, 1) \text{ group - 2} \end{array}$$

A   B   C   D

First row of distance matrix gives distance of each element from first cluster, whereas second row gives distance of each element from second cluster.

The object clustering: Assign the group based on the minimum distance. That's by Medicine A come in group-1 and B,C,D fall in group-2 .

$$G^0 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \end{pmatrix} \begin{array}{l} \text{group-1} \\ \text{group-2} \end{array}$$

Iteration-1, determine centroid: Here the three objects fall in group-2, and only in one in group-1.

So centroid  $C_1$  remain same but  $C_2$  change. The new  $C_2$ ,

$$= \left( \frac{2+4+5}{3}, \frac{1+3+4}{3} \right)$$

$$C_2 = \left( \frac{11}{3}, \frac{8}{3} \right)$$

Iteration -2, object centroid distance: The distance matrix formed by the new centroid.

$$D^1 = \begin{pmatrix} 0 & 1 & 3.61 & 5 \\ 3.14 & 2.36 & 0.47 & 1.89 \end{pmatrix} \begin{array}{l} C_1 = (1, 1) \text{ group - 1} \\ C_2 = \left( \frac{11}{3}, \frac{8}{3} \right) \text{ group - 2} \end{array}$$

Objects clustering: Assign the group based on minimum distance in distance matrix.

$$G^1 = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix} \begin{array}{l} \text{group-1} \\ \text{group-2} \end{array}$$

Iteration 3, determine centroids: the new centroid calculated according the previous steps. The new centroids are

$$C_1 = \left( \frac{1+2}{2}, \frac{1+1}{2} \right) = \left( \frac{3}{2}, 1 \right), \quad C_2 = \left( \frac{4+5}{2}, \frac{3+4}{2} \right) = \left( \frac{9}{2}, \frac{7}{2} \right)$$

Object centroid distances: calculate distance and form distance matrix considering the new centroid.

$$D^1 = \begin{bmatrix} 0.5 & 0.5 & 3.20 & 4.61 \\ 4.30 & 3.54 & 0.71 & .71 \end{bmatrix} \quad C_1 = \left(\frac{3}{2}, 1\right) \text{ group - 1}$$

$$C_2 = \left(\frac{9}{2}, \frac{7}{2}\right) \text{ group - 2}$$

Object clustering: Assign each object based on the minimum distance.

$$G^2 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \quad \begin{array}{l} \text{group-1} \\ \text{group-2} \end{array}$$

A    B    C    D

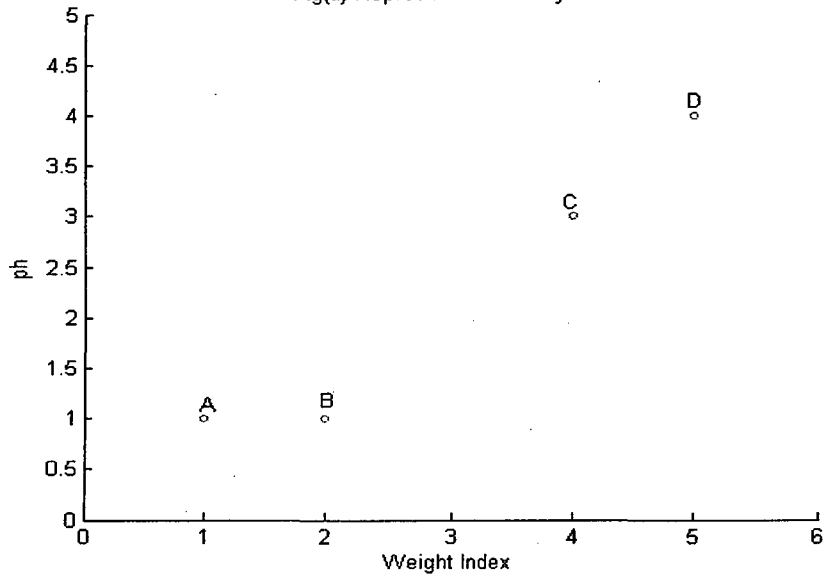
The final result is that  $G^2 = G^1$ . The cluster (group) in the last iteration is same as the first one. Thus K-means reached on the stability. There are no more iteration is needed. The Final grouping:

Object	Attribute 1 (X): weight index	Attribute 2 (Y): ph	Group
Medicine A	1	1	1
Medicine B	2	1	1
Medicine C	4	3	2
Medicine D	5	4	2

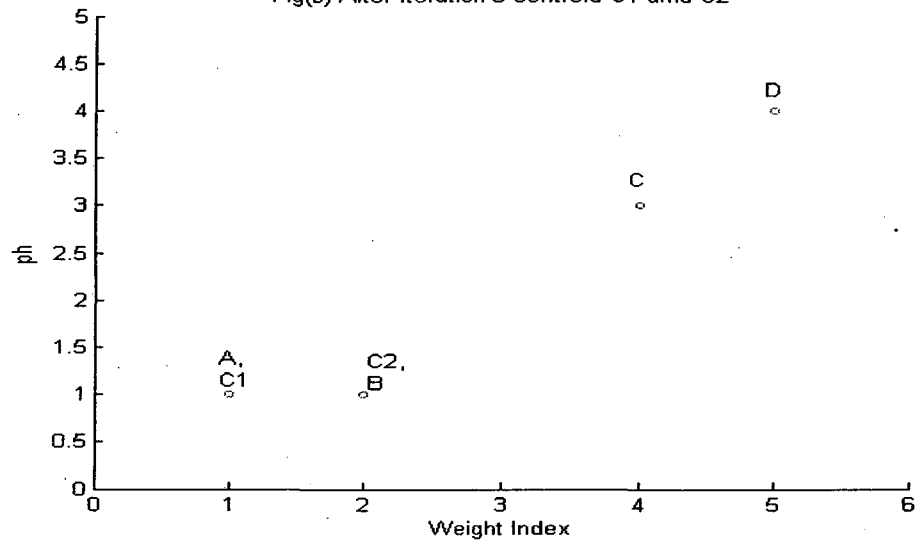
74-14684

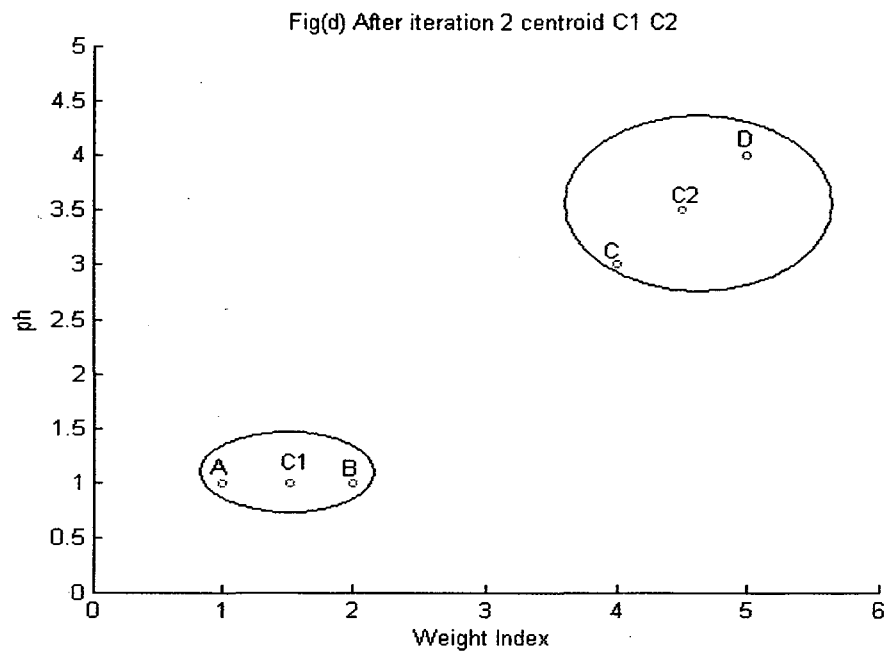
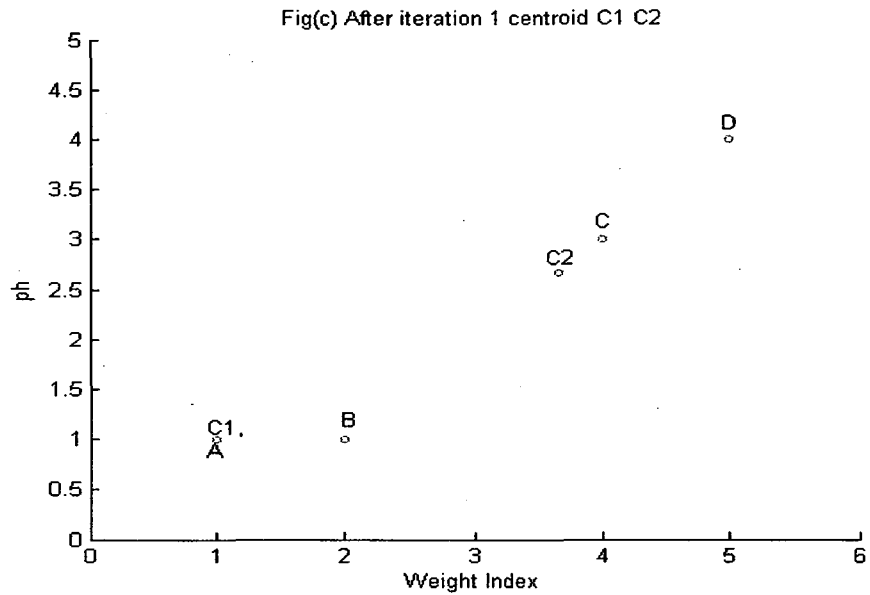


Fig(a) Representation of objects



Fig(b) After iteration 0 centroid C1 and C2





The graphical representation of the k-mean clustering algorithm is show in the above graph. The graph based on the above example.

This method can easily be applied for document clustering. Consider following documents:

D1 : Computer Science.

D2 : Life Science.

D3 : Computer Science and Communication Science.

D4 : Computer and Computer Network.

Now if the documents are to be clustered. The attributes can be considered as frequency of word (or it's variant). Thus we can have following table:

Doc	Computer	Science	Life	Communication	Network
D1	1	1	0	0	0
D2	0	1	1	0	0
D3	1	2	0	1	0
D4	2	0	0	0	1

Now K means can be applied to cluster the documents as discussed above.

#### 2.1.1.2 Bisecting K-means Clustering Algorithm

The basic k-means algorithm can be enhanced. One way of enhancing it is using Bisecting K means [22]. The working procedure of this algorithm is following steps.

- Pick a cluster to split.
- Find the two sub cluster using the basic k-means algorithm.--> Bisecting step
- Repeat step 2, for ITER times and take the split that produces the clustering with the highest overall similarity.
- Repeat step 1, 2 and 3 until the desired number of clusters is reached.

For choose the splitting cluster there are many ways. But here, some methods are given. Like the largest cluster, or size and overall similarity [1, 22].

The bisecting k-means algorithm can produce both type clustering like un-nested or hierarchical clustering. Un-nested clusters are refined using basic k-means. But nested clusters are not refined. The time complexity of bisecting K-means algorithm is linear in the number of document [1, 11, 22, 26]. If the number of cluster is large



and refinement is not used, then bisecting k-means is more efficient than the regular k-means algorithm. (In this case, there is no need to compare every point to every cluster centroid since to bisect a cluster we just consider the points in cluster and their distance to two centroids).

### 2.1.2 Hierarchical

Hierarchical algorithms create a hierarchical decomposition of the dataset  $D$ . The hierarchical decomposition is represented by a dendrogram, a tree that iteratively splits dataset into smaller subsets until each subset consists of only one object [1, 5, 9, 22]. In such a hierarchy each node of the tree represents a cluster of  $D$ . The dendrogram can either be created from the leaves up to the root (agglomerative approach) or from the root down to the leaves (divisive approach) by merging or dividing clusters at each step. In contrast to partitioning algorithms, hierarchical algorithms do not require  $k$  as an input [1, 22]. However, a termination condition is necessary for controlling the termination of merger or division. One example of a termination condition in the agglomerative approach is the critical distance  $d_{min}$  between all the clusters of  $D$ . The basic steps used in agglomerative hierarchical clustering algorithm as follow:

1. Compare the similarity between all pairs of clusters (calculate a similarity matrix and find the similarity (distance) between two cluster).
2. Merge the closest two clusters.
3. Update the similarity matrix to show the pair wise similarity between the new cluster and previous cluster.
4. Repeat the steps 2 and 3 until only a single cluster formed or there is a critical distance  $d_{min}$  between all the clusters of  $D$ .

The main problem with hierarchical clustering algorithms so far has been the difficulty of deriving appropriate parameters for the termination condition, e.g. a value of  $d_{min}$  which is small enough to separate all “natural” clusters and, at the same time large enough such that no cluster is split into two parts. Eycluster [5,22] hierarchical algorithm presented in the area of signal processing automatically derives a termination condition. Eycluster follows the divisive approach. Experiments show that it is very effective in discovering non-convex clusters.

However, the computational cost of Ecluster is  $O(n^2)$  due to the distance calculation for each pair of points [14]. This is acceptable for applications such as character recognition with moderate values for  $n$ , but it is prohibitive for applications on large dataset.

Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH) [2, 6] has the complexity  $O(n)$  using a hierarchical data structure called CF-tree for multiphase clustering. In BIRCH, a single scan of the dataset yields a good clustering and one or more additional scans can be used to improve the quality of cluster further. However, it handles only numerical data and it is order-sensitive. Also, BIRCH does not perform well when the clusters do not have uniform size and shape since it uses only the centroid of a cluster when redistributing the data points in the final phase.

Clustering Using Representatives (CURE) [6, 38] employs a combination of random sampling and partitioning to handle large databases. It identifies clusters having non-spherical shapes and wide variances in size by representing each cluster by multiple points. The representative points of a cluster are generated by selecting well-scattered points from the cluster and shrinking them toward the centre of the cluster by a specified fraction. However, CURE is sensitive to some parameters such as the number of representative points, the shrink factor used for handling outliers, number of partitions. Thus, the quality of clustering results depends on the selection of these parameters.

RObust Clustering using linKs (ROCK) [5, 6, 9] is a representative hierarchical clustering algorithm for categorical data. It introduces a novel concept called “link” in order to measure the similarity/proximity between a pair of data points. Thus, the ROCK clustering method extends to non-metric similarity measures that are relevant to categorical data sets. It also exhibits good scalability properties in comparison with the traditional algorithms employing techniques of random sampling [38]. Moreover, it seems to handle successfully data sets with significant differences in the sizes of clusters.

#### **Example of Document Clustering:**

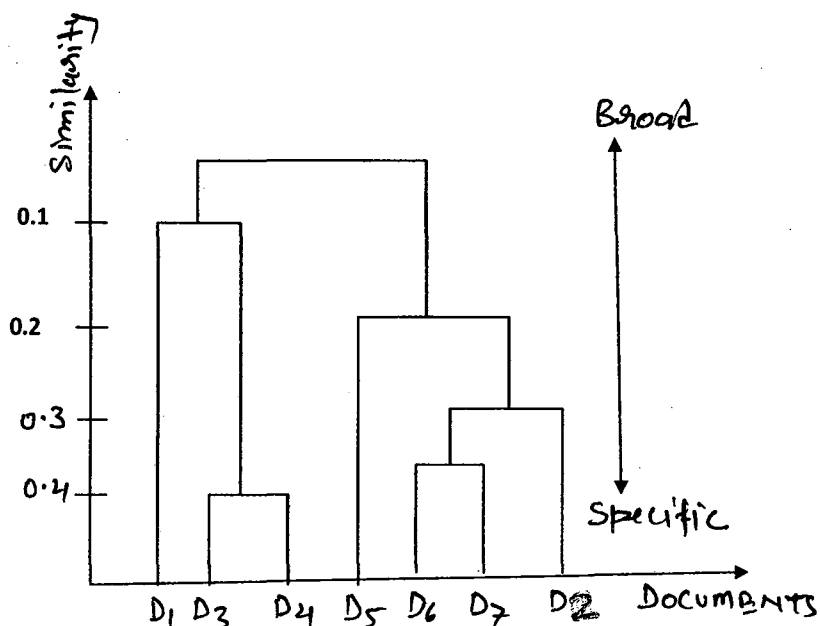
Consider the collection of seven documents in table. We can arrange this small collection of documents in a dendrogram. At the lowest level on the  $y$ - axis

documents are paired with the most similar documents in the collection. Tree is built by collapsing nodes in decreasing order of similarity [20].

The highest similarity is between documents  $D_3$  and  $D_4$ . If we combine the documents  $D_3$  and  $D_4$  in to a composite document, then the next highest similarity is between  $D_6$  and  $D_7$ . This process continues till all the documents are absorbed into a single composite document.

No	Text
$D_1$	Human machine interface for computer applications
$D_2$	A survey of user opinion of computer system response time
$D_3$	The EPS user interface management system
$D_4$	System and human system engineering testing of ESP
$D_5$	The generation of random, binary and order tree
$D_6$	The intersection graph of paths in trees
$D_7$	Graph minors: survey

**Table2.1:** A Sample Collection of seven Documents [20]



**Fig.2.2:** A Dendrogram for a collection of seven documents [20]

## 2.2 Approaches For Clustering Large Data Sets

As we have observed the main problem regarding VSM, it is the large dimension therefore approaches for clustering large data set may work effectively for VSM.

Kaufman and Rousseeuw in 1990 proposed CLARA (Clustering LARge Applications) which relies on sampling to handle large data sets. CLARA draws a sample in random from the data set, applies PAM (Partitioning Around Medoids) to the sample, and finds the medoid of the sample [7, 28, 34, 35]. The quality of clustering at this stage is measured based on the average dissimilarity of all objects in the entire data set, and not only of those objects in the samples. According the experiment reported in [2, 35] indicates that CLARA is more efficient than PAM. The main disadvantage of the CLARA is that, one can't find the best clustering if the any sampled medoid is not among the best  $k$  medoids.

Ng & Han proposed a partitioning algorithm for spatial databases called CLARANS (Clustering Large Applications based on RANdomized Search) [6, 7, 34]. It is an improved  $k$ -medoid method with improved effectiveness and efficiencies. Ng & Han have also discussed methods to determine the "natural" number of clusters  $k$ . CLARANS assumes that all objects to be clustered can reside in main memory at the same time which does not hold for large databases. Furthermore, the run time of CLARANS is prohibitive on large databases.

## 2.3 Phrase Based Clustering

The document clustering may not only be based on single word analysis. It may include phrases. Phrase is the collection of consecutive words that co-occur in a group such as web mining, artificial intelligence. The phrases and their weights are important for document clustering. The phrase matching is done to determine the similarity between documents based on phrase rather than on single word. In other words the similarity calculation between documents is based on a combination of single-term similarity and phrase based similarity [12]. Similarity based on matching phrases between documents, has proven to have a more significant effect on the clustering quality due to its insensitivity to noisy terms that could lead to incorrect similarity measure.

The phrase based document clustering approach is generally based on suffix tree or suffix array clustering. This method adopts the “tri” structure to represent, shared suffix between documents. The base clusters are identified by the shared suffix that is combined in to final cluster based on connected- component graph algorithm [21]. Here, high quality cluster is produced due to the pair-wise document similarity distribution inside each cluster. Then similarities are maximized in each cluster. The clustering approaches which based on the phrased similarity are discussed here.

### 2.3.1 Suffix Array

“A suffix array for a set of string is a lexicographical order of all suffixes of the strings. Two sequences have a matching block if they have contiguous segments that match exactly. A matching block is maximal for a pair of sequences if the sequences differ immediately beyond each point of the block [17, 29].

For suffix array some parameters are defined like  $k$  specifies the length of the shortest matching blocks that the algorithm will detect. A  $K$ -clique is the set of all sequence that have  $k$  length specific matching block. A *score* of a pair of sequences is a numeric measure of their similarity [17, 33]. The score of a set of matching is the sum of the block length and the score of a pair of sequences to the score of the highest scoring consistent set of matching blocks. A consistent set is sub set of blocks that are non-overlapping and same order in the sequences [36]. For actual clustering the minimal score that will cause two sequences to be clustered together. This value referred as clustering threshold. The steps of suffix array algorithm are discussed here:

1. Identify all the matching blocks of length  $k$ :
  - a) Construct all suffixes from the data.
  - b) Sort the suffixes in to a suffix array.
  - c) Group the suffixes that share a prefix of length at least  $k$  into cliques.
  - d) Each clique generate the maximal matching blocks between each pair of suffixes in the clique.
2. Score the resulting sequence pairs:
  - a) For each pair sharing a least one matching block, collect all matching blocks between the two sequences.

- b) Calculate the largest consistent set of matching blocks, and the corresponding score for each pair.
3. Generate the clustering:
- a) Starting with the highest scoring sequence pair and working downward, made clusters hierarchically by connecting sequences.
  - b) Split the clusters according to the clustering threshold.

The general data clustering approaches that considered similarities between multiple objects in order to determine whether to join clusters [36]. Single linkage clustering is the useful algorithm for clustering of sequence fragments Suffix array generation uses a straightforward, left wise radix sort.

### 2.3.2 Suffix Tree Clustering

**Suffix Tree Clustering (STC)** is a linear clustering algorithm. It is based on identifying the phrases that are common to group of documents [25, 8]. The phrase is the ordered sequence of word or words. A base cluster is to be a set of documents that share a common phrase. The steps in suffix tree clustering are following:

#### **Step1. The document -“Cleaning”**

The string of text representation of each document is transformed using a light stemming algorithm (deleting word prefixes in suffixes and reducing plural to singular). Sentence boundaries are marked and non-word token (such as numbers, HTML tags and most punctuations) are striped [25]. The original document string kept as well as pointer from the beginning each word in the transformed string to its position in the original string is also store.

#### **Step2. Identifying Base clustering**

The identification of base clusters can be viewed as the creation of an inverted index of phrases for our document collection. This is done efficiently using a data stricter called a suffix tree.

A suffix tree of a string  $S$  is a compact *tri* containing all the suffixes of  $S$ . We treat the documents as string of words, not characters, thus suffixes contain one or more whole words. In precise terms:

1. A suffix tree is a rooted, directed tree.

2. Each internal node has at least 2 children.
3. Each edge is labeled with a non-empty sub string of  $S$ . The label of a node is defined to be the concatenation of the edge-labels on the path from the root to that node.
4. No two edges out of the same node can have edge-labels that begin with the same word.
5. For each suffix  $s$  of  $S$ , there exists suffix- node whose label equals  $s$ .

The suffix tree of a collection of strings is a compact tree containing all the suffixes of all strings in the collection. Each suffix node is marked to designate from which string it original form.

Each cluster assigned a score that is a function of the number of documents it contains and the word that make up its phrase. The Score  $S(B)$  of base cluster  $B$  with phrase  $P$  is given by:

$$S(B) = |B| \cdot F(|p|)$$

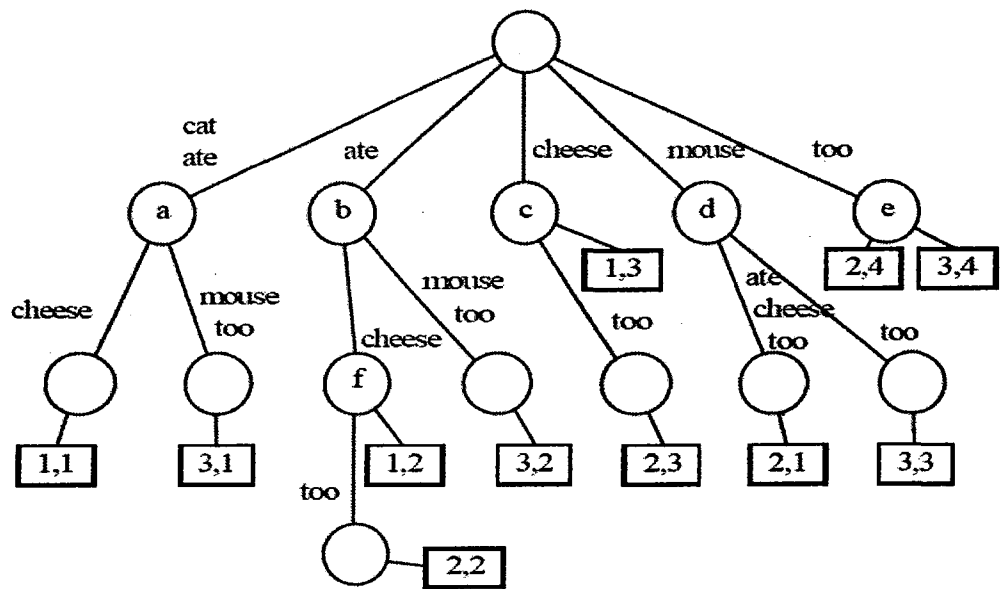
Where  $|B|$  is the number of documents in the base cluster  $B$ , and  $|P|$  is the number of words in that have non-zero score.

### Step 3. Combining Base Cluster

The document may share more than one phrase. As a result, the document sets of distinct base cluster may overlap and may even be identical. To avoid the proliferation of nearly identical cluster, the third step of the algorithm merges base clusters with a highest overlap in their documents set.

**Example:** Consider a set of strings

1. "cat ate cheese"
2. "mouse ate cheese too"
3. "cat ate mouse too".



**Fig. 2.3:** The Suffix tree of the string "cat ate cheese", "mouse ate cheese too" and "cat ate mouse too" [25].

Node	Phrase	Document
a	cat ate	1,3
b	ate	1,2,3
c	cheese	1,2
d	mouse	2,3
e	too	2,3
f	ate cheese	1,2

**Table 2.2:** Six nodes from the example shown in the fig 2.3 and their corresponding base cluster.



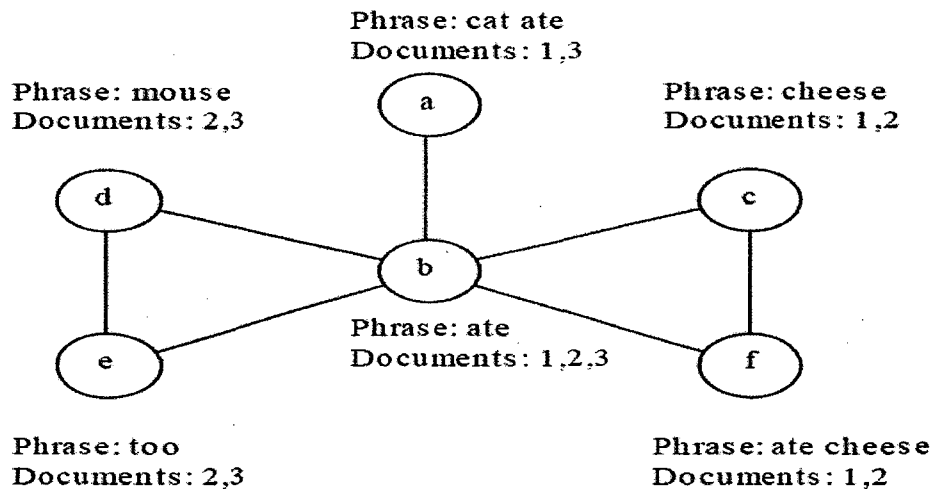


Fig 2.4: The base cluster graph of the example given fig 2.3 and table 2.2 [25].

### 2.3.3 Advantages of Phrase based clustering

Similarity measures are widely used in the information retrieval (IR) community. One of the applications of the similarity measure methods is to determine the similarity among the documents. Most similarity measures do not consider any semantic association (such as synonyms, same stem, etc.) among documents [12, 29]. The inclusion of phrase recognition improves the precision of the retrieval process.

Similarity based on matching phrases between documents is proved to have a more significant effect on clustering quality due to its insensitivity to noisy terms that could lead to incorrect similarity measure. Phrases are less sensitive to noise when it comes to calculating document similarity; this is due to the fact that the probability of finding matching phrases in nonrelated documents is low.

It is clear that the use of phrase in an IR system could have a beneficial effect in terms of efficiency as the inclusion of phrases into the indexing and retrieval procedures [12]. It can also be used for scoring the similarity between two documents according to the matching phrases and their significance.

## 2.4 Genetic algorithm Based Clustering Approach

Genetic algorithms are a computing paradigm inspired by Darwin's theory of evolution. A population of possible solution to a problem is initially created at random. Pairs of individuals are used to combine using cross over operation to produce offspring for the next generation. A mutation process is also used to randomly modify the genetic structure of an individual to produce another individual of the next generation.

Genetic algorithms do not follow an exploration oriented approach; instead they rely on the principle of survival of the fittest. A fitness function plays a central role to identify 'fit' individual from the population towards producing the solution to the problem at hand. Genetic algorithms are applied for a variety of applications, such as the discovery of patterns in text, Clustering, Information retrieval (IR) [10, 23].

A **genetic algorithm-based clustering technique** is called GA-clustering. The searching capability of genetic algorithms can be exploited in order to search for appropriate cluster centers in the feature space such that a similarity metric of the resulting clusters is optimized. The chromosomes, which are represented as strings of real numbers, encode the centers of a fixed number of clusters.

Genetic Algorithm (GA) has proved to be a very powerful mechanism in finding good solutions to difficult problems [21]. The flexibility associated with GA is one important aspect. With the same genome representation and just by changing the fitness function one can have a different algorithm. Fitness functions play important role in document clustering as exploratory phase.

For document clustering GA may minimize the square error of the cluster dispersion [21]:

$$E = \sum_{k=1}^k \sum_{x \in C_k} \|x - m_k\|^2$$

K being the number of clusters,  $m_k$  the centre of cluster  $C_k$ , which makes it similar to the k-means algorithm.

The genome of each gene represents a data point and defines cluster that it also a part of this process. In this scheme all necessary evolution operators are

implemented. The major problem in this representation scheme is that it is not scalable. It seems to be computationally efficient when the number of data points is not too large.

## 2.5 Neural Network

Neural network is a connectionist approach based on the modal of artificial perceptron for computing that involves developing mathematical structures with the ability to learn. Neural networks are reported to have the ability to derive meaning from complicated or imprecise data and are able to extract patterns and detect trends that are too complex to be noticed by either humans or other computer techniques [6, 35]. A trained neural network is considered analogous to an "expert" of the information it has been given to analyze. The expert subsequently is used to provide projections given new examples.

Neural networks have broad applicability to real world problems and have already been successfully applied in many domains. Neural networks are investigated to be well suited for prediction or forecasting problems or tasks.

In vector space model neural network can be implemented using three nodes: QUERY, TERM and DOCUMENT. The links are QUERY-TERM links and DOCUMENT-TERM links. A link between query and term indicates presence of term in query. The weight of query term is calculated as  $tf-idf$ . Similarly a term document link with weight  $tf-idf$  is present for each term present in document [6].

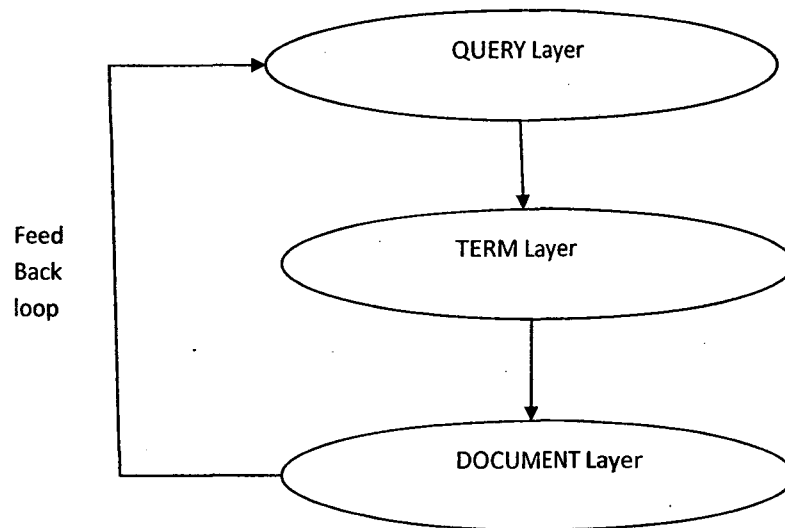
For a feed-forward neural network QUERY node is activated by setting its output to one. Based on this, input weights for corresponding TERM nodes are obtained. In the next phase TERM nodes are activated and all nodes that connect the TERM node to a DOCUMENT node are activated.

The document node contains the sum of all the weight associated with each term in the document.

For a collection of  $t$  TERMS, the DOCUMENT node associated with document  $j$  will now have the value:

$$DOC_j = \sum_{i=1}^t (tf_{ij})(idf_j)$$

The DOCUMENT node now has a weight associated with it that measures the relevance of document to given query. It can easily be seen that this weight is equivalent to simple dot product similarity coefficient.



**Fig. 2.5:** Neural network with Feedback.

## 2.6 Cluster Parameter

For a large document collection, a clustering algorithm may create hundreds of clusters in an initial pass. Although the burden on the user has reduced, it is still tedious to scan hundreds of clusters [20]. We can specify controlling parameters for the clustering algorithm such as,

- A minimum and maximum size of clusters
- A matching threshold value for including documents in a cluster
- The degree of overlap between clusters
- A maximum number of clusters

The maximum and minimum size of clusters forces the clustering algorithm to build the cluster of a manageable size. Clusters that are too large may have a multi-topic

theme. Without a limit on the maximum size of a cluster, a single large cluster tends to attract many documents leading to a diffuse cluster. A minimum cluster size makes it worthwhile to examine member of document of a cluster.

There is a tradeoff between the number of cluster and sizes of clusters. The number of clusters increases or decreases according the size of cluster.

The matching threshold parameter can be tweaked to generate more or fewer clusters. It is the minimum degree of similarity between a member and a cluster representative [20]. If the threshold high, then fewer documents will be eligible to join a cluster and a number of cluster increases. When the threshold is low, clusters become large and fewer in number.

The maximum number of clusters to which a cluster can be assigned is proportional to the degree of cluster overlap. If the cluster overlap is high, cluster may not be clearly distinctive, but a low degree of overlap will ensure greater separation between clusters.

## **2.7 Cluster Based Search**

The problem of synonymy is diminished in a search engine with a clustered file organization. In an inverted file organization, query keywords must exactly match word occurrences in text. The clustered file organization matches a keyword against asset of cluster representatives. Each cluster representative consists of the popular words used in documents related to a common topic. The end user can provide any one of the popular words to retrieve the cluster [20].

In a flat clustering organization, we compare a query against the centroids of the clusters. A centroid is the average representative of a group of documents built from the composite text of all member documents. Cluster based keyword search have a higher precision then search with an invert of file organization. The key word based search often fails because the right keyword was not provided in the query. Matches are exact, and queries with correct keyword and operators give high-precision result. A cluster search is more tolerant and relies on matching a large number of centroid keywords against a few query keywords. The likelihood of an accurate match is higher in the cluster search.

Let consider a single keyword query –“Jaguars”- submitted to a web search engine [20]. There are at least three (automobile .a football team and animals) for the set of hits returned by a search engine. Presenting the list of hits ordered by cluster makes it easy for the reader to quickly spot a group of relevant documents. An ambiguous query like “Jaguars” will include non relevant documents in the initial list, but a relevant cluster will reveal a set of documents that are likely to be relevant to the query.

## 2.8 Applications of Clustering Approaches

Clustering has been gone through because of its wide applicability. The application fields like,

- Optical character recognition,
- Speech recognition,
- Web mining, information retrieval,
- Search engines, and topological analysis
- Clustering has been investigated as process for improving document search or organization, information retrieval and automatic key extraction. It means the similar documents will tends to be relevant to the same queries, and automatically display the group of such documents can improve recall by effectively broadening a search query
- Encoding/decoding as example applications of k-means.

However, a survey of the current literature on the subject the some other practical applications, such as "data detection ...for burst-mode optical receiver [s]", and recognition of musical genres, which are specialized examples of what Alpaydin mentions.

### Proposed Work

---

#### 3.1 Motivation

Document clustering is an important field in text mining and information retrieval. Document clustering concept can help in improving the precision and recall in information retrieval system. In other way we can say that it groups the result of a search engine in to homogeneous groups according to the users query. This leads to reduced search space and improved relevance. The grouping is based on similarity between documents. Similarity score between Pair of documents depends on similarity measure being used. This motivated us to do some experiment on document clustering with different similarity measures to have in insight in to documents clustering process.

#### 3.2 Proposed work

We propose to work on K-means clustering algorithm which is simple yet a reasonably efficient clustering method. In order to observe the effect of various similarity measures on application of K-means algorithm for document cluster we propose to do following experiment.

In our experiment we will select different (say - n) queries for a standard search engine. We will merge the result of all the queries after converting each result (HTML document) to text file. After that we will cluster the result in to n clusters using k-means (here  $k = n$ ). Our idea is that the results of same query should have same similarities to form homogeneous groups. We will experiment with different similarity measure used in VSM. Following steps will be followed in conducting the experiments.

Step 1. Prepare the data set on that we are going to apply the clustering approaches.

- Down load the results for different queries from the search engine.
- Convert the down loaded data (files) from HTML to Text file.
- Perform stemming and remove stop words.
- Merge all text files in to one folder.

Step 2. Form the vector space model for text data obtained in previous step giving a matrix showing weights of each term in each documents. (The weights are assigned on *tf\*idf* measure )

Step 3. Use the matrix obtained in step 2 as the data input for clustering process.

Apply the clustering algorithm (K-means) for that matrix by using the different similarity coefficients (Euclidian, Cosine and Dot product measure).

Step 4. Analysis of the result (Compare the results obtained for different similarity measures)

- Compare on the basis of average intra cluster distance
- Compare the result of different similarity measure coefficient with the similarity measure coefficients and actual result.



**Implementation of the proposed work**

---

**The implementation of K –Means clustering algorithm**

On the basis of proposed work we present here results our experiments. Experiments were performed in Matlab.

**4.1 First Data set**

We merged the results of four queries from yahoo search engine and tried to cluster them using K-means algorithm. Our data set as follows.

Sn. No	Query	Number of documents
1	Computer Architecture	4
2	Data Base	4
3	Programming	4
4	Web Mining	4

**Table 4.1:** Data set of 16 documents.

In the above data set we have four queries, each specifying one cluster thus we have four clusters. Top documents of the result obtained formed data set of each cluster.

The dataset was converted into a matrix of size 16×1539.

We used K-means with three different similarity measures- Euclidian Distance, Cosine and Dot Product to cluster the data set in to four clusters.



The result:

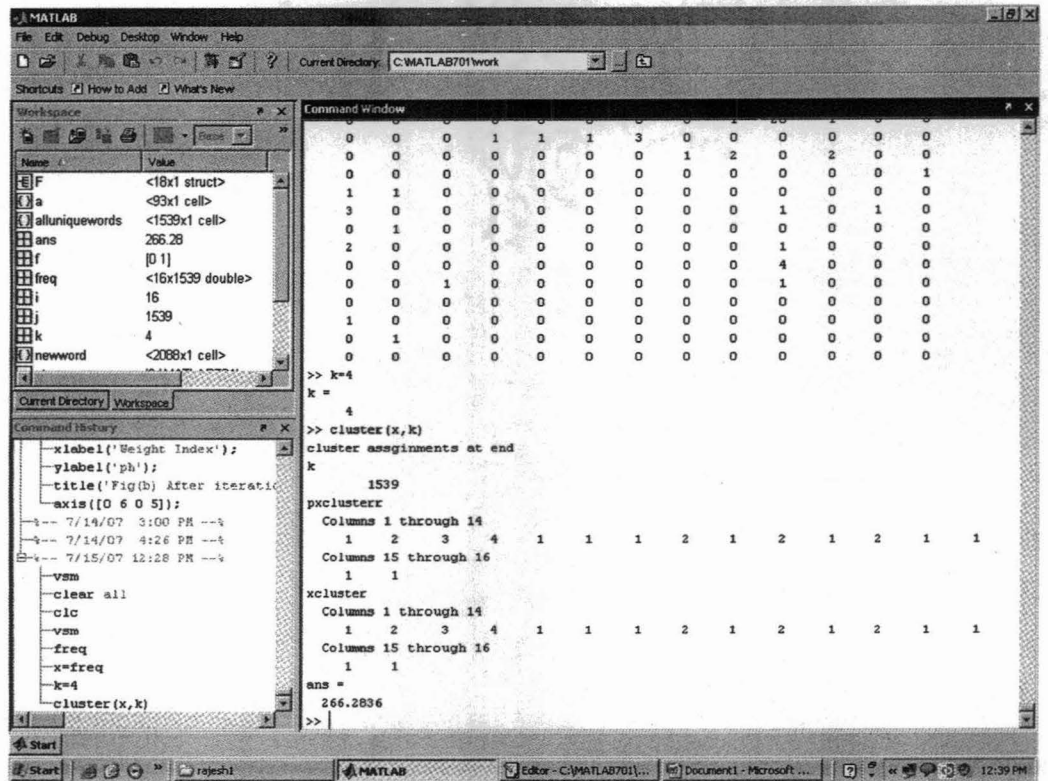


Fig. 4.2: Screen shot from Matlab for the result of K- means clustering.

Columns 1 through 14

1 2 3 4 1 1 1 2 1 2 1 2 1 1

Columns 15 through 16

1 1

ans = 266.2836 -> Average intra cluster distance

Cluster	Number of documents	Number of correctly classified Document
Cluster 1	4	3
Custer2	4	2
Cluster3	4	1
Cluster4	4	1

Table 4.2: Result of data set for Euclidian Distance measure.

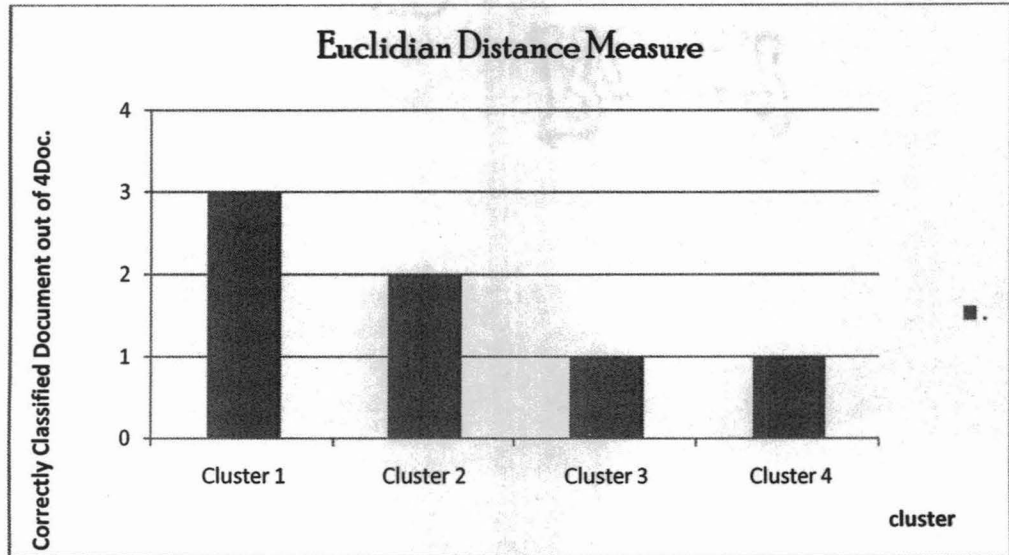


Fig. 4.3: Bar representation of the Result of data set for Euclidian Distance measure.

#### 4.1.2 Cosine Measure Similarity Coefficient

X cluster Columns 1 through 14

1 2 3 4 2 4 4 2 2 2 3 2 4 4

Columns 15 through 16

4 4

ans = 360.8491

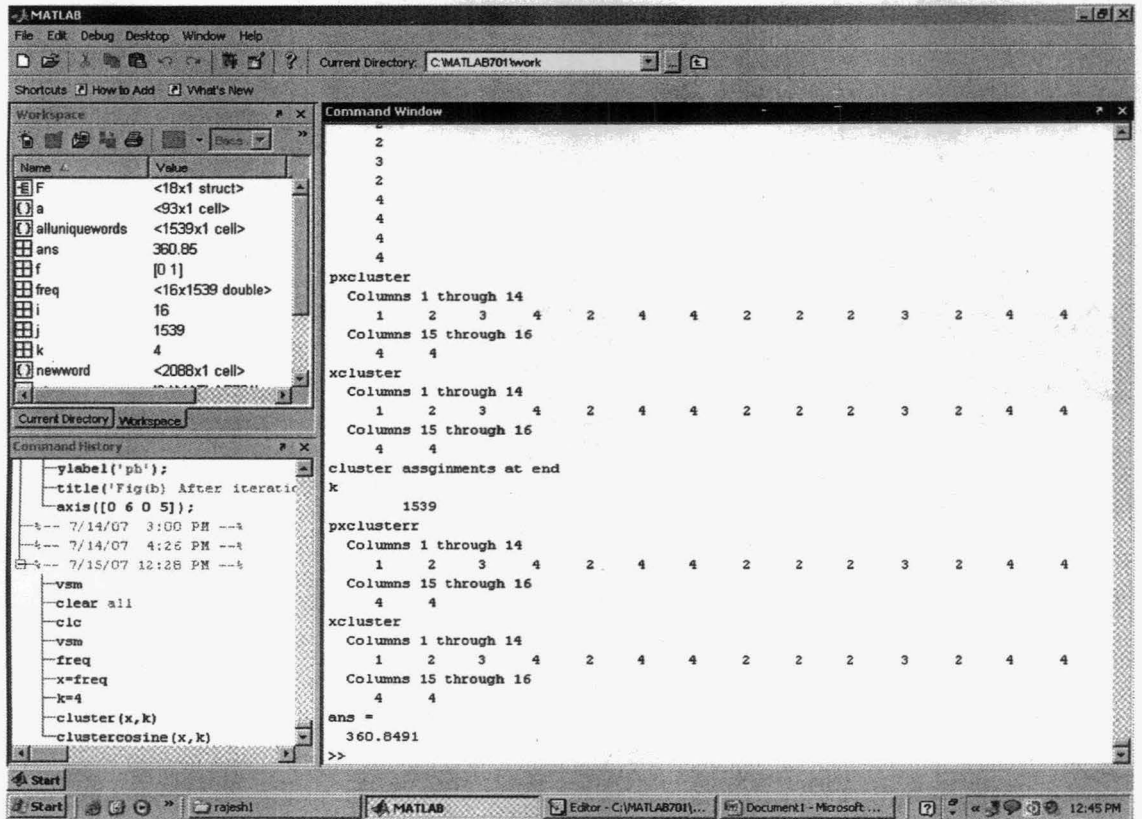
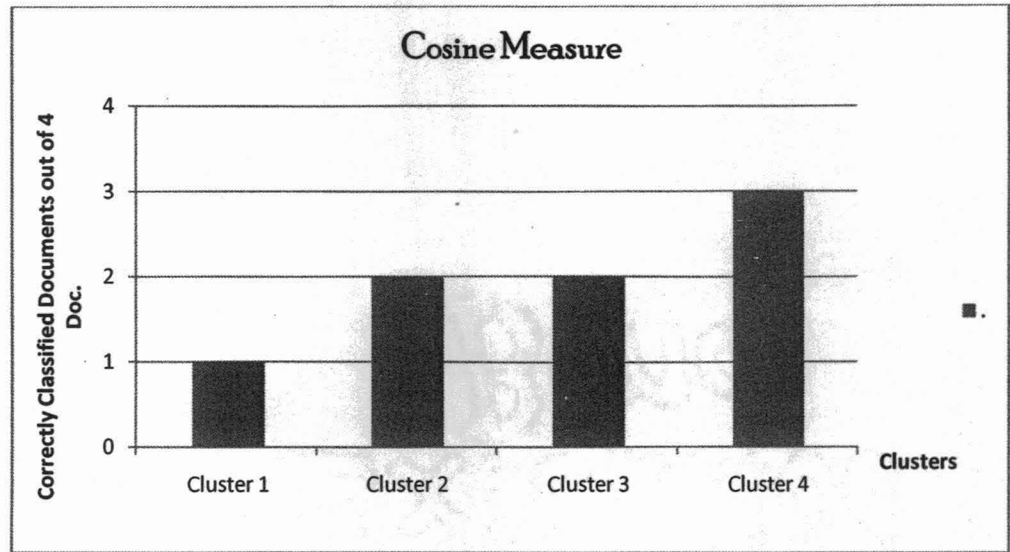


Fig. 4.4: Screen shot from Matlab for the result of K- means clustering for Cosine measure.

Cluster	Number of documents	Number of correctly classified Document
Cluster 1	4	1
Custer 2	4	2
Cluster 3	4	2
Cluster 4	4	3

Table 4.3: Result of Data set for Cosine measure similarity coefficient.



**Fig 4.5:** Bar representation of the Result of data set for Cosine measure SC.

There is document in the cluster 1. It matched the only single document. Two documents in the cluster 2 and cluster3 also have two documents but the cluster4 matched the three documents.

#### 4.1.3 The Dot Product or Inner Product

The result and table of the dot product given blow.

Cluster	Number of documents	Number of correctly classified Document
Cluster 1	4	1
Custer 2	4	3
Cluster 3	4	2
Cluster 4	4	2

**Table 4.4:** Result of Data set for Dot Product similarity coefficient.

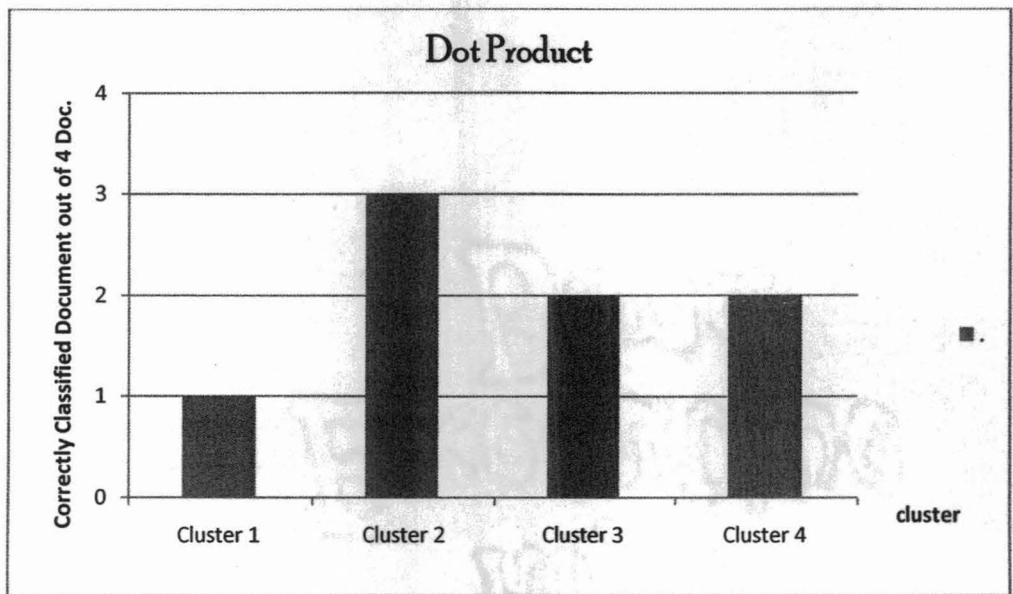


Fig 4.6: Bar representation of the Result of data set for Dot Product SC.

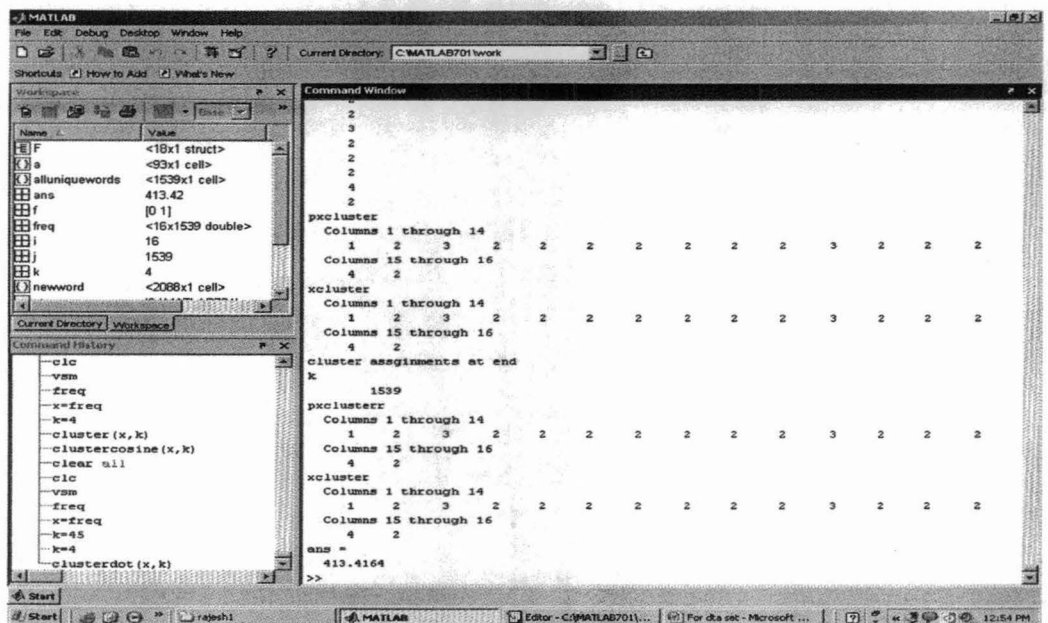


Fig. 4.7: Screen shot from Matlab for the result of K-means clustering for Dot Product SC.

X cluster Columns 1 through 14

1 2 3 4 2 2 2 2 2 2 3 2 2 2

Columns 15 through 16

4 2

ans = 413.4164

## 4.2 Second Data set

We merged the results of six queries from Google search engine and tried to cluster them using the K-means clustering algorithm. Our data set as follows:

Sn. No.	Query	Number of Documents
1	Computer Network	6
2	Artificial Intelligence	5
3	Data Base	5
4	Operating System	14
5	Softer ware	12
6	Web Mining	18

**Table 4.5:** Data set of 60 documents.

In the above data set we have six queries, each specifying one cluster thus we have six clusters. Top documents of the result obtained formed data set of each cluster.

We used K- means with three different similarity measures- Euclidian Distance, Cosine and Dot product to cluster the data set in to six clusters.

### 4.2.1 Euclidian Distance Measure Coefficient

The data set has sixty documents. There are six kinds of documents each kind has different number of documents. The results as follows:

The % of correctly classified document =  $\frac{\text{No. of correctly Classified Document}}{\text{Total no.of Document in cluster}} \times 100$



Cluster	Number of Documents	Number of Correctly Classified Documents	% of Correctly Classified Document
Cluster 1	6	3	50
Cluster 2	5	1	20
Cluster 3	5	4	80
Cluster 4	14	4	28.5
Cluster 5	12	2	16.66
Cluster 6	18	2	11.11

Table 4.6: Results of Data set for Euclidian Distance measure.

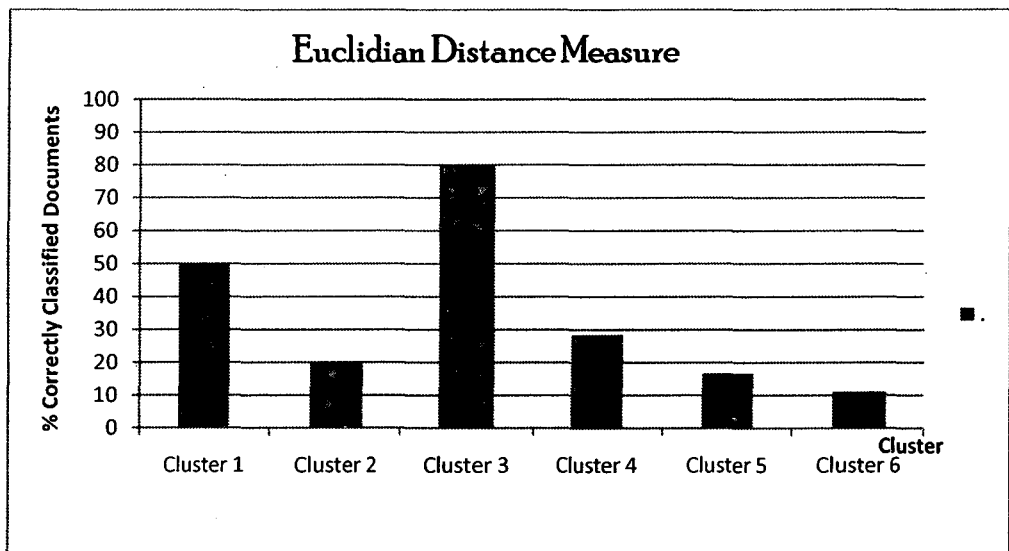


Fig. 4.8: Bar Graph representation of the Result of data set for Euclidian distance measure.

xcluster

Columns 1 through 14

5 1 3 4 3 6 1 4 4 5 5 3 3 3

Columns 15 through 28

1 5 3 1 4 3 1 5 5 3 1 3 6 3

Columns 29 through 42

3 2 4 3 3 1 5 1 1 3 4 1 3 4

Columns 43 through 56

4 5 2 1 3 4 4 3 3 2 1 1 3 3

Columns 57 through 60

1 3 3 3

ans = 1.6114e+003

#### 4.2.2 Cosine Measure

Cluster	Number of Documents	Number of Correctly Classified Documents	Percentage Correctly Classified Document
Cluster 1	6	3	50
Cluster 2	5	1	20
Cluster 3	5	1	20
Cluster 4	14	5	35.71
Cluster 5	12	2	16.66
Cluster 6	18	2	11.11

Table 4.7: Results of Data set for Cosine measure.

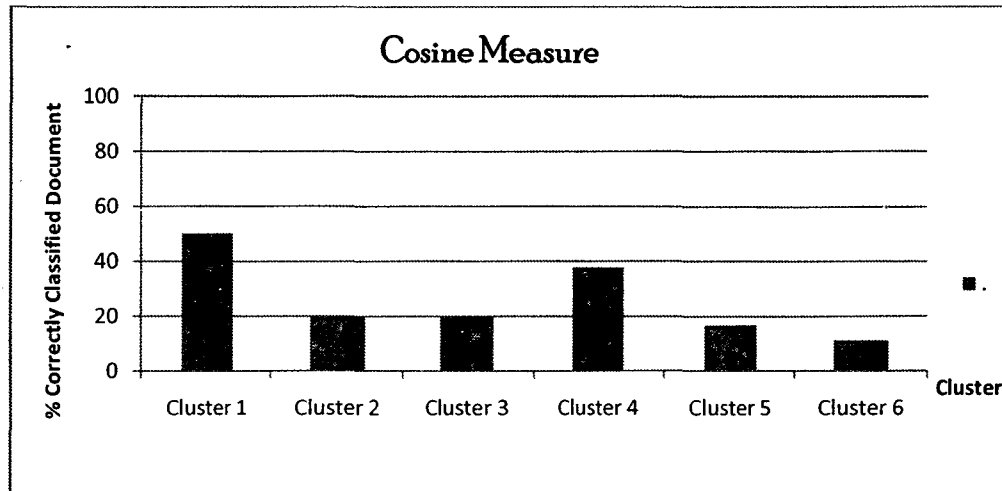


Fig. 4.9: Bar Graph representation of the Result of data set for Cosine measure.

Xcluster

Columns 1 through 14

1 2 3 4 5 6 3 4 1 2 2 2 1 2

Columns 15 through 28

2 2 4 1 4 4 1 2 6 1 2 4 6 1

Columns 29 through 42

4 1 4 1 4 2 2 2 2 4 4 1 2 4

Columns 43 through 56

4 2 1 2 5 4 4 2 6 2 1 1 2 4

Columns 57 through 60

2 5 1 5

Ans = 1.9978e+003

#### 4.2.3 Dot or Inner Product

xcluster

Columns 1 through 14

2 2 3 2 5 6 2 2 2 2 2 2 1 2

Columns 15 through 28

2 2 2 2 2 4 2 2 2 2 2 2 6 2

Columns 29 through 42

2 2 2 2 2 2 2 2 2 4 4 2 2 2

Columns 43 through 56

2 2 2 2 2 4 2 2 6 2 2 2 2 2

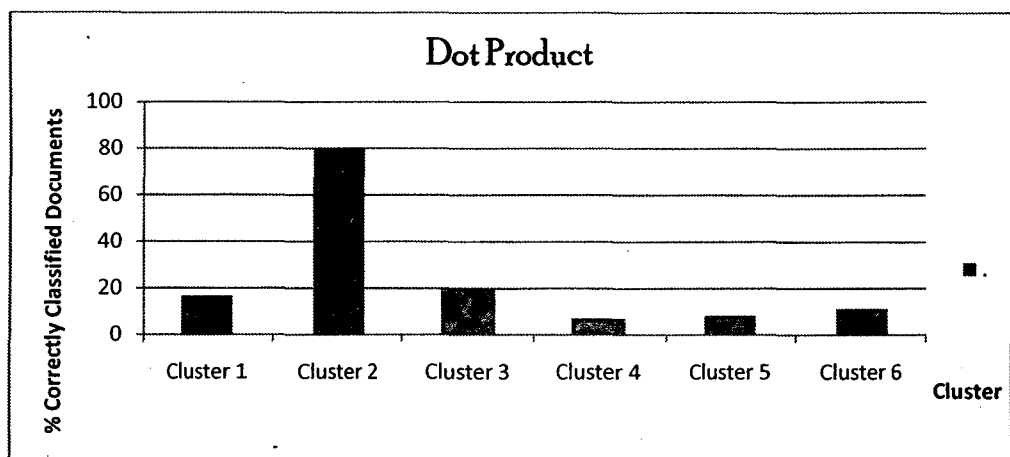
Columns 57 through 60

2 2 1 2

ans = 2.0515e+003

Cluster	Number of Documents	Number of Correctly Classified Documents	% Correctly Classified Document
Cluster 1	6	1	16.66
Cluster 2	5	4	80
Cluster 3	5	1	20
Cluster 4	14	1	7.15
Cluster 5	12	1	8.33
Cluster 6	18	2	11.11

**Table 4.8:** Results of Data set for Dot Product measure.



**Fig. 4.10:** Bar Graph representation of the Result of data set for Dot Product Measure.

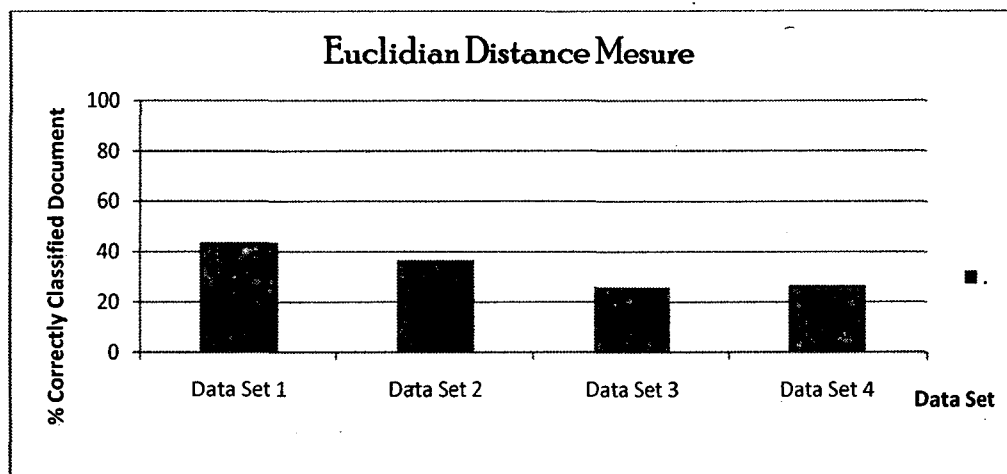
#### 4.3 Final Results on The Basis of Four Data Sets

The comparison of different similarity measure coefficients is done on the basis of the above results. The following the graph and table clearly explain all these.

### 4.3.1 Euclidian Distance Measure

Data Sets	Number of Documents in The Data Sets	Number of Correctly Classified Documents	% of Correctly Classified Document
Data set 1	16	7	43.75
Data Set 2	30	11	36.66
Data Set 3	50	13	26
Data Set 4	60	16	26.66

**Table 4.9:** The Results of Different Data sets for Euclidian Distance Measure.



**Fig. 4.11:** Bar Graph representation of the Results of different data sets for Euclidian Distance Measure.

### 4.3.2 Cosine Measure

Data Sets	Number of Documents in The Data Sets	Number of Correctly Classified Documents	% Correctly Classified Documents
Data set 1	16	8	50
Data Set 2	30	8	26.66
Data Set 3	50	13	26
Data Set 4	60	14	23.33

**Table 4.10:** The Results of Different Data sets for Cosine measure.

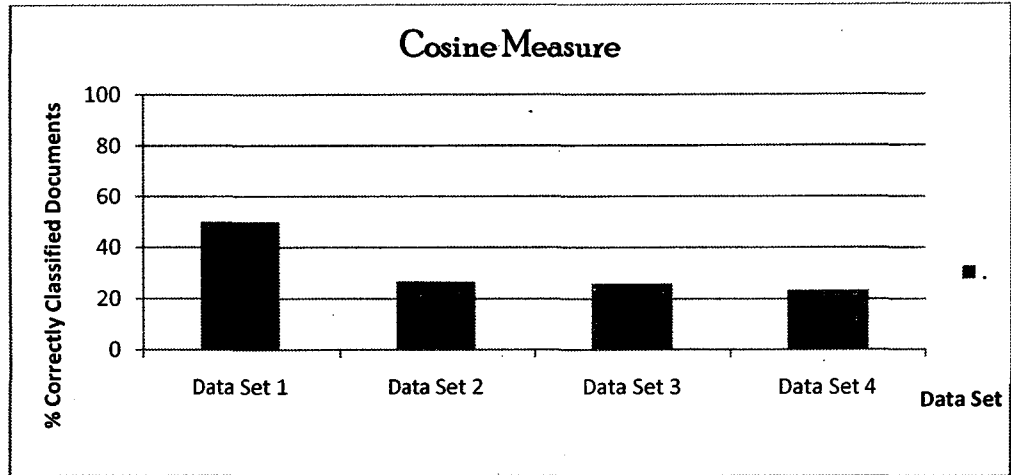


Fig. 4.12: Graph representation of the Results of different data sets for Cosine Measure.

#### 4.3.3 Dot Product

Data Sets	Number of Documents in The Data Sets	Number of Correctly Classified Documents	% Correctly Classified Document
Data set 1	16	8	50
Data Set 2	30	9	30
Data Set 3	50	16	32
Data Set 4	60	10	16.66

Table 4.11: The Results of Different Data sets for Dot Product.

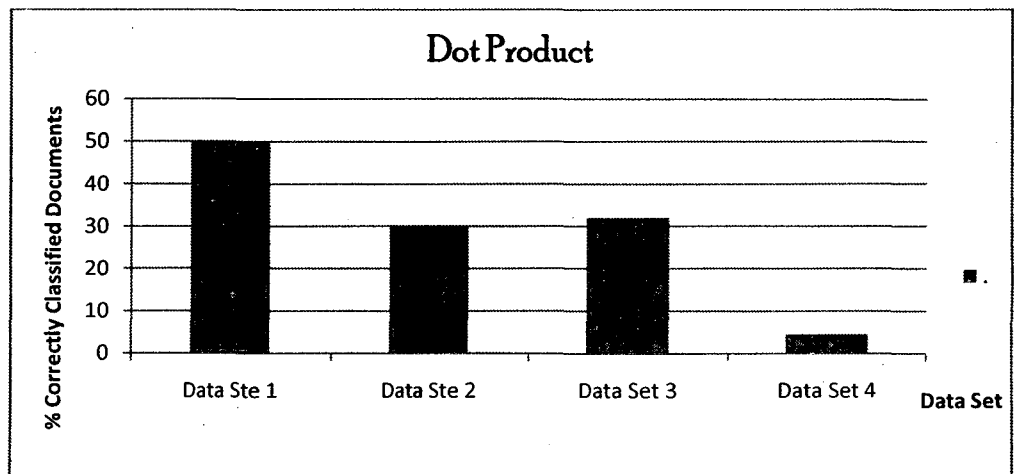


Fig. 4.13: Bar Graph representation of the Results of different data sets for Dot product.

### Conclusion

---

Document clustering is an emerging area of study and research in the field of text mining specifically for returning relevant information from the web. In addition to traditional clustering methods document clustering uses techniques from multidisciplinary fields such as Information retrieval, Natural language processing (NLP), Machine learning etc. in order to gain insight, understand and interpret document clustering process.

Text documents can not be clustered as such instead they have to be represented using some model. Based on this model similarity measure and hence clustering method is selected. The purpose of this dissertation is to address the issue of document clustering using Vector space model (VSM).

In this work we first have explained representation of documents in vector space model. Then we have tried to identify analyze and compare various similarity measures that have been used to find similarity between documents. We have made a survey of various clustering algorithm that have been used for clustering documents. Further we have performed some experiments based on our study to see empirically effect of various similarity measures on document clustering.

Due to limitation of time we have done only few experiments. The work can be extended by taking the results from different search engines. We have experiments only with K-Means algorithm and three similarity measures- Dot Product, Cosine Measure, Euclidian Distance Measure. Experiment can be conducted by including more similarity measures for the experiment can be performed by considering different clustering algorithms and a comparative study of the results can be done.

## References

---

- [1]. A. El-Hamdouchi and P. Willet, *Comparison of Hierarchic Agglomerative Clustering Methods for Document Retrieval*, The Computer Journal, Vol. 32, No. 3, 1989.
- [2]. A. Hinneburg, and D. A. Keim, *An Efficient Approach to Clustering in Large Multimedia Databases with Noise*, in Proceeding of International Conference on knowledge Discovery and Data Mining (KDD98), pp. 58-65, August 1998.
- [3]. A.K. Jain, M.N. Murty and P.J. Flynn, *Data Clustering: A Review*, ACM Computing Surveys. 31(3): 264-323, Sept 1999.
- [4]. Bjerne Larsen and Chinatsu Aone, *Fast and Effective Text Mining Using Linear-time Document Clustering*, KDD-99, San Diego, California, 1999.
- [5]. B.C.M. Fung, K. Wang, M. Ester, *Hierarchical Document Clustering Using Frequent Itemsets*, SIAM international conference on Data mining, SDM 03, San Francisco Canada , USA 59-70, May 2003.
- [6]. David A. Grossman, Ophir Frieder. *Information Retrieval, Algorithm and heuristics*, Landon: Kul;war Acedamic Publishers, 1998.
- [7]. D. R. Cutting, D. R. Karger, J. O. Pedersen, and J. W. Tukey, *Scatter/gather: A cluster-based approach to browsing large document collections*, In Proceedings of the Fifteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 318-329, 1992.
- [8]. E. Ukkonen. *On-line construction of suffix trees*, Algorithmica, 14:249-60, 1995.
- [9]. F. Murtagh, *A Survey of Recent Advances in Hierarchical Clustering Algorithms*,



The Computer Journal, 26(4): 354-359, 1983.

- [10]. Gordon, M.D, *User-based document clustering by redescribing subject descriptions with a genetic algorithm*, Journal of the American Society for Information Science, 2, 1991, pp: 311-322.
- [11]. G. Zhaug, B. Kleyner and M. Hsu, *A local search approach to k- clustering*, Tech report HTPL-1999-119,1999.
- [12]. Hammouda , K.M. Kamel, M.S, *Efficient phrase-based document indexing for Web document clustering*, IEEE trisections on Knowledge and data engineering,vol 16, no. 10, page 1279-1296, October 2004.
- [13]. Illhoi Yoo and Xiaohua Hu, *Clustering Ontology- enriched Graph Representation for Biomedical Documents based on Scale –Free Network Theory*, NFS career grant, Drexel University Philadelphia, USA.
- [14]. J. Kogan and Tebouller M, *A unified framework for clustering data with entropy-like k-means algorithms: theory and applications in preparation*.
- [15]. J. Han and M. Kamber, *Data Mining: Concepts and Techniques*, San Francisco: Morgan Kaufmann, August 2000.
- [16]. J. Koga, M.Teboulle and V.Volkovich, *Text mining with hybrid clustering schemes*, In M.W. Berry and W.M. Pottenger, editors, Proceeding of the Workshop on Text mining (held in conjunction with the third SIAM international conference on Data mining), pages5-16, 2003
- [17]. Ketil Madel, E.Coward and Inge Jonassen, *Fast sequence clustering using suffix arry Algorithm*, Proceeding of Bioinformaics 19<sup>th</sup> IEEE symposium on bioinformatics, page 1221-1226 (10) Jan 2003.
- [18]. Khaled M. Hammouda, M.S. Kamel, *Efficient Phrase based Document Indexing for Web Document Clustering*, IEEE transactions on knowledge and data

engineering, Vol 16,no.10 , Oct2004.

- [19]. L. Kaufman and P. J. Rousseeuw, *Finding Groups in Data: an Introduction to Cluster Analysis*, John Wiley and Sons, 1990.
- [20]. Manu Konchady, *Text Mining application Programming*, Boston, Massachusetts: Charles River Media 2006.
- [21]. Marco Paihno, Fernando Bacao, *Using Genetic algorithms in Clustering Problem*, Geo computation conference 2000
- [22]. M. Steinbach, G. Karypis, and V. Kumar, *A comparison of document clustering Techniques*, KDD Workshop on Text Mining'00, 2000.
- [23]. Mitchell, M. *An Introduction to Genetic Algorithms*, MIT Press, 1996.
- [24]. O. Zamir, O. Etzioni, O. Madani and R. M. Karp, *Fast and intuitive clustering of Web Documents*, In Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining, pages 287-290, 1997
- [25]. O. Zamir, O. Etzioni, *Web document clustering: a feasibility demonstration*, in Proceedings of 19<sup>th</sup> international ACM SIGIR conference on research and development in information retrieval (SIGIR 98), 1998, pp 46-54.
- [26]. Paul Bradley and Usama Fayyad, *Refining Initial Points for K-Means Clustering*, Proceedings of the Fifteenth International Conference on Machine Learning ICML98, Pages 91-99. Morgan Kaufmann, San Francisco, 1998.
- [27]. P. Willett, *Recent trends in hierarchical document clustering: a critical review*, Information processing and management, 24: 577-97, 1988.
- [28]. P. Bradley, U. M. Fayyad and C. Reina, *Scaling clustering algorithms to large Databases*, in Proceedings of 4<sup>th</sup> International Conference on Knowledge Discovery and Data Mining (California), pp. 9-15, AAAI 1998.
- [29]. R.B. Yates, B. R.-neto, *Modern Information Retrieval*, Pearson Education 2004.

- [30]. R. C. Dubes and A. K. Jain. *Algorithms for Clustering Data*. Prentice Hall College Div, Englewood Cliffs, NJ, March 1998.
- [31]. R. T. Ng and J. Han, *Efficient and effective clustering methods for spatial data mining*, in Proceedings of the Twentieth International Conference on Very Large Databases (VLDB'94) (Santiago, Chile), pp. 144-155, Chile, September 1994
- [32]. S. Wermter, Chihli Hung, *Neural Network-based Document Clustering Using Wordnet Ontologies*, conference in bio-informatics and pateran matching, 2005.
- [33]. S.J. Puglisi, W.F Smyth and Andrew Turpin, *Suffix Arrays: What Are They Good For ?* 17<sup>th</sup> Australasian Database Conference ACSC Hobart, Australia. Conferences in Research and Practice in Information 2006.
- [34]. Soumen Chakerabarti, *Mining The web Discovering Knowledge From Hypertext Data*, Morgan Kaufmann Publishers, San Francisco 2003.
- [35]. Sushmita Mitra, T.Acharya, *Data Mining ,Multimedia,Soft Clustering and Bioinfoemetics* John willy &sons, inc., publication. 2004
- [36].U. Manber and G. Myers, *Suffix Arrays: A New Method for On- Line String Searches*, SIAM J. Computing, vol. 22, no. 5, pp. 935- 948, 1993.
- [37]. X. Cui, T. E. Potok, and P. Palathingal, *Document Clustering using Particle Swarm Optimization*, In Proceedings of the 2005 IEEE Swarm Intelligence Symposium, June, 2005, Pasadena, California, USA
- [38]. Zhao Y. and Karypis G., 2004, *Empirical and Theoretical Comparisons of Selected Criterion Functions for Document Clustering*, Machine Learning, 55 (3): pp. 311-331.