Document clustering based on keyword frequency and concept matching technique in Hadoop

R.Priyadarshini, Latha Tamilselvan

Abstract—. The term big data has come into use in recent years. It is used to refer to the ever-increasing amount of information that organisations are storing, processing and analysing. Owing to the growing number of information sources big data based file systems are necessary. Due to the explosion growth of digital information, automatic document clustering or categorization has become more important. Document management and clustering is more important for content management systems. It is a necessary to mine and store web documents in CMS. In this study online document based content management with automatic URL indexing is dealt with. It's highly possible that the content in CMS will be redundant over the web as most of the time the content will be gathered from already existing websites. Back tracking the source of such content will become obsolete and also changes to the source are difficult to be tracked. So to solve this problem document based content management (DCMS) is very much essential. Stored documents in DCMS comprises of huge amount of data so there is need for document clustering. Previous works on clustering documents have no consideration for the semantic information as they consider only the structural information. In this study, a novel semantic and similarity measure based technique is proposed that concurrently considers both structural and semantic information of document. Semantic analysis based clustering is applied to the text documents and then similarity measure is devised among the documents based on machine learning algorithms using Apache hadoop. In order to achieve accurate clustering and efficient retrieval, initially the documents are stored in hadoop distributed file system and they are clustered using K-means algorithm.Then the clustering is also done using concept matching technique and time for formation of clusters were plotted and compared.

Index Terms— Concept Matching, Semantic document clustering, Document Clustering, Hadoop, Document based content management system.

1 Introduction

Due to the enormous growth of digital information, automatic document clustering or categorization has become more important. Document management is most essential in the content management systems. There are numerous types of web based Content Management Systems available. This kind of CMS compensates the explosion of web content in blogs, social networks etc. The problem in web is that it is very difficult to find relevant information in World Wide Web. To address this problem "semantic web" has been proposed as the extension of current web [11]. In the proposed study semantic similarity technique is used in document management which will aid in quick and easy document retrieval. Clustering is a key concept in document mining. A clustering process aims to analyze the similarities between data objects and build groups of them. The grouped objects can then be used to browse easily through a very large list of data sets. Document clustering aims to automatically segregate documents into groups based on similarities of their contents.

Each group consists of documents that are similar between themselves and dissimilar to documents of other groups. Those documents are categorized as document cluster having high inter-cluster similarity and low inter cluster similarity respectively [8]. This task is accomplished by unsupervised learning. Unsupervised learning is one of the machine learning technique which is a type of artificial intelligence that provides computers with the ability to learn without being explicitly programmed. This technique focuses on the development of programs that can teach themshelves to grow and chamge when exposed to new data. It concerns the development of systems that can learn form data. In traditional document clustering methods, Vector VSM is used, this technique suffers dimensionality problem [13]. In proposed system clustering is done by frequency of occurrence of keyword using K-means clustering algorithm. Further clustering is improved by semantic relatedness and concept matching techniques.

2 Document clustering techniques

2.1 Basics of Clustering

Clustering is a key concept in document mining. A clustering process aims to analyze the similarities between data objects and build groups of them. The grouped objects can then be used to navigate easily through a very large list of data sets.

2.2 Document Clustering

Document clustering aims to automatically divide documents in to groups based on similarities of their contents. Each groups consist of documents that are similar between themshelves, tha have high intra-cluster similarity and dissimilar to documents of other groups that have low inter-cluster similarity. Clustering documents can be considered as an unsupervised task that attempts to classify docu-

- *R.Priyadarshini is a Research Scholar in SCIMS in B.S.Abdur Rahman University, Chennai, India, 600048. E-mail:rpriyadarshini@bsauniv.ac.in*
- *Latha Tamilselvan is Prof. & Head in Department of IT , B.S.Abdur Rahman University, Chennai , 600048.E-ail:latha.tamilselvan94@gmail.l.com.*
- *A.Mohammed Thoufic Rahman is pursuing P.G. in B.S.Abdur Rahman University, Chennai, 600048. E-mail:md.thoufi@gmail.com.*

ments by discovering underlying patterns, i.e., the learning process is unsupervised, which means that no need to define the correct output (i.e., the actual cluster into which the input should be mapped to) for an input.

2.3 Machine Learning Algorithms

Machine learning is a type of artificial intelligence that provides computers with the ability to learn without being explicitly programmed. Machine learning focuses on the development of computer programs that can teach themselves to grow and change when exposed to new data. It concerns the construction and study of systems that can learn from data. For example, a machine learning system could be trained on email messages to learn to distinguish between spam and non-spam messages. After learning, it can then be used to classify new email messages into spam and non-spam folders.

2.4 Concept Matching Technique

The entities are recogonized in the documents and the relvancy with the concept will be calculated based on the threshold value represented in the Alchemy API. Concept matching will be done with help of AlchemyAPI. A relevance score is calculated for each text documents, and the results are returned sorted by relevancy. Use the relevance score to determine the concept of the documents and documents are clustered based on the concept identified by the API.

3 Previous Related Work

Xiping Liu, Changxuan Wan, and Lei Chen [1] Clustering XML documents by matching the given xml query has been implemented. There are two main approaches in this paper: Conventional approach - It clusters results after search results are retrieved. Clusters search results actively, which has characteristics of clustering on the fly. The generated clusters are organized into a cluster hierarchy with different granularities to enable users to locate the results of interest easily. Experimental results demonstrate the meaningfulness of the proposed semantics as well as the efficiency of the proposed methods.Rui Máximo Esteves , Chunming Rong [2] , This paper compares k-means and fuzzy c-means for clustering a noisy realistic and big dataset. In a huge dataset, the execution times of both algorithms have high variances according to the initial seeding. Generally, k-means is slower than the fuzzy version. They made the comparison using a free cloud computing solution Apache Mahout/Hadoop and Wikipedia's latest articles. They found that in a noisy dataset, fuzzy c-means can lead to worse cluster quality than k-means. The convergence speed of k-means is not always faster.

Chien-Liang Liua,∗, Tao-HsingChangb, Hsuan-HsunLic [3] This paper focuses on semi-supervised clustering and proposes a novel algorithm called fuzzy semi-Kmeans to perform document clustering with a small amount of labeled documents. *K-means clustering* model and uses the seeds to bias clustering toward a good region of the search space. Fuzzy-semi-Kmeans provides the flexibility to employ different fuzzy membership function to measure the distance between data. This work conducts experiments on three datasets and compares fuzzy semi-Kmeans with several methods. Results are efficient. The experimental results indicate that fuzzy semi-Kmeans can generally outperform the other methods.
Khaled B. Shaban [4], In this paper, semantic understanding based approach to cluster documents is presented. The approach is based on semantic notions to represent text, and to measure similarity between text documents. The representation scheme reflects existing relations between concepts and facilitates accurate similarity measurements that result in better mining performance. They tested the system against different standard clustering techniques and different data sets. The semantic approach has enabled more effective document clustering than what conventional techniques would provide.
Jeong Hee Hwang a, Keun Ho Ryu b [5], Proposed a tree decomposition method for efficiently clustering XML documents. It clusters XML documents by a global criterion function, by considering the weight of common structures. It initially extracts representative structures of frequent patterns from XML documents using a sequential pattern mining algorithm. The experimental results compare to previous work show the effectiveness of this approach.

From the above findings of existing work, it is clear that semantic analysis of any document or system yields effective results. There is no clustering technique based on semantic analysis and similarity measures. In our proposed work clustering will be done initially by keyword occurrence and customizing K-Means algorithm. Further it is improvised by semantic clustering in hadoop file system using machine learning algorithms.

4 System Architecture

The proposed framework is classified in to two main phases. Figure 1 shows the first phase of the proposed system. The framework consists of five main modules such as 1.Creation of user interface, 2.Collection and storage of documents, 3.Clustering of documents using K-Means algorithm, 4.Enhancement of clustering algorithm based on occurrence of keyword, 5.Improved clustering by similarity measures 6. Retrieval of source documents accurately.

Figure 1 depicts that user upload collections of text documents from local system to Hadoop Distributed File System with the help of user interface. Before clustering, the stored documents are converted into sequence files and given to mahout component of hadoop as inputs.

The user interface is used to create virtual documents and it is stored in Hadoop distributed file system as shown in the figure 1. It will cluster the text documents based on K-Means clustering and the algorithm is customized based on keyword occurrence for easy retrieval of documents.

After clustering the documents the results are moved to the NoSQL database for generating graph and searching documents through user interface.

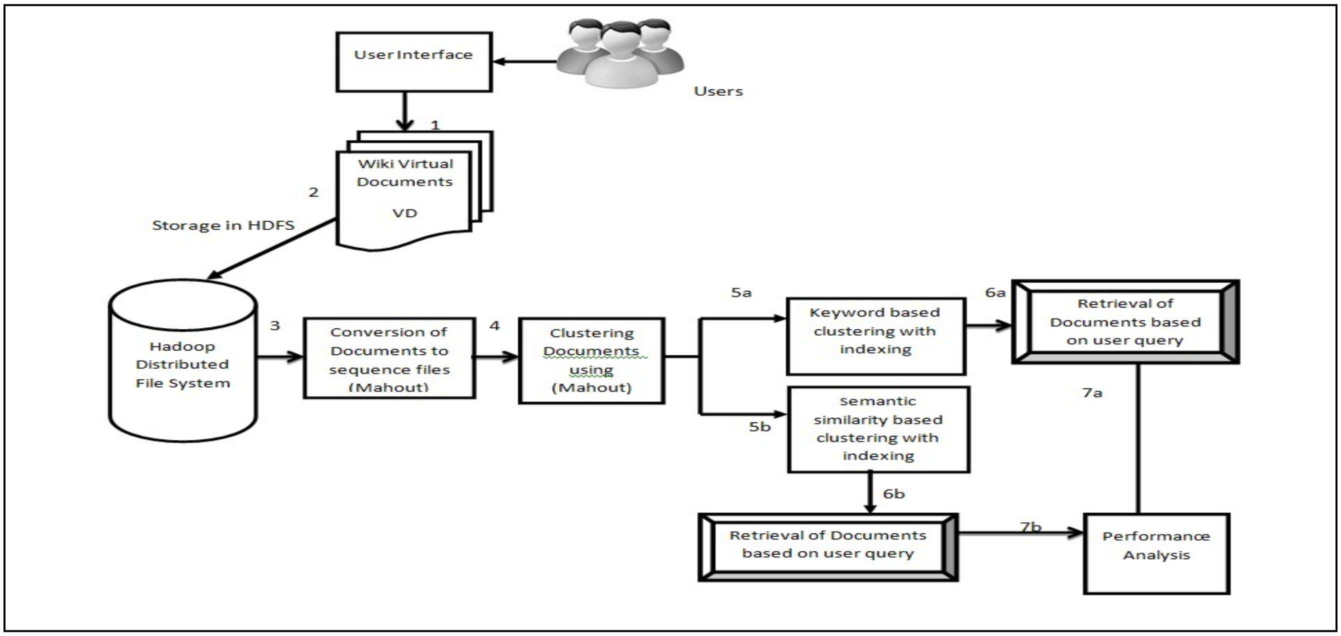

User Interface
Users
1
Wiki Virtual Documents
VD
2
Storage in HDFS
Hadoop Distributed File System
3
Conversion of Documents to sequence files (Mahout)
4
Clustering Documents using (Mahout)
5a
Keyword based clustering with indexing
6a
Retrieval of Documents based on user query
5b
Semantic similarity based clustering with indexing
7a
6b
Retrieval of Documents based on user query
7b
Performance Analysis

Figure 1 : System Architecture

After clustering the documents the results are moved to the NoSQL database for generating graph and searching documents through user interface.

5. Methodology

5.1. Creation of user interface:

The user interface is created for to store documents from local systems to Hadoop Distributed File System (HDFS) and retrieval of documents from clusters by the user query. Functions available in user interface are Login page , Upload files option along with the search option.

5.2. Storing files into HDFS:

Text documents are collected from local system and it is stored in the Hadoop Distributed File System with the help of user interface automatically. By specifying path we can store the documents where ever in the HDFS.

5.3. Clustering of documents based on keyword:

Clustering documents using K-Means clustering algorithm and clustered resultant documents are moved to the database for generating graph and searching documents. The following algorithm illustrates that with the frequency of occurrence of keywords in a document the documents are clustered by means of cluster vectors and the clusters are recomputed again based on the weights assigned to the keywords.

5.4. Keyword based Clustering:

K-Means is a simple but well known algorithm for grouping objects, clustering. Again all objects need to be represented as a set of numerical features. In addition the user has to specify the number of groups (referred to as k) he wishes to identify. Each object can be thought of as being represented by some feature vector in an n dimensional space, n being the number of all features used to describe the objects to cluster.

The algorithm then randomly chooses k points in that vector space, these point serve as the initial centers of the clusters. Afterwards all objects are each assigned to the center they are closest to. Usually the distance measure is chosen by the user and determined by the learning task. After that task is computed, for each cluster a new center is computed by averaging the feature vectors of all objects assigned to it. The process of assigning objects and recomputing centers is repeated until the process converges. The algorithm can be proven to converge after a finite number of iterations.

Several tweaks concerning distance measure, initial center choice and computation of new average centers have been explored, as well as the estimation of the number of clusters k.

5.5. Frequency calculation for keywords:

The frequency of occurrence of the keyword is calculated in each document using word count algorithm, and weight is assigned to the keyword according to the formula.

Weight (t,d) = tf * idf -------------- (1)

tf = No. of terms t in (d) / total no. of terms in (d) ----(i)

idf = log_10(No. documents in Corpus / df (t)) = log_10(N/dft) ----(ii)

The term frequency of term (t) in document (d) is defined as the number of times that (t) occurs in (d). We want to use tf when computing query-document match scores. Raw term frequency is not what we want: A document with 10 occurrences of the term is more relevant than a document with one occurrence of the term. But not 10 times more relevant. Relevance does not increase proportionally with term frequency.

Document frequency is related to number of times a term appeared in a document. Rare terms are more informative than frequent terms, on the contrary to stop words. Consider a query term that is frequent in the collection (e.g., high, increase, line) A document containing such a term is more likely to be relevant than a document that doesn't, but it's not a sure indicator of relevance. We used document frequency (df) to capture this in the score. For frequent terms, we want positive weights for words like high, increase, and line, but lower weights than for rare terms. df (=N) is the number of documents that contain the term.

idf (inverse document frequency) of (t) is use log(N/dft) instead of N/dft to "dampen" the effect of idf. The tf-idf weight of a term is the product of its tf weight and its idf weight. Best known weighting scheme in information retrieval.

Note: the "-" in tf-idf is a hyphen, not a minus sign! tf-idf weighting increases with the number of occurrences within a document and increases with the rarity of the term in the collection.

5.6. Clustering Based on the Concept Matching:

The concept matching technique is implemented based on the Natural Language Processing tool (NLP tool) namely Alchemy API. AlchemyAPI employs sophisticated text analysis techniques to concept tag documents in a manner similar to how humans would identify concepts. The concept tagging API is capable of making high-level abstractions by understanding how concepts relate, and can identify concepts that aren't necessarily directly referenced in the text.

The entities in the document are recognized and the content in the documents are analyzed. The relevancy of the content with that of entity is calculated. Based on the highest relevancy value, the the documents are clustered in the corresponding clusters. The formations of clusters are based on the key entities identified from the documents.The clusters are created by K-Means algorithm based on the occurrence of the keyword.

The concept matching is implemented using the NLP tool which in turn categorizes the documents in tha appropriate clusters. A relevance score is calculated for each concept based on statistical analysis, and the results are returned sorted by relevancy. The relevance score is used to determine the keyword's relative importance.

Table 1
Algorithm for K-Means clustering based on keyword frequency

Assumptions:
Let variable n be number of features (Documents)used to describe objects to clusters.
Let S be the set of feature vectors (|S| is the size of the set)
Let A be the set of associated clusters for each feature vector
Let freq(x,y) be the frequency calculation function
Let c[n] be the vectors for our clusters.
Init :
Set intial cluster in the specified path.
Algorithm randomly chooses the center points in the vector space.K-Points serves as intial centers of the clusters. three images.

```
Let S' = S
  //choose n random vectors to start our clusters
  for i=1 to n
   j = rand(|S'|)
  c[n] = S'[j]
  S' = S' - {c[n]} //remove that vector from S' so we can't choose it again
end
//assign initial clusters
for i=1 to |S|
A[i] = argmax(j = 1 to n) { freqS[i], c[j]) }
End
Run:
Let change = true
while change = false //assume there is no change
//reassign feature vectors to clusters)
  for i = 1 to |S|
  a = argmax(j = 1 to n)  {freq(S[i], c[j]) }
   if a != A[i]
   A[i] = a
    change = true //a vector changed affiliations
             //recompute our cluster vectors and run again
    end
    end
 //recalculate cluster locations if a change occurred
 if change
    for i = 1 to n
     mean, count = 0
         for j = 1 to |S|
            if A[j] == i
             mean = mean + S[j]
             count = count + 1
           end
         end
         c[i] = mean/count
       end
        end
```

6 GRAPHICAL ANALYSIS

6.1 Clustering Based on Keyword Frequency

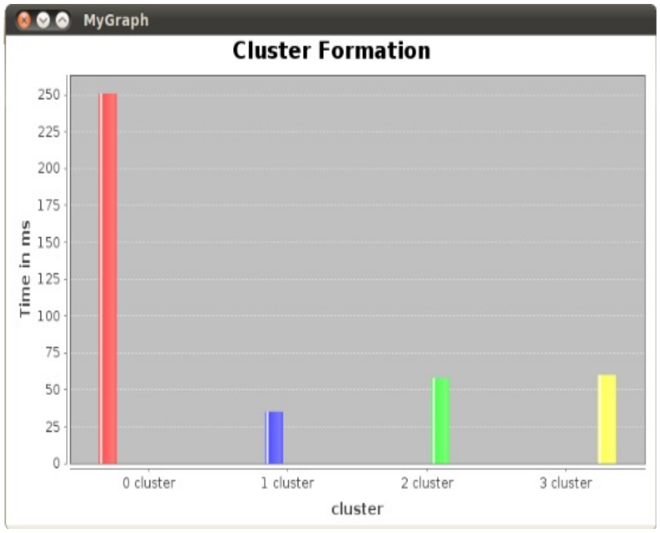


Figure 2: Graph generated for Cluster Formation

K-means algorithm is used to form clusters, for n number of keywords n+1 clusters are formed. The time taken for generating clusters is plotted in the Y-axis and Number of clusters are plotted in X-axis.100 documents are tested and 3 keywords are given in the search query. Figure 2 shows the cluster formation with time.

6.2 Clustering Based on Concept Matching

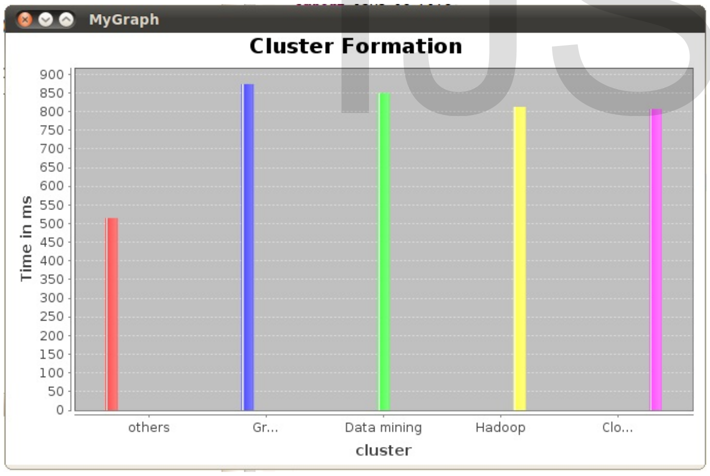


Figure 3: Graph generated for Cluster Formation

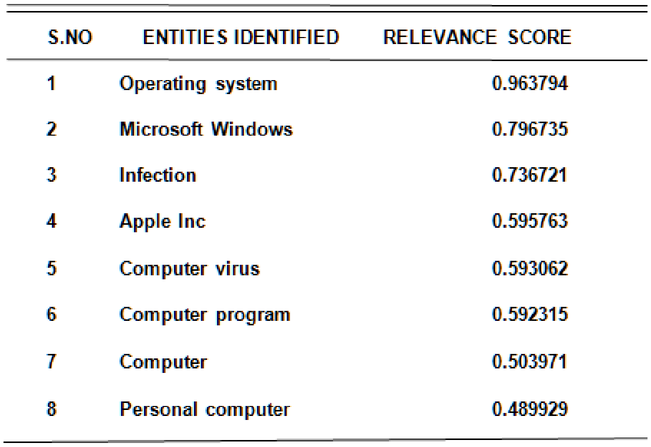
Alchemy API is used to find out the relevancy score for the entities recognized in the concept given in the document. 100 documents were tested and based n keywords the concept of the documents are analysed and stored in the appropriate clusters.Figure 3 represents the graph generated for cluster semantically.

Table 2 shows the relevancy score grnerated by the NLP toolfor the documents uploaded in the Hadoop system.

7. Conclusion

In this paper, Text documents have been uploaded in the Hadoop Distributed File System and the input documents are converted into sequence files and stored in HDFS. Keyword based clustering based on K-Means algorithm is implemented in mahout component of hadoop. The files are retrieved using the user query through user interface and finally the graph is generated to display the clustered documents from MongoDB. Semantic document analysis is performed based on similarity measures and concept matching. NLP tool is used to cluster documents in a meaningful way.The time calculated for the keyword based clustering and semantic clustering is calculated and graphs are plotted. In future the threshold for number of cluster formation will be fixed by testing more number of documents and the same concept will be used for Big Data and performances will be analyzed.

TABLE 2
RELAVANCE SCORE GENERATION



S.NO	ENTITIES IDENTIFIED	RELEVANCE SCORE
1	Operating system	0.963794
2	Microsoft Windows	0.796735
3	Infection	0.736721
4	Apple Inc	0.595763
5	Computer virus	0.593062
6	Computer program	0.592315
7	Computer	0.503971
8	Personal computer	0.489929

The relevance score is calculated based on the key entities in the documents loaded to the system. The score which is highest and soo much accurate will be loaded to the appropriate clusters.

Acknowledgment

The authors wish to thank the HUG- Hadoop User Group Forum who contributed in selecting appropriate tools for the logic formulated in the research.

References

[1] Xiping Liu, Changxuan Wan, and Lei Chen, “Returning Clustered Results for Keyword Search on XML Documents”, IEEE Transactions On Knowledge and Data Engineering, Vol. 23, No. 12, December 2011.

[2] Rui Máximo Esteves , Chunming Rong, “Using Mahout for clustering Wikipedia’s latest articles: A comparison between k-means and fuzzy c-means in the cloud”, 978-0-7695-4622-3/11 $26.00 © 2011 IEEE

[3] Kathleen Ericson∗, Shrideep Pallickara“On the performance of high dimensional data clustering and classification algorithms”, Elsevier Publication,ScienceDirect.com,13 june 2012.

[4] J. Jayabharathy\ S. Kanmani2 and A. Ayeshaa Parveen1, ”Document Clustering and Topic Discovery based on Semantic Similarity in Scientific Literature” , 2011 IEEE.

[5] Jeong Hee Hwang a, Keun Ho Ryu b, “A weighted common structure based clustering technique for XML documents”, Elsevier Publication,ScienceDirect.com, 2010.

[6] Khaled B. Shaban, “A Semantic Approach for Document Clustering”, JOURNAL OF SOFTWARE, VOL. 4, NO. 5, JULY 2009.

[7] B.Drakshayani, E V Prasad, “Text Document Clustering based on Semantics ”, International Journal of Computer Applications (0975 – 8887) Volume 45– No.4, May 2012.

[8] Neepa Shah, Sunita Mahajan, “ Semantic based Document Clustering: A Detailed Review”, International Journal of Computer Applications (0975 – 8887) Volume 52– No.5, August 2012.

[9] Chien-Liang Liua,∗, Tao-HsingChangb, Hsuan-HsunLic, “Clustering documents with labeled and unlabeled documents using fuzzy semi-Kmeans ”,Elsevier Publication,ScienceDirect.com, 221(2013)48–64.

[10] Vivek Kumar Singh, Nisha Tiwari, Shekhar Garg, “Document Clustering using K-means, Heuristic K-means and Fuzzy C-means” , Proccedings of International Conference on Computational Intelligence and Communication Networks, IEEE , 10.1109/CICN.2011.62, pp.297-301.

[11] Daniel Díaz-Sánchez, Florina Almenarez, Andrés Marín, Davide Proserpio, and Patricia Arias Cabarcos, “Media Cloud: An Open Cloud Computing Middleware for Content Management”, IEEE Transactions on Consumer Electronics, Vol. 57, No. 2,May2011, doi: 10.1109/TCE.2011.5955247

[12] David Sánchez , Montserrat Batet, David Isern, Aida Valls, "Ontology-based semantic similarity: A new feature-based approach", Journal of Expert systems with applications , Elseveir , 2012, Vol.no.39, pp. 7718-7728.R.Priyadarshini, LathaTamilselvan , "Document based semantic content management system", Information technology journal.

[13] Loulwah AlSumait, Carlotta Domeniconi, “Local Semantic Kernels for Text Document Clustering,” In Workshop on Text Mining, SIAM International Conference on Data Mining, 2007