Journal of Network Communications and Emerging Technologies (JNCET) www.jncet.org
Volume 5, Special Issue 2, December (2015)

Clustering documents in information Retrieval System using ROCK

Sunita Rani
B.S.Anangpuria Institute of Technology & Mgt., Faridabad,India.

Abstract–Document clustering has important role in information retrieval systems. In an information retrieval system documents are parsed and terms are extracted. All of these terms have been put in an index. Many existing document clustering techniques use the ‘‘bag-of-words’’ model to represent the content of a document. However, this representation is only effective for grouping related documents when these documents share a large proportion of lexically equivalent terms. But the problem of unstructured information and a corpus having documents of different contexts will make the clustering task difficult. In this paper a robust clustering algorithm for categorical attributes ROCK is used to cluster documents for a sample set of documents.

Index Terms – Cluster, ROCK, Bag-of-words.

This paper is presented at International Conference on Recent Trends in Computer and information Technology Research on 25th& 26th September (2015) conducted by B. S. Anangpuria Instituteof Technology & Management, Village-Alampur, Ballabgarh-Sohna Road,Faridabad.

1. INTRODUCTION

An information retrieval system satisfies the information need of a user. Search engines are also a type of information retrieval systems which is used widely now a days .The great amount of scientific information being published makes it difficult for users of search engines to identify relevant information. For example, in the biomedical domain alone around 1,800 new papers are published daily (Hunter and Cohen 2006). Automatic document clustering provides a possible solution to this information overload problem, whereby users can quickly visualize the search space or search results, using labeled clusters of articles that have been grouped into topical and sub-topical categories. Automatic document clustering (that automatically groups related documents into clusters) is a powerful technique for large-scale topic discovery from text that can help to tackle the problem of information overload. For example, document clustering allows unsupervised discovery of the main topics or themes of the documents within a corpus. This is referred to as clustering based navigation of the search space.

2. RELATED WORK

A literature survey shows that it is very difficult to extract context of the information seeker of the IR system. Lopes [12] proposed context taxonomy for IR composed of two categorizations, one for the context features potentially useful in an IR system and other for possible uses of these features in an IR system. In this proposal, context is considered an interactional problem. It is considered that it does not only deal with the environmental features surrounding the user and its activities, but also concerns the interaction in other tasks and situations in similar domains. Kelly [16] suggests that it may be possible to infer topic familiarity from information search behaviour.

Lynda Tamine-Lechani et al. [14] proposed a multi-dimensional concept of context in IR. In the paper five context specific dimensions have been proposed which are device context, spatio-temporal context, user context, task/problem context and document context. In [15] N.J. Belkin et al.

Propose how to use user’s context to personalize IR system. Abdelkrim Bouramoul et al. [13] used an incremental approach to categorize users by constructing a contextual base. The latter was composed of two types of context (static and dynamic) obtained using the users’ profiles.

A.K.Sharma et al [19] proposed an ontology driven pre and post ranking based IR system. In the current paper the concept of context is extended for deciding the pre and post ranking of the documents in the IR system.

Sunita et al propose a context based indexing and ranking in information retrieval systems [11]. In that paper index considered is a context based index in which all the documents have been clustered according to their context.

The components of context based ranking system are as follows:

(a) Indexer : The job of the indexer is to parse the documents of the page repository and make entry of every token in the index.

(b)Context Based Index: This index stores all the tokens of the documents in the corpus. The index also stores the frequency of the tokens in different documents.

(c)Context Repository :It is list of all the available contexts in the IR system. If a new document is added in the repository then indexer decides the context of the documents and the document is added in the list of documents in that context.

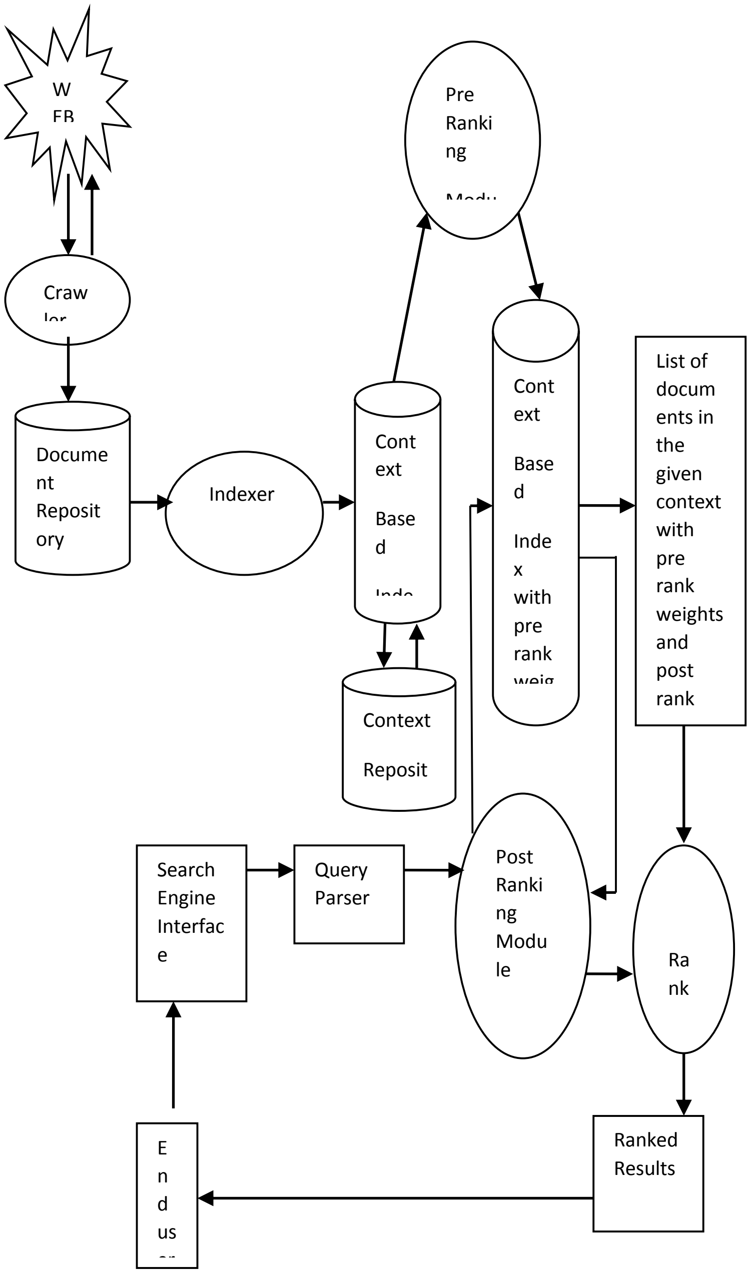


Fig.-1: Architecture of Context Based Ranking System

(d)Pre Ranking Module: This module assigns a pre rank weight to all the documents. This weight will decide the pre rank of the document.

(e)Post Ranking Module: This module calculates post rank weight of all the documents available in a context selected by the user for the query. This weight will further decide the final rank of the document.

(d)Context Based Index with pre rank weights: This is an index storing all the documents with their assigned context and also the pre rank weight of the document. This index is used by the ranker module to calculate the final rank of the document

An effort has been made to cluster documents according to their similarity. ROCK (RObust Clustering using linKs) algorithm is used to cluster a set of sample documents. It reduces the no of documents returned to the user.The architecture of the context based ranking system [11] is shown in Fig-1.

3. ROCK Algorithm for Clustering

ROCK is a hierarchical clustering technique to cluster documents. The methods for hierarchical clustering can be classified as either being agglomerative (bottom-up) or divisive (top-down), based on how the hierarchical decomposition is formed. To compensate for the rigidity of merging or splitting, the quality of hierarchical agglomeration can be improved by analyzing object linkage at each hierarchical partitioning (such as in ROCK and Chameleon), or by first performing micro clustering (that is, grouping objects into micro clusters) and then operating on the micro clusters with other clustering techniques, such as iterative relocation (as in BIRCH). ROCK plays an important role in clustering as it defines links between neighbours and thus results in good quality clusters. Here is a brief introduction about the ROCK algorithm. ROCK [1][6] performs agglomerative hierarchical clustering and explores the concept of links for data with categorical attributes. The various attributes are defined below:-

- *Links* - The number of common neighbours between two objects.
- *Neighbours* - If similarity between two points exceeds certain similarity threshold, they are neighbours i.e., if similarity (A,B)≥θ then only two points A, B are neighbours, where similarity is a similarity function and θ is a user-specified threshold.
- *Criterion Function* - The objective is to maximize the criterion function to get the good quality clusters. By maximizing we mean maximizing the sum of links of intra

cluster point pairs while minimizing the sum of links of inter cluster point pairs.

$$\mathrm{El} = \sum_{\mathrm{i=1}}^{\mathrm{k}} \mathrm{ni} * \sum_{\mathrm{i=Pq,Pr \in Ci}}^{\mathrm{k}} \mathrm{ni} \; \frac{link(Pq, \mathrm{Pr})}{\mathrm{Ni}^{\;1+2f(\theta\,)}}$$

Where Ci denotes cluster i, ni is the number of points in Ci, k is the number of cluster, θ is the similarity threshold.

- *Goodness Measure* - While performing clustering the motive of using goodness measure is – to maximize the criterion function and to identify the best pair of clusters to be merged at each step of ROCK.

Jaccard's coefficient is a good similarity measure because it can find the similarity between the categorical data. For sets A and B of keywords used in the documents, the Jaccard coefficient [4] may be defined as follows:

(A, B) = (|A) B|) / (A Y B|)

After determining the similarity for each pair of documents in the sample data set of N documents, the calculated similarity will be represented in the form of N x N Similarity Matrix, which will be transformed into an Adjacency Matrix with the help of similarity threshold θ≥0.4 (i.e., if similarity (A, B)≥θ then only two documents A, B are neighbours). The Adjacency Matrix is then multiplied by itself (i.e. A x A) to generate Link Matrix. Finally, the approach is to apply Criterion Function and Goodness Measure in an iterative fashion until we get clusters of similar documents representing different subject areas as well as noise, if any. Jaccard coefficient also known as Tanimoto coefficient is the best suited similarity coefficient for finding the similarity between the categorical data. It finds out the similarity by finding the intersection among the two documents divided by the union of the two documents [6]. It works on the mechanism of finding the similar strings among the two documents. If the value of the strings matched between the two documents are more, then they are similar to each other but if the value is less then they both are dissimilar. The Jaccard's value lies between 0 and1. And if the value is 0 then both the documents are different and if the value is 1 then both the documents are just same. A threshold value has to be defined to get the desired results of similarity.

4. ROCK ALGORITHM

ROCK

(A sample set of documents. Number of k clusters to be

found. The similarity threshold for this task: θ≥0.3)

{

Take k and θ≥0.3

BEGIN

1. Initially, place each document into a separate cluster.

2. Construction of Similarity Matrix: Constructing the similarity matrix by computing similarity for each pair of queries (A,B) using measure for instance i.e.

Similarity (A, B) = (|A ∩ B|)/(A U B|)

3. Computation of Adjacency Matrix: Compute Adjacency Matrix (A) using similarity threshold θ≥0.3 i.e. if similarity(A, B)≥θ then 1;else 0

4. Computation of Links: Compute Link Matrix by multiplying Adjacency Matrix to itself i.e. A x A to find the number of links.

5. Calculation of Goodness Measure: The goodness measure for each pair of documents is calculated by using the following function:

$$\mathrm{g(Ci, Cj)}$$

$$= \frac{link[Ci, Cj]}{(\mathrm{ni+nj})^{\;1+2f(\theta)} - (\mathrm{ni})^{\;1+2f(\theta)} - (\mathrm{nj})^{\;1+2f(\theta)}}$$

Where f (θ) = (1-θ)/(1+θ).

6. Merge the two documents with the highest similarity (goodness measure).

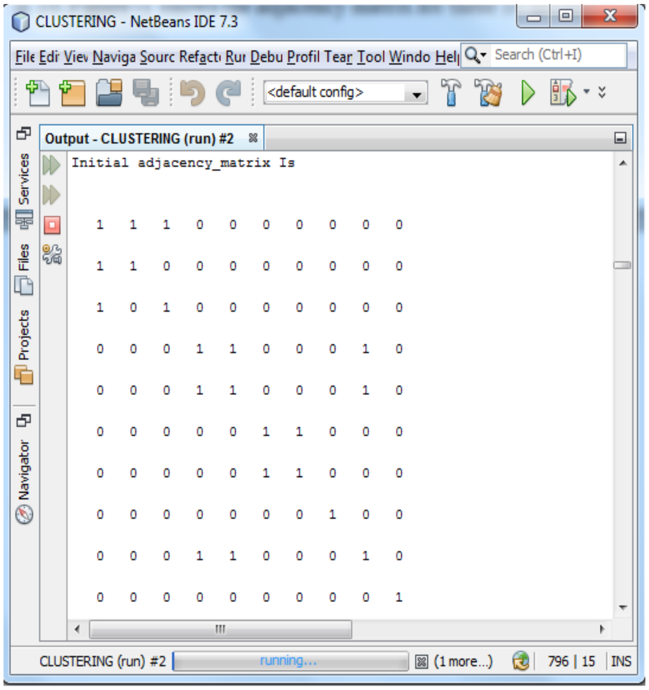
7.When no more entry exists in the goodness measure table then stop algorithm by resulting in k number of clusters and noise (if any) otherwise go to step 4.

END

}

5. IMPLEMENTATIONS

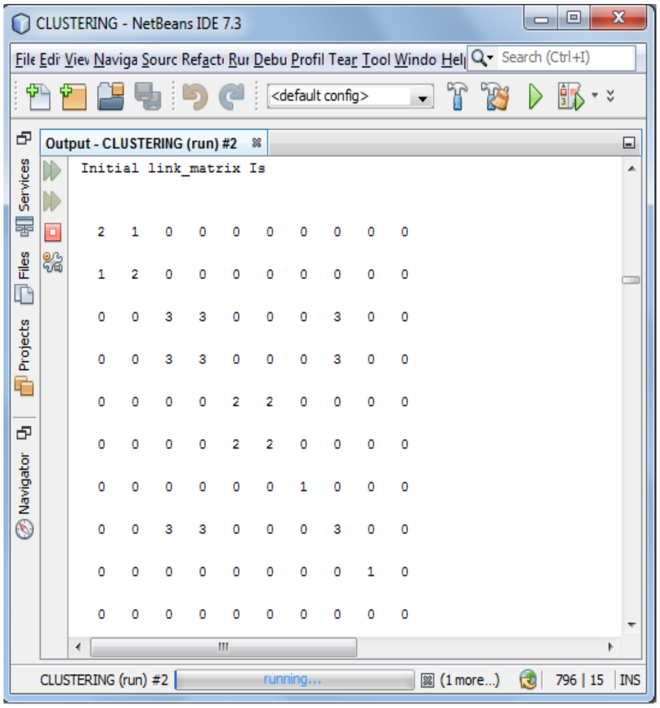
This algorithm results in the specified number of clusters. The algorithm has been implemented in java for a sample set of documents. The no of documents in the document set is 10. Figure-2 shows the adjacency matrix for these 10 documents. Then a link matrix is calculated from this adjacency matrix using the ROCK algorithm. Figure – 3 shows a link matrix. Then the ROCK algorithm will generate initial 10 clusters. Figure – 4 display a list of initial 10 clusters. Then these 10 clusters are further merged into 5 clusters using goodness measures. Figure – 5 display final 5 clusters.



```
Initial adjacency_matrix Is


 1   1   1   0   0   0   0   0   0   0

 1   1   0   0   0   0   0   0   0   0

 1   0   1   0   0   0   0   0   0   0

 0   0   0   1   1   0   0   0   1   0

 0   0   0   1   1   0   0   0   1   0

 0   0   0   0   0   1   1   0   0   0

 0   0   0   0   0   1   1   0   0   0

 0   0   0   0   0   0   0   1   0   0

 0   0   0   1   1   0   0   0   1   0

 0   0   0   0   0   0   0   0   0   1
```

CLUSTERING - NetBeans IDE 7.3
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
<default config>
Output - CLUSTERING (run) #2
Services
Files
Projects
Navigator
CLUSTERING (run) #2
running...
(1 more...)
796 | 15
INS

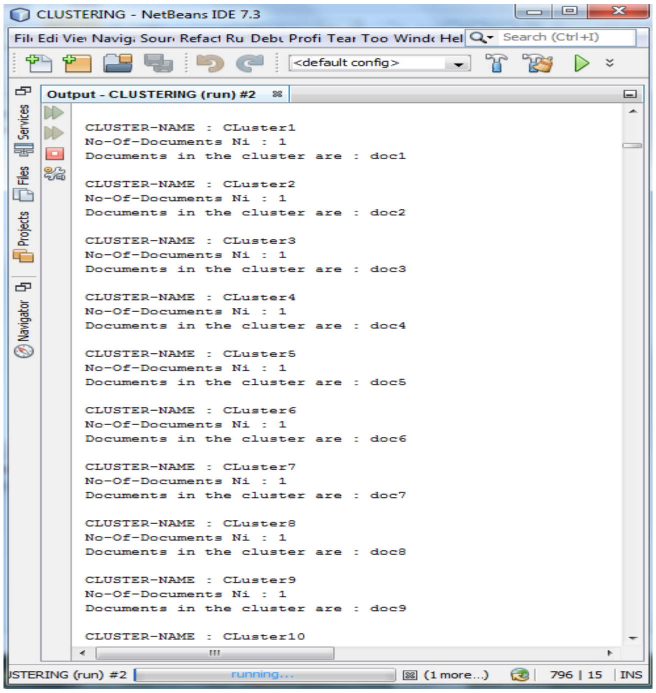
Fig. 1. Adjacency matrix of 10 sample documents



```
Initial link_matrix Is


 2   1   0   0   0   0   0   0   0   0

 1   2   0   0   0   0   0   0   0   0

 0   0   3   3   0   0   0   3   0   0

 0   0   3   3   0   0   0   3   0   0

 0   0   0   0   2   2   0   0   0   0

 0   0   0   0   2   2   0   0   0   0

 0   0   0   0   0   0   1   0   0   0

 0   0   3   3   0   0   0   3   0   0

 0   0   0   0   0   0   0   0   1   0

 0   0   0   0   0   0   0   0   0   0
```

CLUSTERING - NetBeans IDE 7.3
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
<default config>
Output - CLUSTERING (run) #2
Services
Files
Projects
Navigator
CLUSTERING (run) #2
running...
(1 more...)
796 | 15
INS

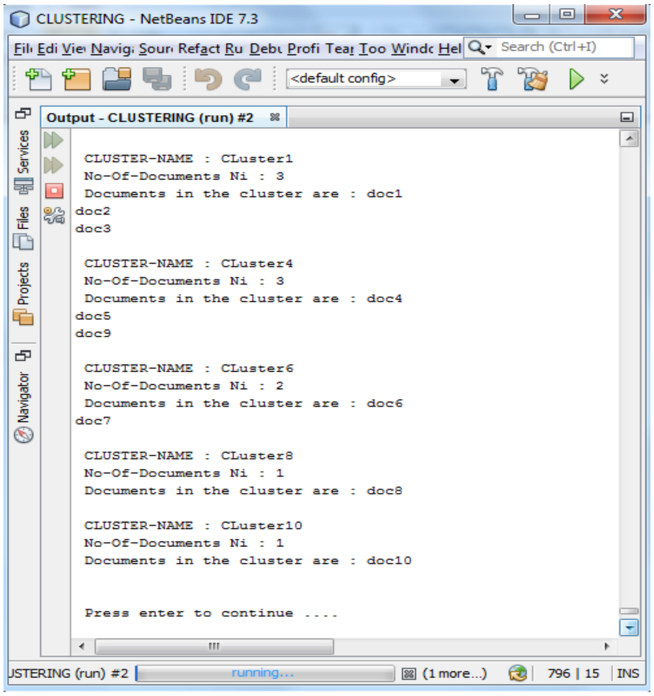
Fig. 2. Link matrix



```
CLUSTER-NAME  : CLuster1
No-Of-Documents Ni  : 1
Documents in the cluster are : doc1

CLUSTER-NAME  : CLuster2
No-Of-Documents Ni  : 1
Documents in the cluster are : doc2

CLUSTER-NAME  : CLuster3
No-Of-Documents Ni  : 1
Documents in the cluster are : doc3

CLUSTER-NAME  : CLuster4
No-Of-Documents Ni  : 1
Documents in the cluster are : doc4

CLUSTER-NAME  : CLuster5
No-Of-Documents Ni  : 1
Documents in the cluster are : doc5

CLUSTER-NAME  : CLuster6
No-Of-Documents Ni  : 1
Documents in the cluster are : doc6

CLUSTER-NAME  : CLuster7
No-Of-Documents Ni  : 1
Documents in the cluster are : doc7

CLUSTER-NAME  : CLuster8
No-Of-Documents Ni  : 1
Documents in the cluster are : doc8

CLUSTER-NAME  : CLuster9
No-Of-Documents Ni  : 1
Documents in the cluster are : doc9

CLUSTER-NAME  : CLuster10
```

CLUSTERING - NetBeans IDE 7.3
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
<default config>
Output - CLUSTERING (run) #2
Services
Files
Projects
Navigator
STERING (run) #2
running...
(1 more...)
796 | 15
INS

Fig. 3. . Initial 10 clusters for 10 documents



```
 CLUSTER-NAME  : CLuster1
 No-Of-Documents Ni  : 3
 Documents in the cluster are : doc1
doc2
doc3

 CLUSTER-NAME  : CLuster4
 No-Of-Documents Ni  : 3
 Documents in the cluster are : doc4
doc5
doc9

 CLUSTER-NAME  : CLuster6
 No-Of-Documents Ni  : 2
 Documents in the cluster are : doc6
doc7

 CLUSTER-NAME  : CLuster8
 No-Of-Documents Ni  : 1
 Documents in the cluster are : doc8

 CLUSTER-NAME  : CLuster10
 No-Of-Documents Ni  : 1
 Documents in the cluster are : doc10


 Press enter to continue ....
```

CLUSTERING - NetBeans IDE 7.3
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
<default config>
Output - CLUSTERING (run) #2
Services
Files
Projects
Navigator
STERING (run) #2
running...
(1 more...)
796 | 15
INS

Fig. 4. Final 5 clusters after merging

6. RESULTS AND DISCUSSIONS

As in our example we have implemented the clustering on a corpus having 10 sample documents only. The implementation further creates 5 clusters for the prescribed goodness measure threshold. As we can see from the snapshot Cluster1 is having 3 documents, Cluster4 is having 3 documents, Cluster6 is having 2 documents, Cluster8 is having 1 document and Cluster10 is having 1 document. If we will not perform the clustering then for any user query almost all the 10 documents will be retrieved to the user. In clustered IR system if a user fire a query then documents from the cluster which is most relevant to the user query will be retrieved. For example let suppose only two documents are relevant to user query in this sample IR system.

We compare proposed IR system (having clustered documents) with conventional IR system on a performance measure called precision. Precision is the fraction of the documents retrieved that are relevant to the user's information need.

$$\text{Precision} = \frac{|\{\text{relevent documents}\} \cap \{\text{retreived documnets}\}|}{|\{\text{retreived documents}\}|}$$

For conventional IR system

Relevant documents = 2 retrieved documents = 10 so

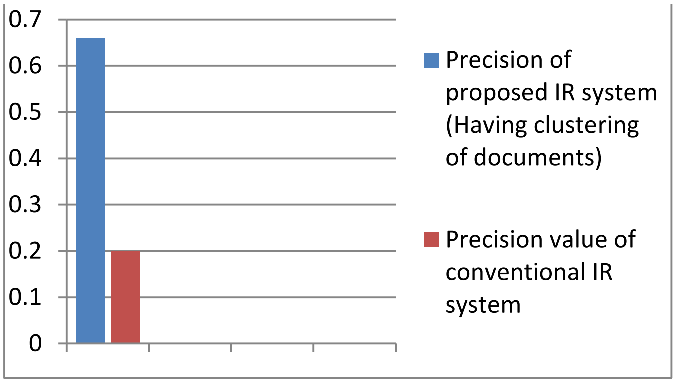
precision = 0.2

For proposed IR system (Having clustered documents) Let suppose query is relevant to Cluster4. Cluster4 stores three documents out of two documents are relevant to user's query. So

Relevant documents = 2 retrieved documents = 3 (Cluster4 have 3 documents)

so precision = 2/3 = 0.66

Figure-6 showing bar chart comparing the precision value of two systems.



7. Precision based comparison of proposed and conventional IR systems.

7. CONCLUSION AND FUTURE WORK

In this paper the role of clustering documents in information retrieval systems has been discussed. An implementation of ROCK algorithm is also discussed with a initial corpus of 10 documents. It is concluded that documents can be clusters easily using ROCK algorithm. A future work how to use clusters in information retrieval systems is needed. Further work is also needed to clusters documents according to the context of the document.

REFERENCES

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

[2] Dutta M,Kaskoti Mahanta A,Pujari Arun K, "QROCK:A quick version of the ROCK algorithm for clustering of categorical data," Pattern Recognition Letters,Vol.26,Nov.2005, pp. 2364-2373, doi: 10.1016/ j.patrec. 2005. 04. 008

[3] Z. Huang. "A Fast Clustering Algorithm to Cluster Very Large Categorical Data Sets in Data Mining", CSIRO Mathematical and Information Sciences, Australia.

[4] Anna Huang, "Similarity Measures for Text Document Clustering", Volume: 2008, Issue: April, Pages: 49–56, Mendeley.

[5] Shyam Boriah Varun Chandola Vipin Kumar, "Similarity Measures for Categorical Data: A Comparative Evaluation", 2008 Volume: 30, Issue: 2, Publisher: Citeseer,Pages: 3.

[6] Rizwan Ahmad,Dr. Aasia Khanum,Document, "Topic Generation in Text Mining by Using Cluster analysis with EROCK", 2010, International Journal of Computer Science & Security, Volume (4) : Issue (2).

[7] Rui Xu, Donald Wunsch "Survey of clustering algorithms", Volume: 16, Issue: 3, Publisher: Institute of Electrical and Electronics Engineers, Inc, 445 Hoes Ln, Piscataway, NJ, 08854-1331, USA,, Pages: 645-678.

[8] Florian Beil,Martin Ester,Xiaowei Xu, "Frequent Term-Based Text Clustering" ,in 2002 Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining.

[9] Athena Vakali , Jaroslav Pokorny , Theodore Dalamagas, "An Overview of Web Data Clustering Practices",2004, Proceedings of the 9th International Conference on Extending Database Technology - EDBT'04, Springer-LNCS 3268.

[10] Qiongbing Zhang, Lixin Ding, Shanshan Zhang, "A Genetic Evolutionary ROCK Algorithm" 2010 International Conference .

[11] Sunita Rani , Vinod Jain , Geetanjali Gandhi "Context Based Indexing and Ranking in Information Retrieval Systems" International Journal of Computer Science and Management Research Vol 2 Issue 4 April 2013ISSN 2278-733X.

[12] C.T. Lopes, "Context Features and their use in Information Retrieval," In : Third BCS-IRSG Symposium on Future Directions in Information Access. Padua, Italy, September 2009.

[13] A. Bouramoul, M. K. Kholladi, and B .L. Doan, "PRESY : A Context based query reformulation tool for information retrieval on the Web," In JCS : Journal of Computer Science, Vol 6, Issue 4, pp. 470-477, 2010., ISSN 1549-3636, New York, USA. April 2010.

[14] L. Tamine, M. Boughanem, and M. Daoud, "Evaluation of contextual information retrieval effectiveness: overview of issues and research," In Journal of Knowledge and Information Systems. Volume 24 Issue 1, pp. 1-34. Springer, Londres, United Kingdom. July 2010.

[15] Belkin, G. Muresan, X. Zhang, "Using User's Context for IR Personalization," In Proceedings of the ACM/SIGIR Workshop on Information Retrieval in Context 2004.

[16] Kelly, D. & Cool, C. (2002) Effects of topic familiarity on information search behavior. In Proceedings of the Second ACM/IEEE-CS Joint Conference on Digital Libraries – JCDL 2002 (pp. 74-75). New York: ACM.

[17] Abdelkrim Bouramoul1, Mohamed-Khireddine Kholladi1 and Bich-Lien Doan2 "Using Context To Improve The Evaluation Of Information Retrieval Systems" Computer Science Department, Misc Laboratory, University of Mentouri Constantine. B.P. 325, Constantine 25017, Algeria.

[18] Dr. A.K.Sharma, Parul Gupta. "Context based Indexing in Search Engines using Ontology." IJCA: International Journal on Computer Application,Vol 1, No. 14,Pages 302 ISBN :- 978-93-80746-13-5

[19] Dr. A.K.Sharma, Parul Gupta. "Ontology driven Pre and Post Ranking based Information Retrieval in Web Search Engines" in IJCSE : International Journal on Computer Science ang Engineering, Vol 4 Pages 1241-1246, ISSN: - 0975-3397.

Authors

Sunita Tomar is working as a lecturer in Computer Application Department at B.S.Anangpuria Institute of Technology and Management, Faridabad since Eight years. She has completed Master of technology in 2013 from Jamia Hamdard University.Her area of Research includes Database and IR systems.