iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 2 **Issue:** X **Month of publication:** October 2014

DOI:

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology(IJRASET)

Proposed & Implemented Clustering Algorithm for Indexing in Search Engine

Sneh kalra[#1]

Ph.d Scholar, @Modern Vidya Niketan University

Abstract— This paper proposes clustering algorithm for implementing indexing phase of search engine. The goal of making an index is to optimize speed and performance in finding relevant documents for a search query. Without an index, the search engine would scan every document in the corpus, which would require considerable time and computing power. Main goal is to improve the quality of web search engines. A complete search index would make it possible to find anything easily.
Keywords—Inverted Index, Posting List, Similarity Matrix, Parsed data, Clustered Data

I. INTRODUCTION

Search engine indexing is the process of a search engine collecting, parses and stores data for use by the search engine. It is the search engine index that provides the results for search queries, and pages that are stored within the search engine index that appear on the search engine results page. Without a search engine index, the search engine would take considerable amounts of time and effort each time a search query was initiated, as the search engine would have to search not only every web page or piece of data that has to do with the particular keyword used in the search query, but every other piece of information it has access to, to ensure that it is not missing something that has something to do with the particular keyword. Search engine spiders, are how the search engine index gets its information, as well as keeping it up to date and free of spam.

So the indexing phase of search engine can be viewed as a Web Content Mining process Starting from a collection of unstructured documents, the indexer extracts a large amount of information like the list of documents, which contain a given term. It also keeps account of number of all the occurrences of each term within every document. This information is maintained in an index, which is usually represented using an inverted file (IF). The index consists of an array of the posting lists where each posting list is associated with a term and contains the term as well as the identifiers of the documents containing the term. For example, consider the posting list ((Study; 5) 2, 4, 10, 24, 27) indicating that the term Study appears in five documents having the document identifiers 2,4, 10,24,27 respectively. The following figure shows the example entries in index file. Clustering is a widely adopted technique aimed at dividing a collection of data into disjoint groups of homogenous elements.

Term	No. of docs in which term appears	Doc id of docs in which term appears
Index	50	12,34,45,49…
Search	59	15,20,34,55…
catalog	15	3,6,9,12…

Fig. 1 Example Entries in Index File

A cluster is therefore a collection of objects, which are similar between them and are dissimilar to the objects belonging to other clusters. Clustering algorithms attempt to group together the documents based on their similarities. Thus documents relating to a certain topic will hopefully be placed in a single cluster. So if the documents are clustered, comparisons of the documents against the user's query are only needed with certain clusters and not with the whole collection of documents.

II. RELATED WORK

In this paper, a review of previous work on index organization is given. In this field of index organization and maintenance, many algorithms and techniques have already been proposed but they seem to be less efficient in efficiently accessing the index.

First Proposed algorithm was Threshold based clustering algorithm in which the number of clusters is unknown. However, two documents are classified to the same cluster if the similarity between them is below a specified threshold. This threshold is defined by the user before the algorithm starts. It is easy to see that if the threshold is small; all the elements will get assigned to different clusters. If the threshold is large, the elements may get assigned to just one cluster. Thus the algorithm is sensitive to specification of threshold.

Another Proposed work was Suffix Tree Clustering Algorithm In STC, as documents may share more than one phrase with other documents. Each document might appear in a number of base clusters. In some cases, the overlap between the clusters goes very high.

The proposed algorithm has tried to remove the shortcomings of the existing algorithms. It produces a better ordering of the documents in the cluster. This algorithm picks the first document as cluster representative, then selects the most similar document to it and puts it in the cluster, it further selects document which is most similar to the currently selected document and repeats until the first cluster becomes full with n/k documents. The same process is then repeated to form the rest of the clusters.

Thus the most similar documents are accumulated in the same cluster and are assigned consecutive document identifiers. Thus the algorithm is more efficient in compression of the index.
Another proposed work was the K means clustering algorithm, which initially chooses k documents as cluster representatives and then assigns the remaining n-k documents to one of these clusters on the basis of similarity between the documents. New centroids for the k clusters are recomputed and documents are reassigned according to their similarity with the k new centroids. This process repeats until the position of the centroids become stable. Computing new centroids is expensive for large values of n and the number of iterations required to converge may be large.
The given paper proposes Algorithm for Clustering based Indexing and then implements the proposed algorithm for Indexing of search engine.

III. PROPOSED WORK

The indexing phase of the search engine collects information from the web documents gathered in the web repository .This global large sized index is stored in the form of inverted files. The paper has proposed an algorithm for indexing using Clustering and then implements the algorithm to show the indexing in Search Engine.

A. Architecture of Clustering Based Indexing in Search Engines

Let D = {D1, D2, . . . , DN} be a collection of N textual documents to which consecutive integer document identifiers 1, . . . ,N are initially assigned. Moreover, let T be the number of distinct terms ti, i = 1, . .. , T present in the documents, and Mt the average length of terms.
The total size CSize (D) [27] of an IF index for D can be written as:
C size(D) = CSizelexicon (T.μt) + ∑ Encodem(d_gaps(ti))

i=1 to T

where CSizelexicon (T.ut) is the number of bytes needed to code the lexicon, while d_gaps (ti) is the d_gap [28] representation of the posting list associated to term ti, and Encodem is a function that returns the number of bytes required to code a list of d gaps according to a given encoding method m.
The compression of index is achieved by applying clustering to the web pages so that the similar web pages are in the same cluster and hence assigned closer identifiers. A clustering algorithm has been proposed, which converts the individual documents into k ordered clusters, and hence documents are reassigned new document identifiers so that the documents in the same cluster get the consecutive document identifiers.

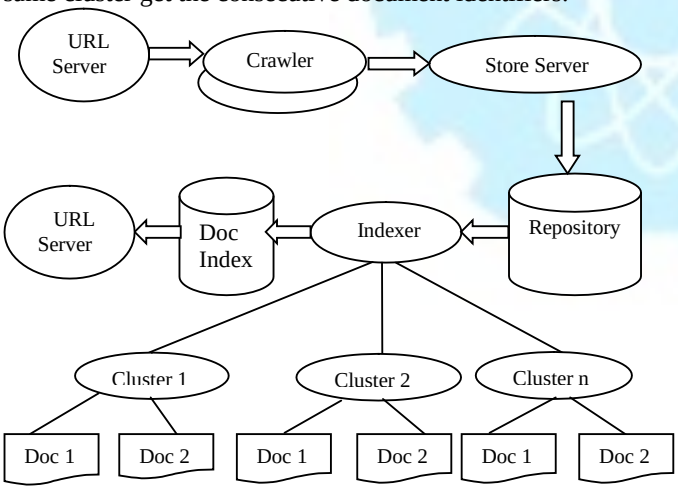


Fig. 2 Architecture of Clustering based indexing System in Search Engines

The clustering of the documents is done on the basis of similarity between the documents, which is first of all calculated using some similarity measure.

B. Algorithm for Computing the Similarity Matrix

Let D={D1, D2,……Dn) be the collection of N textual documents being crawled to which consecutive integers document identifiers 1…n are assigned. Each document Di can be represented by a corresponding set Si such that Si is a set of all the terms contained in Di. Let us denote that set by D* such that D*={S1, S2,…….. Sn}. The similarity of any two documents Si and Sj can be computed using the similarity measure
Similarity_measure (Si, Sj) =|Si Λ Sj | / |Si U Sj |

INPUT – The set D*= {S1, S2, S3, S4…Sn} where Si is a set of all the terms of document Di. The number k of clusters to create.
OUTPUT – k ordered clusters representing a reordering of D
The algorithm that calculates the similarity of each document with every other document using the similarity_measure given above is given in figure

```
Algorithm docum_similarity

for i=1 to n
begin
 sim[i][i]=0;
 for j=i+1 to n
 begin
     sim[i][j]=similarity_measure(Si,Sj)
     sim[j][i]=sim[i][j]
end for
end for
```

Fig. 3 Algorithm for computing similarity matrix (docum_similarity)

The above algorithm constructs the document similarity matrix. The number of calculations performed leads to formation of the upper triangular matrix. The rest of the values in the similarity matrix are assigned automatically as we know similarity_measure (i, j) = similarity_measure (j, i).

C. The Algorithm for Clustering

The clustering algorithm which clusters together the similar documents is given below:

```
Algorithm docum_clustering
i=1
for f=1 to k      //for number of clusters
begin
cf=0  //Initially cluster is empty with no document
for e=1 to n/k  // for no. of documents in one cluster
            begin
                        for j=1 to n
                         Select max from sim[i][j]
                          cf= cf U Si
                          D*=D* -Si
                          for l=1 to n
                          begin
                                                  sim[i][l]=0
                                                  sim[l][i]=0
                          end
```

Fig. 4 Algorithm for clustering (docum_clustering)

It may be noted that the algorithm starts with the first cluster which is empty initially. The first document from the collection is considered and put in the first cluster. Now, using the similarity matrix, the most similar document to it is considered. All the entries of the row and column associated with the first document are made zero as this document cannot be added to any other cluster. The most similar document picked is put in the same cluster. Now the second document that was considered takes the role of the first document and the most similar document to it is considered and this procedure repeats for n/k times when the first cluster gets full. Now the second cluster is considered and the same procedure repeats until all the clusters get full. Thus at the end, we get k clusters each with n/k number of similar documents

D. Example Illustrating Clusters Formation

Let us now have panoramic view as to how the clustering of the documents takes place. For e.g. if we have 10 documents – A, B, C, D, E, F, G, H, I, J & value of k is 2 i.e. 2 clusters are to be made, then according to the algorithm, the similarity among the documents is computed using the similarity measure and hence the formed similarity matrix using property that similarity measure (i,j) = similarity measure(j,i) will be

	A	B	C	D	E	F	G	H	I	J
A	**0**	**5**	**3**	**6**	**9**	**8**	**2**	**3**	**4**	**1**
B	**5**	**0**	**5**	**4**	**6**	**2**	**3**	**5**	**7**	**8**
C	**3**	**5**	**0**	**5**	**2**	**3**	**6**	**9**	**4**	**7**
D	**6**	**4**	**5**	**0**	**2**	**4**	**6**	**5**	**4**	**9**
E	**9**	**6**	**2**	**2**	**0**	**8**	**5**	**3**	**6**	**5**
F	**8**	**2**	**3**	**3**	**8**	**0**	**8**	**9**	**5**	**2**
G	**2**	**3**	**6**	**6**	**5**	**8**	**0**	**6**	**5**	**4**
H	**3**	**5**	**9**	**5**	**3**	**9**	**6**	**0**	**3**	**6**
I	**4**	**7**	**4**	**4**	**6**	**5**	**5**	**3**	**0**	**5**
J	**1**	**8**	**7**	**9**	**5**	**2**	**4**	**6**	**5**	**0**

Fig. 5 Full Similarity Matrix

The output after calculating similarity for first five documents will be:

	A	B	C	D	E	F	G	H	I	J
A	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
B	**0**	**0**	**0**	**4**	**0**	**0**	**3**	**0**	**7**	**8**
C	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
D	**0**	**4**	**0**	**0**	**0**	**0**	**6**	**0**	**4**	**9**
E	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
F	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
G	**0**	**3**	**0**	**6**	**0**	**0**	**0**	**0**	**5**	**4**
H	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
I	**0**	**7**	**0**	**4**	**0**	**0**	**5**	**0**	**0**	**5**
J	**0**	**8**	**0**	**9**	**0**	**0**	**4**	**0**	**5**	**0**

Fig. 6 Matrix after formation of first cluster

IV. IMPLEMENTED WORK

For indexing the documents, firstly we have to parse the documents. After that similarity matrix is created and then k means algorithm is applied for creating the clusters. Finally Clusters will be created.

A. Flowchart for creating clustering

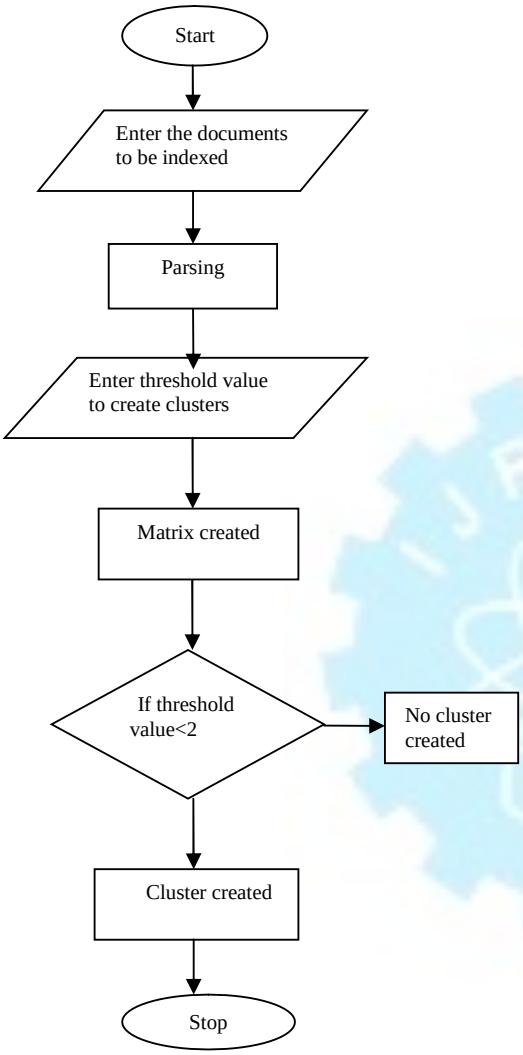


Fig. 7 Flowchart for creating clustering

B. Snapshots of Implemented Work

1) Given input & parsed data

The following snapshot represents the parsed data, Which is the initial step for indexing the data

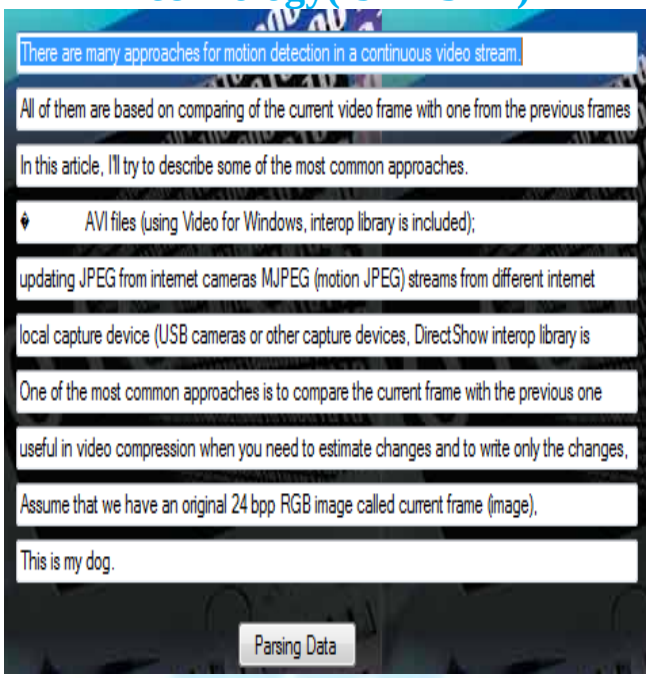


Fig. 8 Given Input

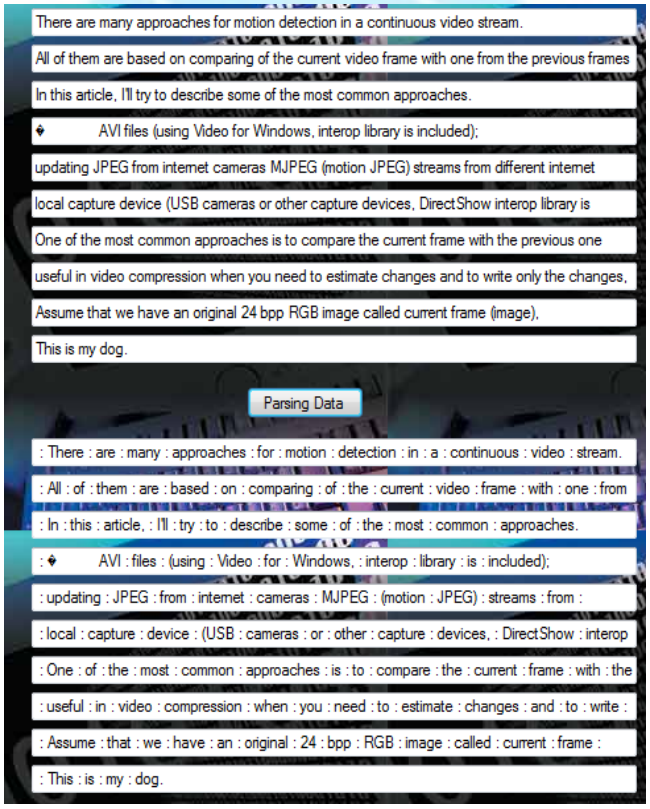


Fig.9 Parsed Data

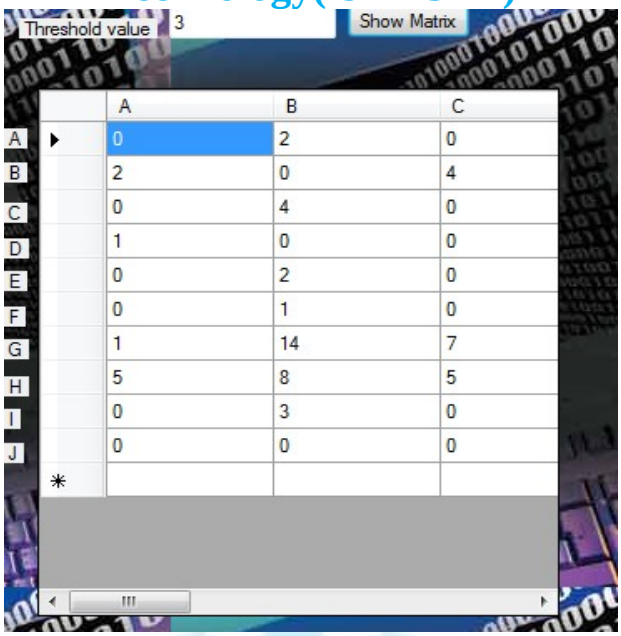


Fig. 10 Matrix representation

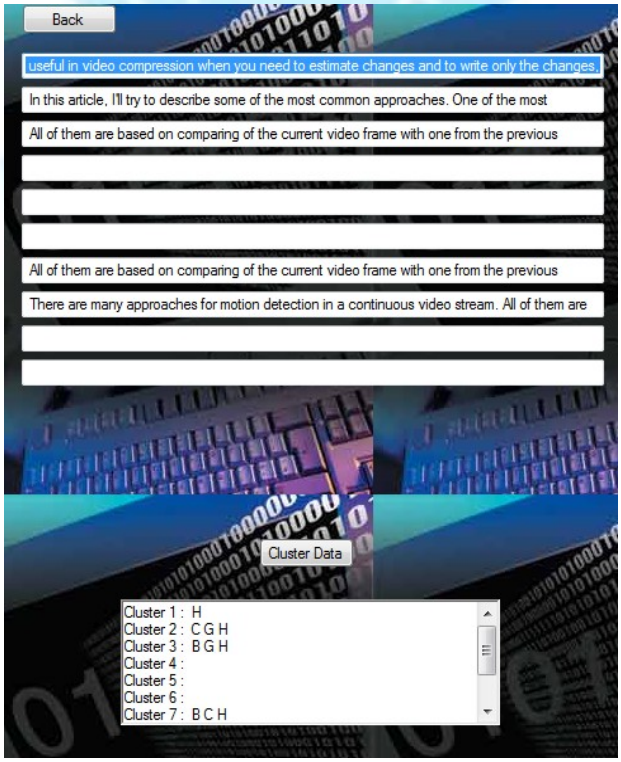


Fig. 11 Clustered data

V. CONCLUSION

This paper proposed an algorithm for indexing & implemented indexing phase of search engine. Indexing using clustering technique has improved the quality of indexing as clusters have been created on one level. The implemented algorithm is superior to the other algorithms as a summarizing and browsing tool. A critical look at the literature indicates that in contrast to the earlier proposed algorithms, the Implemented work produces a better ordering of the documents in the cluster.

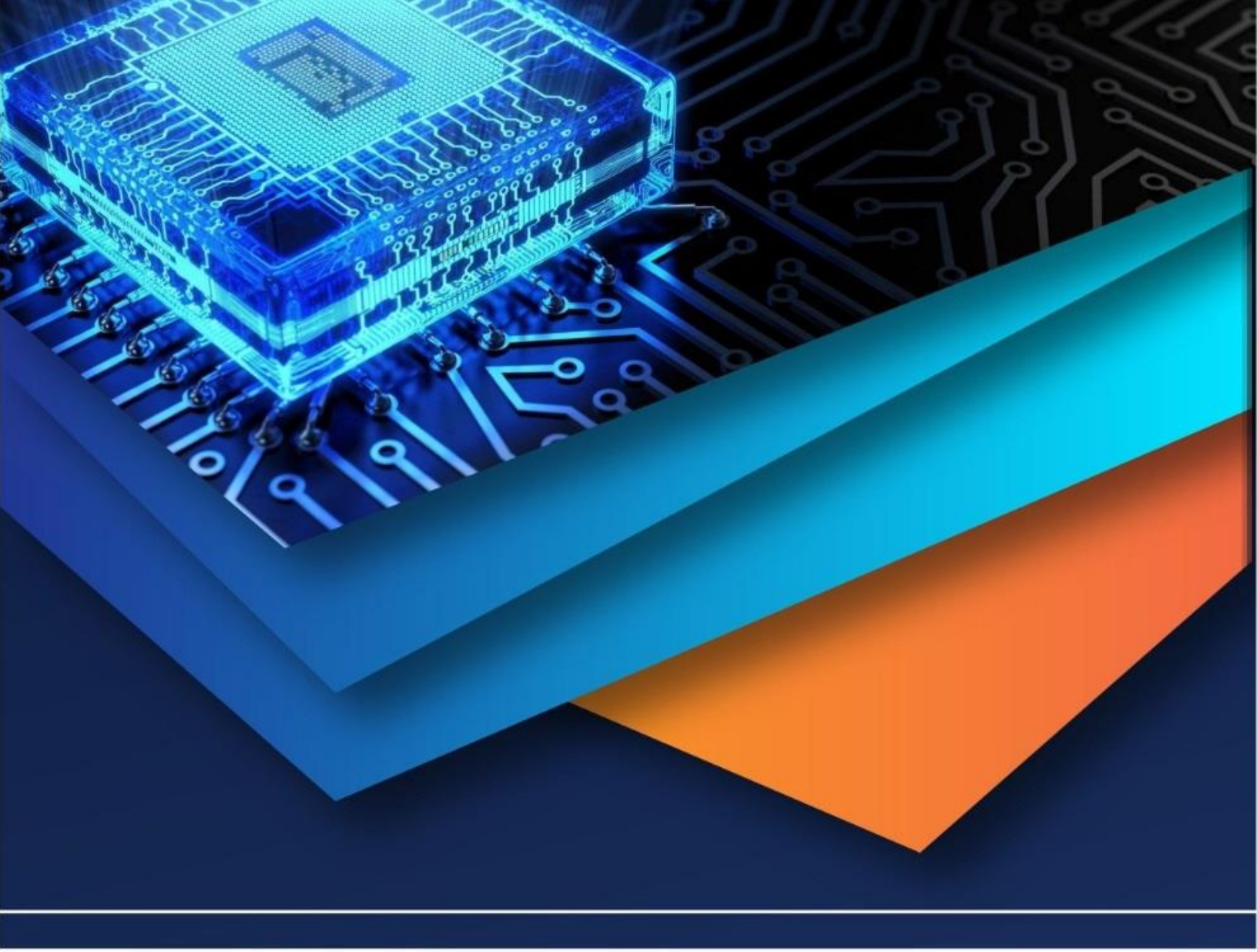
VI. ACKNOWLEDGMENT

It is with deep sense of gratitude and reverence that I express my sincere thanks to Mrs. Parul Gupta for her guidance, encouragement, help and useful suggestions throughout. Her untiring and painstaking efforts, methodical approach and individual help made it possible for me to complete this work in time. I consider myself very fortunate for having been associated with the scholar like her. Her affection, guidance and scientific approach served a veritable incentive for completion of this work.

I am also thankful to Dr.P.C.Vashisth, HOD CS & IT Department, MVN University, Palwal for his constant encouragement, valuable suggestions and moral support and blessings. This acknowledgement will remain incomplete if I fail to express my deep sense of obligation to my parents and God for their consistent blessings and encouragement.

REFERENCES

[1] Parul Gupta, Dr. A.K. Sharma, Hierarchical Clustering based Indexing in Search Engines, communicated to International Journal of Information and Communication Technology.
[2] A Framework for Hierarchical Clustering Based Indexing in Search Engines Parul Gupta , A.K. Sharma.
[3] Sanjiv K. Bhatia. Adaptive K-Means Clustering. American Association for Artificial Intelligence, 2004.
[4] G. Adami, P. Avesani, D. Sona, Clustering Documents in a Web Directory, in Proceedings of the 5th International Workshop on Web Information and Data Management (ACM WIDM 03), September 2003.
[5] Korfhage, R., 1997. Information Storage and Retrieval.
[6] Cooper. W.S., 1969. "Is inter indexer consistency a hobgoblin?" American
[7] The Anatomy of a Large-Scale Hypertextual Web Search Engine Sergey Brin and Lawrence Page Computer Science Department, Stanford University, Stanford, CA 94305, USA
[8] S. Chakrabarti. Mining the Web. Morgan Kaufmann, 2003.
[9] K. Alsabti, S. Ranka, and V. Singh, ªAn Efficient k-means Clustering Algorithm,º Proc. First Workshop High Performance Data Mining, Mar. 1998.
[10] Michael R. Anderberg. Cluster analysis for applications. Academic Press, 1973
[11] Kilgour, Frederick G."The Evolution of the Book search engine ". New York: Oxford University Press, 1998; Katz, Bill. Cuneiform to Computer: A History of Reference Sources. Lanham, MD: Scarecrow Press, 1998 .
[12] Weinberg, Bella Hass. "Indexes and Religion: Reflections on Research in the History of Indexes". The Indexer, 21 (3): 111-118 (1999).
[13] C. C. Aggarwal, F. Al-Garawi, and P. S. Yu. "Intelligent crawling on the World Wide Web with arbitrary predicates". In WWW10, Hong Kong, May 2001 .
[14] K. Alsabti, S. Ranka, and V. Singh, ªAn Efficient k-means Clustering Algorithm,º Proc. First Workshop High Performance Data Mining, Mar. 1998.
[15] Michael R. Anderberg. Cluster analysis for applications. Academic Press, 1973.
[16] Searching Research Papers Using Clustering and Text Mining (9781- 4673-6155-2/13/ © 2013 IEEE)



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY