A Novel Indexing Technique for Web Documents using Hierarchical Clustering

Deepti Gupta[†], **Komal Kumar Bhatia**[††], **and A.K. Sharma**[†††]

[†]School of Computer Science and Information Technology, Shobhit University Meerut, INDIA
[††]Department of Computer Science, YMCA Institute of Engineering Faridabad, INDIA
[†††] Department of Computer Science, YMCA Institute of Engineering Faridabad, INDIA

Summary
The information on the WWW is growing at an exponential rate; therefore, search engines are required to index the downloaded Web documents more efficiently. Web mining techniques like clustering can be used for this purpose. In this paper, a novel technique to index the documents is being proposed that not only indexes the documents more efficiently but also uses hierarchical clustering to keep the information based upon similarity measure and fuzzy string matching. This technique keeps the related documents in the same cluster so that searching of documents becomes more efficient in terms of time complexity.

Key words:
Search Engine, Indexer, Hierarchical Clustering.

1. Introduction

Information retrieval tools, like search engines [1], download web pages, texts, images and other multimedia from World Wide Web. A typical search engine (see Figure 1.1) comprises of the following components:

- Crawler: Given a URL, it combs through the pages on the web and gathers the required information for the search engine.
- Indexer: While an index of 100,000 documents can be queried within millisecond; a sequential scan may take hours. Hence the need of an indexer that optimizes speed and performance for finding relevant documents for a search query .Besides indexing, The Indexing process collects, parses and stores data in order to facilitate fast and accurate information retrieval.
- Page Repository: The information retrieved by the web crawler is stored in a database called page repository. The indexer indexes the various documents contained in repository. The documents are identified by doc ID, length and URL.

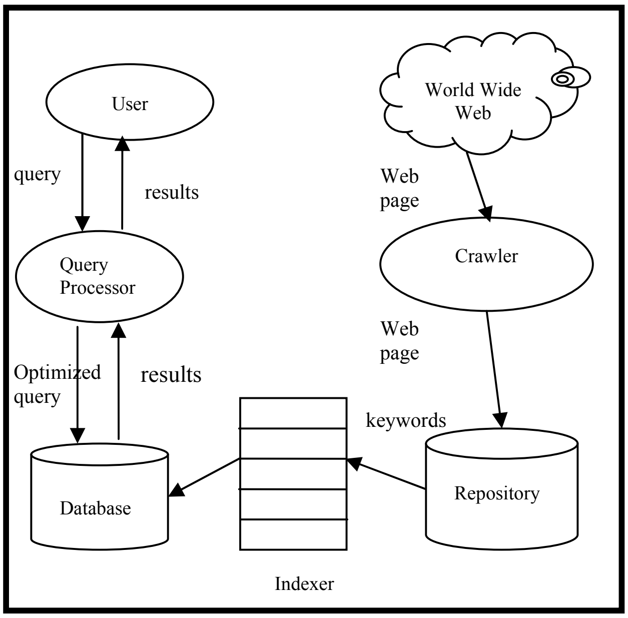


Figure 1.1: General architecture of the Search Engine

The performance of a search engine is limited because of the following problems:

- Low precision: It happens due to the irrelevance of many of the search results leading to difficulty in finding the relevant information. This problem is also termed as "Information Overkill".
- Low recall: The search engine is unable to index all the information in the web whereas the un-indexed information may also be relevant.

Thus, there is a need to develop efficient indexing technique.

In this paper, a novel technique is being proposed that not only indexes the downloaded web documents efficiently but also uses a web-mining technique [2] to make the indexed information searchable, enabling the search engines to provide more relevant results.

Manuscript received September 5, 2009
Manuscript revised September 20, 2009

2. Related Work

Willett [3] presents a detailed study of applying hierarchical clustering algorithms in the document clustering. Agglomerative Hierarchical Clustering [7, 8] algorithms have mostly been used. These algorithms are applied to large document collections. For example, single-link methods typically take O(n^2) time while complete-link methods typically take O (n^3) time.

Cutting et al. [4] adopted various partition-based clustering algorithms for clustering document such as Buckshot and Fractionation. Fractionation is an approximation to Agglomerative Hierarchical Clustering, where the search for the two closest clusters is not performed locally and in a bound region instead of searching globally as in the case of document clustering.

Jain [5] provides an elaborate survey of various clustering techniques. The study presents an overview of pattern clustering methods from a statistical pattern recognition perspective, with a goal of providing useful advices and references to fundamental concepts accessible to the broad community of clustering practitioners. It explicates the taxonomy of clustering techniques, and identifies cross-cutting themes and recent advances. Some important applications of clustering algorithms have been applied in various fields such as image segmentation, object recognition, and information retrieval.

Andrei [6] provides various commercial and scientific applications require analysis of user behavior in the internet. New web user sessions classification method is the main goal of this research. In this paper web usage analysis is described. Previously Levenshtein metric was applied to web sessions domain in hierarchical Clustering.

A critical look at the available literature indicates that till date only similarity measure was used to make clusters, resulting in only similar pages with less emphasis on relevancy of the pages. In this paper the fuzzy string matching (Levenshtein metric) has also been used to make clusters through keywords using agglomerative hierarchical clustering, leading to creating of clusters related and relevant of documents. There is the need of mathematical formula to generate the clusters. The proposed formula is $D_{ij} = \alpha\, d_{ij} + \beta\, s_{ij}$, where d_{ij} is Euclidean metric and s_{ij} Levenshtein metric, α and β are constants.

3. The Analytical Framework

A novel indexing technique based on an Agglomerative Hierarchical clustering algorithm is being proposed. It employs both Euclidean metric [9] and Levenshtein metric [6] for similarity measure and fuzzy string matching respectively. The downloaded document and the keywords contained there in and stored in a repository by the crawler see Figure 1.1.The indexer extracts all words from the entire set of documents and eliminates non-content-bearing words i.e. stop words such as “a”, “and”,” the” etc from each documents. For each document, the number of occurrences of each word is counted and a list of common words called keywords is generated in Table 3.1:

Table 3.1 Document and Keywords

Document	Keywords
Document 1	Crawler, Search engine, Security, and Crawl
Document 2	Crawler, Search Engine
Document 3	Database File, Database Management

Each keyword is assigned a word ID as shown in Table 3.2. For instance the Word ID of keyword of document 1 is listed in Table 3.2:

Table 3.2 Keywords and Word ID

Keywords	Word ID
Crawler	1
Search Engine	2
Security	3
Crawl	4

The information contained in Table 3.1 and Table 3.2 is employed to generate the required index and the steps followed are given below:

3.1 Step 1 - Graphical Representation

Create a table of Association of keywords and documents i.e. the Table 3.1 showed and entry for each keyword and the list of containing in each word as shown in Table 3.3. An equivalence graph represents keyword; document association is given in Figure 3.1.

3.2 Step 2 - Euclidean Distance

Calculate the Euclidean distance between the various points located in Figure 3.1. It may noted from figure that a coordinate (x, y) represents that a keywords x is present in document y. Euclidean metric is used to explicate the similarity measure for generating the clusters. It describes distance measures that are commonly used for computing the dissimilarity of objects described by interval-scaled variables and it is typically computed based on the distance between each pair of objects. The most popular distance measure is Euclidean distance, which is defined as

$$d(m,n)=\sqrt{}(x_{i1}-x_{j1})^2+(x_{i2}-x_{j2})^2+ \ldots+ (x_{in}-x_{jn})^2 \qquad \ldots.(3.1)$$

where $m=(x_{i1},x_{i2},\ldots.,x_{in})$ and $n=(x_{j1},x_{j2},\ldots\ldots.,x_{jn})$ are two n- dimensional data objects.

Euclidean distance between (1, 1) and (1, 7) is

$$d_{11}= \sqrt{}(1-1)^2+ (1-7)^2 = 6$$

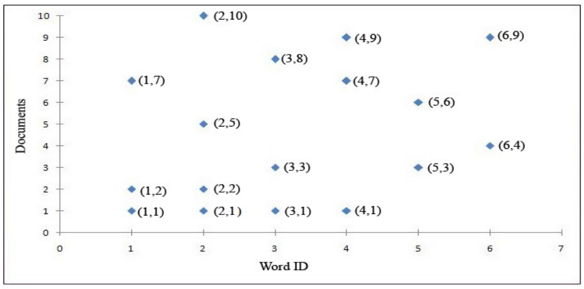


Figure 3.1:-Graphical representation between keyword ID and Documents

Keywords	Word ID	Documents
Crawler	1	1,2,7
Search Engine	2	1,2,5,10
Security	3	1,3,8
Crawl	4	1,7,9
Database Management	5	3,6
Database file	6	4,9

Table 3.3 : keywords,Word ID, Documents

3.3 Step 3 - Levenshtein Distance: A Fuzzy String Matching Algorithm

A new approach fuzzy string matching is added. Apply Levenshtein metric [10] is for fuzzy string matching on keywords which is used to generate the base clusters. Levenshtein distance is obtained by finding the cheapest way to transform one string into another. Transformations are the one-step operations of (single-phone) insertion, deletion and substitution. In the simplest versions, substitutions cost two units except when the source and target are identical in which case the cost is zero. Insertions and deletions cost half of that substitution.

```
Algorithm Keyword_distance(kw1,kw2:wordtype):real;
begin
    dist[0,0]:=0;
    for i:=1 to length_kw2 do
        dist[i,0]:=i;
    for j:=1 to length_kw1 do
        dist[0,j]:=j;
    for i:=1 to length_kw2 do
    begin
      for j:=1 to length_kw1 do
      begin
          above:=dist[i-1,j]+weight(kw1[j],o);
          aboveleft:=
             dist[i-1,j-1]+weight(kw1[j],kw2[i]);
             left:=dist[i,j-1]+weight(o,kw2[i]);
            dist[i,j]:=min(left,aboveleft,above);
      end;
    end;
end;
```

Figure 3.2: Levenshtein algorithm in pseudo-code. The algorithm works dynamically, so that, for each p_1, p_2 prefixes of keyword1, keyword2, it determines the least cost of operations mapping p_1 to p_2. The version used here adds a step relativizing the distance measure to the keyword length (of the longer word).

3.4 Step 4 - Generating the Clusters

Levenshtein and Euclidean metrics have been used to generate the base clusters [11]**.** The proposed formula is $D_{ij}= \alpha\ d_{ij}+ \ \beta\ s_{ij}$, where d_{ij} is Euclidean metric and s_{ij} Levenshtein metric, α and β are constants, and AHC approach has been applied.

3.5 Step 5- Combining the Base Clusters

The base clusters obtained through step 1 to 4 are combined to higher level clusters. The average linkage method [12] is used to merge the base cluster into a higher level clusters and so on as shown in Figure 3.3. The proposed framework of indexing the Web Documents using Agglomerative Hierarchical Clustering is shown Figure 3.4.

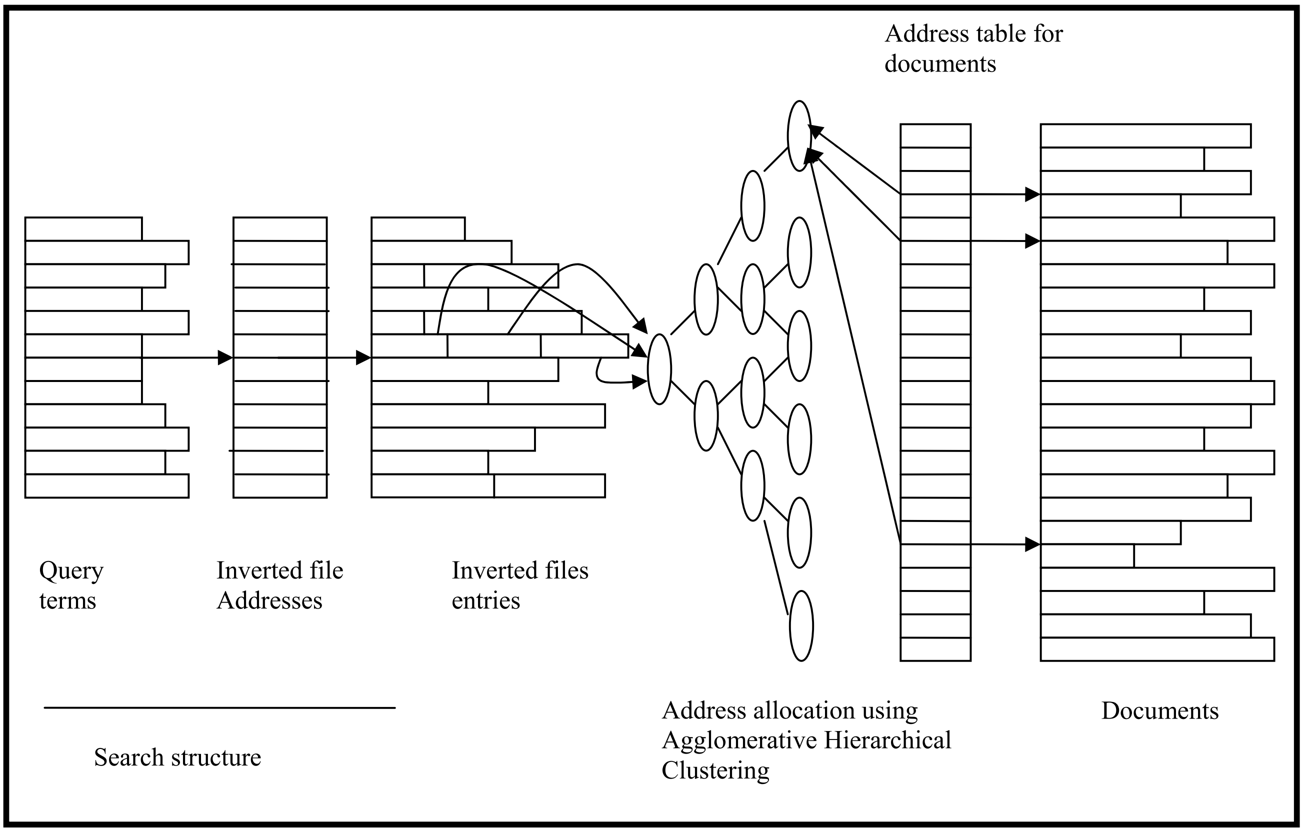

Address table for documents
Query terms
Inverted file Addresses
Inverted files entries
Search structure
Address allocation using Agglomerative Hierarchical Clustering
Documents

Figure 3.4: Proposed framework of indexing the Web Documents using Agglomerative Hierarchical Clustering

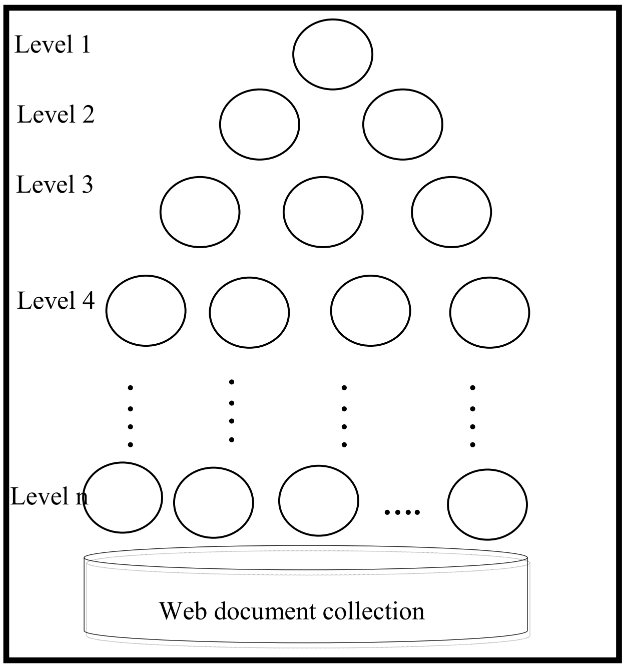


Figure 3.3: The Generalized Architecture

Here the distance between two clusters is defined as the average of distances between all the pairs of objects, where each pair is made up of one object from each group. In the average linkage method, ***A(r,s)*** is computed as ***A(r,s) = T_{rs} / (N_r * N_s),*** where ***T_{rs}*** is the sum of all pair-wise distances between cluster ***r*** and cluster ***s. N_r*** and ***N_s*** are the sizes of the clusters ***r*** and ***s*** respectively**.** At each stage of hierarchical clustering, the clusters ***r*** and ***s***, for which ***A(r,s)*** is the minimum, are merged.

Example

In this example there are 10 web documents and 6 keywords as shown in Figure 3.1. Each keyword and list of containing in each word is shown in Table 3.3. By applying step I to step IV, Calculate the distance (D_{ij}) between all the pairs of keywords. The keywords i and j for which D_{ij} is minimum, are merged then some group of keywords are made and this is called the base clusters as shown in Figure 3.5. Each base cluster is assigned a cluster ID as shown in Table 3.5.

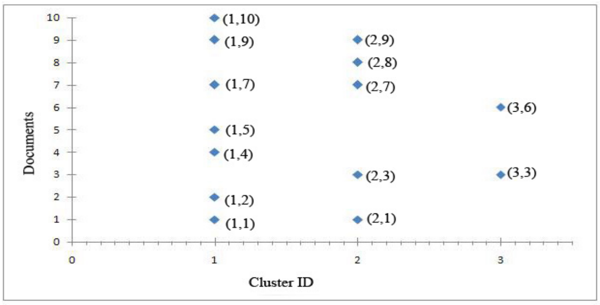


Figure 3.5:- Graphical Representation of Cluster ID and Documents

Table 3.4 Clusters and their respective cluster IDs

Clusters	Cluster ID
Crawler, Search Engine, Crawl, Database File	1
Security, Crawl	2
Database Management	3

Now applying step I to step V. Till step IV apply as done above. In step V, calculate the $A(r,s)=T_{rs}$ /(N_r * $N_{s)}$, where T_{rs} is the sum of all pair-wise distances between cluster r and cluster s. N_r and N_s are the sizes of the clusters r and s respectively. At each stage of hierarchical clustering, the clusters r and s, for which A(r,s) is the minimum, are merged.

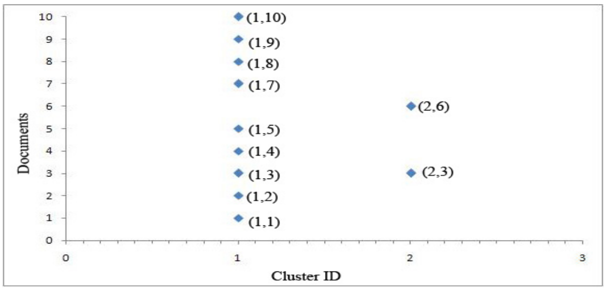


Figure 3.6:- Graphical Representation of Cluster ID and Documents

Table 3.5: Clusters and their respective cluster IDs

Clusters	Cluster ID
Crawler, Search Engine, Crawl, Database File, Security	1
Database Management	2

After applying step I to V, one cluster is made which has all documents. So that searching becomes more efficient.

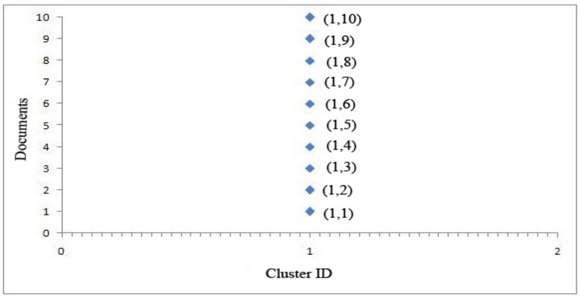


Figure 3.7:- Graphical Representation of Cluster ID and Documents

Table 3.6: Clusters and their respective cluster IDs

Cluster	Cluster ID
Crawler, Search Engine, Crawl, Database File, Security , Database Management	1

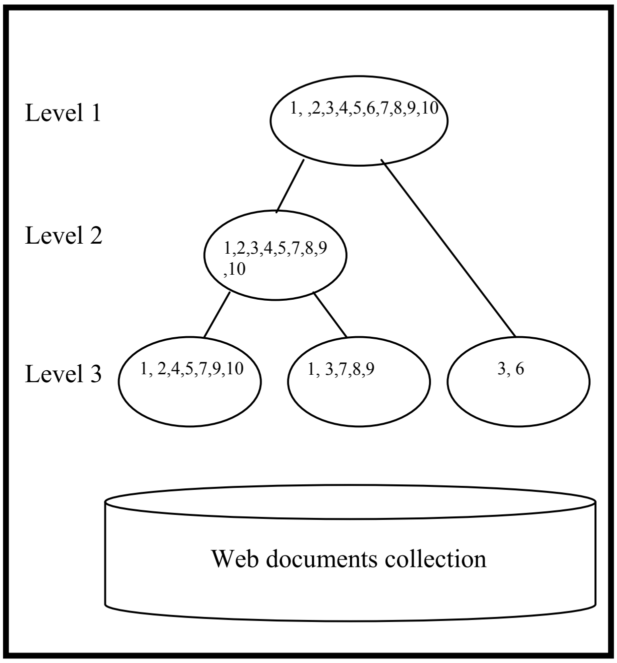
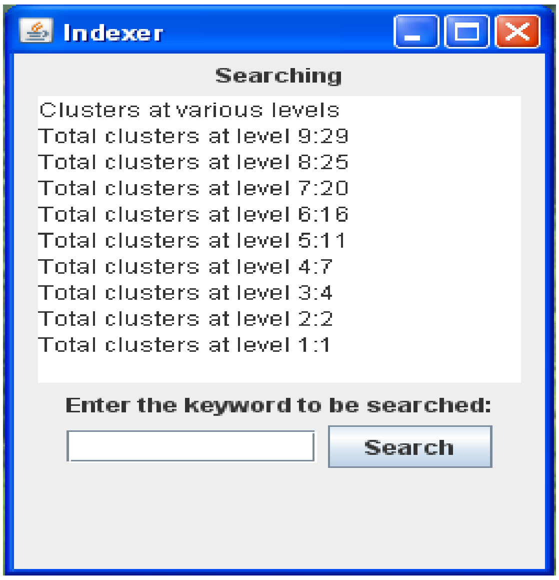


Figure 3.8 Web Document Clustering

4. Results, Snapshots and Analysis

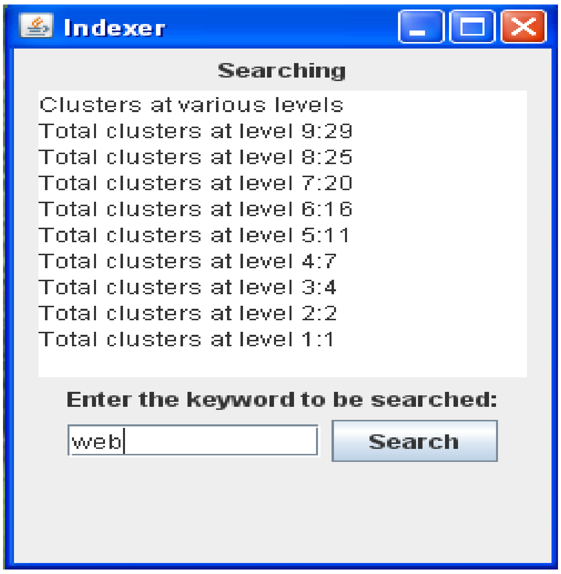
In this section the snapshot of Clustering of web document through keywords is shown Figure 3.9.Searching of a keyword is written in query interface and shown in Figure 3.10. Figure 3.11 is shown result and presence of keywords in which documents. In next Figure 3.12 clustering between five documents through keyword and results are shown in Figure 3.13.



```
Clusters at various levels
Total clusters at level 9:29
Total clusters at level 8:25
Total clusters at level 7:20
Total clusters at level 6:16
Total clusters at level 5:11
Total clusters at level 4:7
Total clusters at level 3:4
Total clusters at level 2:2
Total clusters at level 1:1
```

Enter the keyword to be searched:

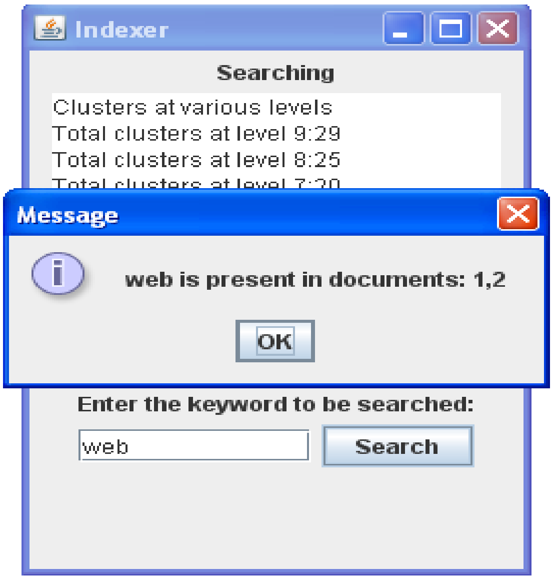
Figure 3.9: Snapshot of Indexer s/w



```
Clusters at various levels
Total clusters at level 9:29
Total clusters at level 8:25
Total clusters at level 7:20
Total clusters at level 6:16
Total clusters at level 5:11
Total clusters at level 4:7
Total clusters at level 3:4
Total clusters at level 2:2
Total clusters at level 1:1
```

Enter the keyword to be searched: web

Figure 3.10: Snapshot for searching another word from given no of documents

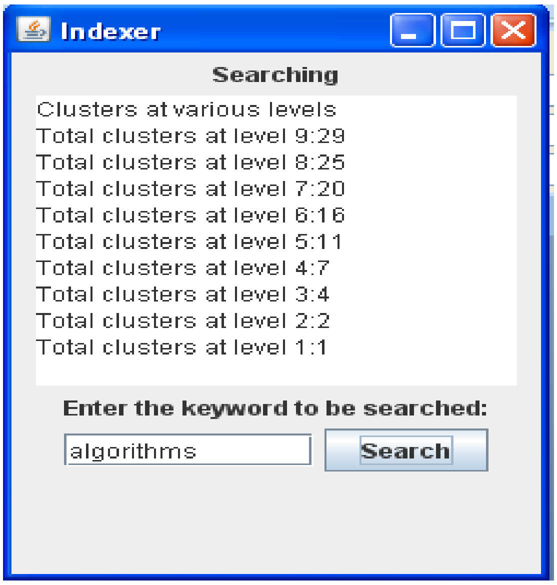


```
Clusters at various levels
Total clusters at level 9:29
Total clusters at level 8:25
Total clusters at level 7:20
```

web is present in documents: 1,2

Enter the keyword to be searched: web

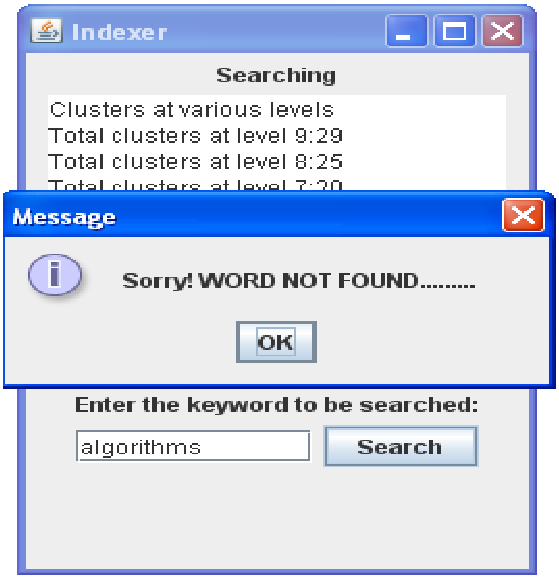
Figure 3.11: Snapshot of results



```
Clusters at various levels
Total clusters at level 9:29
Total clusters at level 8:25
Total clusters at level 7:20
Total clusters at level 6:16
Total clusters at level 5:11
Total clusters at level 4:7
Total clusters at level 3:4
Total clusters at level 2:2
Total clusters at level 1:1
```

Enter the keyword to be searched: algorithms

Figure 3.12: Snapshot for searching another word from given no of documents



Searching

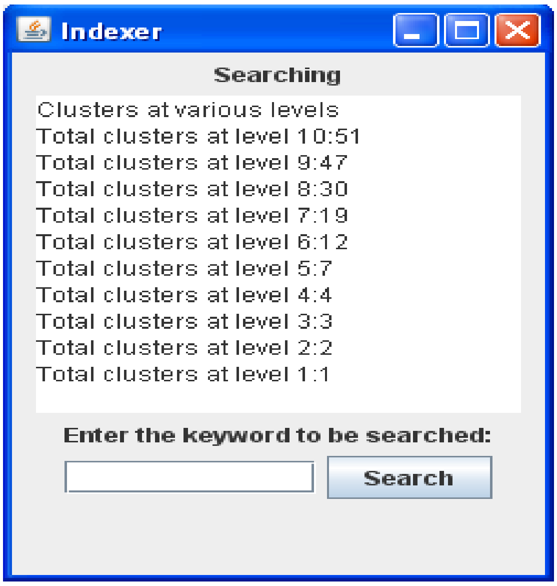
Clusters at various levels
Total clusters at level 9:29
Total clusters at level 8:25
Total clusters at level 7:20

Sorry! WORD NOT FOUND.........

Enter the keyword to be searched:

algorithms

Figure 3.13: Snapshot of results

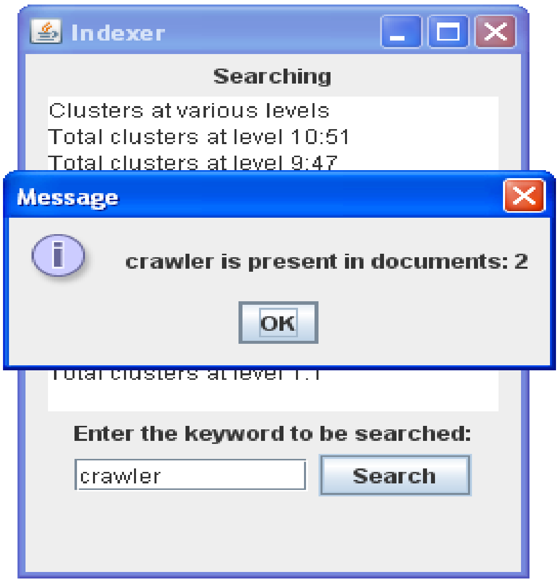


Searching

Clusters at various levels
Total clusters at level 10:51
Total clusters at level 9:47
Total clusters at level 8:30
Total clusters at level 7:19
Total clusters at level 6:12
Total clusters at level 5:7
Total clusters at level 4:4
Total clusters at level 3:3
Total clusters at level 2:2
Total clusters at level 1:1

Enter the keyword to be searched:

Figure 4.14: Snapshot of Indexer s/w



Searching

Clusters at various levels
Total clusters at level 10:51
Total clusters at level 9:47
Total clusters at level 1:1

crawler is present in documents: 2

Enter the keyword to be searched:

crawler

Figure 4.13: Snapshot of results

The complexity of this novel algorithm is nearly O(n). The algorithm can effectively find out the points in particular range from a given query point

5. Conclusion and Future research

A novel indexing technique for document retrieval on the web was proposed and a fuzzy logic based approach was used for indexing which provides relevant result in less time. This technique keeps the related documents in the same cluster so that searching of documents becomes more efficient in terms of time complexity.

In future work we can also develop index by using classification so that more efficient classification rules can be mined to make indexing more efficient and scalable.

References

[1] The Anatomy of a Large-Scale Hypertextual Web Search Engine Sergey Brin and Lawrence Page *Computer Science Department,*

[2] BERENDT, B. 2000. Web usage mining, site semantics, and the support of navigation. In Proceedings of the Workshop WEBKDD'2000 Web Mining for E-Commerce—Challenges and Opportunities, Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (Boston, August).

[3] P.Willet. Recent Trends In Hierarchical Document Clustering: A Critical Review. Information Processing And Management,24: 577-97,1988.

[4] D.R.Cutting, D.R.Karger, J.O.Pedersen And J.W.Tukey. Scatter/Gather: Cluster-Based Approach To Browsing Large Document Collections. In Proceedings Of The 15th International Acm Sigir Conference On Research And Development In Information Retrieval, Pages 318-29,1992.

[5] A.K. Jain, M.N.Murty And P.J.Flynn , Data Clustering: A Review, Acm Computing Surveys. 31(3):264-323, Sep 1999.

[6] Andrei Scherbina: Clustering Of Web Access Sessions, 2008

[7] D. Fasulo, "An analysis of recent work on clustering algorithms," Department of Computer Science and Engineering,University of Washington, Tech. Rep. # 01-03-02, 1999. [Online]. Available: <citeseer.nj.nec.com/fasulo99analysi.html> *Stanford University, Stanford, CA 94305, USA* sergey@cs.stanford.edu and page@cs.stanford.edu

[8] Stephen P. Borgatti: "How to explain hierarchical clustering" http://www.analytictech.com/networks/hiclus.htm

[9]Sung-Hyuk Cha "Comprehensive Survey on Distance/Similarity Measures between Probability Density Functions" Issue 4, Volume 1, 2007

[10] Phonetic Distance between Dutch Dialects John Nerbonne Wilbert Heeringa Erik van den Hout Peter van der Kooi Simone Otten Willem van de Vis Alfa-Informatica, BCN, Rijksuniversiteit Groningen nerbonne@let.rug.nl

[11] Osmar R. Zaïane: "Principles of Knowledge Discovery in Databases - Chapter 8: Data Clustering" http://www.cs.ualberta.ca/~zaiane/courses/cmput690/slides/Chapter8/index.html

[12] Andrew Moore: "K-means and Hierarchical Clustering – Tutorial Slides" http://www-2.cs.cmu.edu/~awm/tutorials/kmeans.html

Deepti Gupta received the M.Sc degree in Mathematics with Hons. From C.C.S University, Campus in the year 2006. Presently, she is working as Lecturer in School of Computer Science and Information Technology in Shobhit University, Meerut. She is also pursuing M.tech in Computer Engineering and her areas of interests are Search Engines, Crawlers and Data Mining.

Dr. Komal Kumar Bhatia received the B.E, M.Tech. and Ph.D. degrees in Computer Science Engineering with Hons. from Maharishi Dayanand University in 2001 2004 and 2009, respectively. Presently, he is working as Assistant Professor in Computer Engineering department in YMCA Institute of Engineering, Faridabad. He is also guiding Ph.Ds in Computer Engineering and his areas of interests are Search Engines, Crawlers and Hidden Web.

Prof. A. K. Sharma received his M.Tech. (Computer Sci. & Tech) with Hons. From University of Roorkee in the year 1989 and Ph.D (Fuzzy Expert Systems) from JMI, New Delhi in the year 2000. From July 1992 to April 2002, he served as Assistant Professor and became Professor in Computer Engg. at YMCA Institute of Engineering Faridabad in April 2002. He obtained his second Ph.D. in IT from IIIT & M, Gwalior in the year 2004. His research Interest include Fuzzy Systems, Object Oriented Programming, Knowledge Representation and Internet Technologies.