DOI: 10.21917/ijsc.2012.0067

IMPROVING PERSONALIZED WEB SEARCH USING BOOKSHELF DATA STRUCTURE

S.K. Jayanthi[1]and S. Prema[2]
[1]Department of Computer Science, Vellalar College for Women, India
[2]Department of Computer Science, K.S.R College of Arts and Science, India
E-mail: prema_shanmuga@yahoo.com

Abstract
Search engines are playing a vital role in retrieving relevant information for the web user. In this research work a user profile based web search is proposed. So the web user from different domain may receive different set of results. The main challenging work is to provide relevant results at the right level of reading difficulty. Estimating user expertise and re-ranking the results are the main aspects of this paper. The retrieved results are arranged in Bookshelf Data Structure for easy access. Better presentation of search results hence increases the usability of web search engines significantly in visual mode.

Keywords:
Web Search Personalization, Bookshelf Data Structure, Agglomerative Hierarchical Clustering, Similarity Measure, Visualization

1. INTRODUCTION

The web users differ widely in their reading ability and capability to understand the meaning, depending on factors such as age, Technical background, and area of specialization. For example, an interior designer may use the query “windows” to find information about models of the windows, while a software engineer may use the same query to find details about windows operating systems. The retrieved results are arranged in Bookshelf Data Structure for easy access. Finally the results are displayed in visual mode. Web search engines, however, typically use optimized algorithms for the average user, and not for the specific individuals [1].In the current scenario, web page results personalization is playing a vital role. The entire Meta search engines are competing with each other to provide the web user with the relevant and efficient content in response to his or her query. The web users expect the best results in the first page itself. They are not having the patient to browse longer in URL mode.

This research is based on the concept based analysis, of the document. The concept-based model analyzes the semantic terms in the document. The proposed system can effectively differentiate the non important terms and semantic terms which hold the concepts that represent the sentence meaning using lexical analyzer. The proposed mining architecture is to measure the similarity between the documents. Clustering of Web search results using bookshelf data structure based on the user profile is the main focus of the paper. It helps the users to navigate quickly to the category they are actually interested and subsequently to the specific web page in visual mode. Experimental results show that the performance of the clustering accuracy is improved.

In summary, this architecture returns much higher quality results yet with similar response time to other systems. The rest of the paper is organized as follows: related work is discussed in Section 2 and provides some preliminaries on Web search results personalization. Section 3 is focused on the issues in the existing work. Section 4 presents the proposed system by describing the Semantic Web Architecture, Information Retrieval, TFIDF calculation, Sentence based concept analysis, Document Clustering, and Session Tracking. Section 5 explains the Bookshelf Data Structure. Section 6 reports the results of the experimental study. Finally, section 7 gives the concluding remarks.

2. RELATED WORK

Web search result personalization based on the user profile touches on several research areas with relevant prior work. Analysing user query based on their domain and producing the relevant information is the main focus on this research work. Effective search systems for children and students have been the focus of increased interest in recent years. Progress in improved user interfaces, crawling and indexing strategies, and models of child-centered relevance are all important in creating a better search experience for children [1]. A common personalization approach involves re-ranking the top *N* search results such that documents likely to be preferred by the user are presented higher [2].Personalizing search results for individual users is increasingly being recognized as an important future direction for web search [3, 4, 5].Parallelizing the web search results incorporated with Bookshelf Data Structure owe to the simplistic applicability of the above mentioned strategy, and the use of simple data structure [6].

3. EXISTING WORK

In existing work, the web search results are produced in URL mode. When the user enters the query then the document similarity is calculated based only on the keyword. If the same keywords are repeated more number of times then the term frequency calculation will produce the fault rate. So measuring the similarity of the documents based only on the keywords will produce the less efficiency in clustering.

4. PROPOSED ARCHITECTURE

The proposed work on concept based clustering captures the semantic structure of each term rather than the frequency of the term within a document. Each sentence is labeled by a semantic tool tip which contributes to the sentence semantics. Concepts can be either words or phrases and are totally dependent on the semantic structure of the sentence. When a new document is found, the concept based model enables a concept match between the new document and the previously processed documents. From the database the similar concepts are extracted and stored. Similarity based on matching of concepts between document pairs, is shown to have a more significant effect on the clustering quality as in Fig.1. The proposed work improves the clustering efficiency and displays the related documents of the search engine at less time.

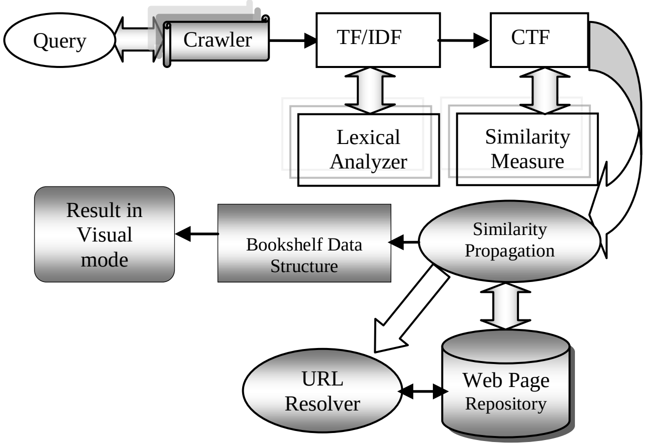


Fig.1. Proposed Architecture

4.1 TERM FREQUENCY/ INVERSE DOCUMENT FREQUENCY

TF/IDF can be calculated for each word using the four values such as number of words in a document, frequency of a word in a document, the number of total documents, and the number of documents where the word appears. Instead of extracting words from an e-text, two-word phrases were extracted and TFIDF is calculated for each of them.

Term frequency (TF) is essentially a percentage denoting the number of times a word appears in a document [10]. It is mathematically expressed as C/T, where C is the number of times a word appears in a document and T is the total number of words in the same document. Inverse document frequency (IDF) takes into account that many words occur many times in many documents. IDF is mathematically expressed as D/DF, where D is the total number of documents in a corpus and DF is the number of document in which a particular word is found. As D/DF increases so do the significance of the given word. TFIDF is calculated as given below,

$$TF/IDF = (C/T) * (D/DF) \tag{1}$$

4.2 SENTENCE BASED CONCEPT ANALYSIS

Conceptual Term Frequency (CTF) is an important factor in calculating the concept-based similarity measure between documents. The number of occurrences on a concept *c* in the verb argument structures of the sentence, *s* is called the CTF as given below,

$$ctf = \sum_{m=1}^{n} ctf_m / s \tag{2}$$

where, *s* is the total number of sentences that contain concept *c* in document *d*. The frequency of a concept is used to analyze the domain of the document. Taking the average of the *ctf* values of concept *c* in its sentences of document *d* measures the overall importance of concept *c* to the meaning of its sentences in document *d*. Thus, calculating the average of the *ctf* values measures the overall importance of each concept to the semantics of a document through the sentences as in Fig.2.

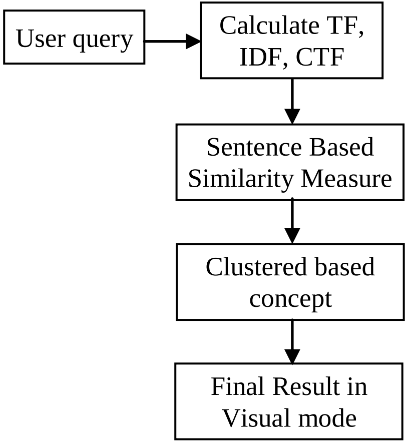

User query
Calculate TF, IDF, CTF
Sentence Based Similarity Measure
Clustered based concept
Final Result in Visual mode

Fig.2. Concept Analysis

4.3 SIMILARITY MEASURE CALCULATION

The documents in the database are assigned with unique identification number. The keywords in the documents are assigned with unique identifier. Using stemming process, the stop words are removed. Based on the concept, the number of occurrences of keywords is measured. Based on the counting of keywords in the documents, its similarity is measured. In turn it is compared with the other documents for finding the ratio of similarity and if so they are grouped together as shown in Table.1. The sample code written in Java is given below,

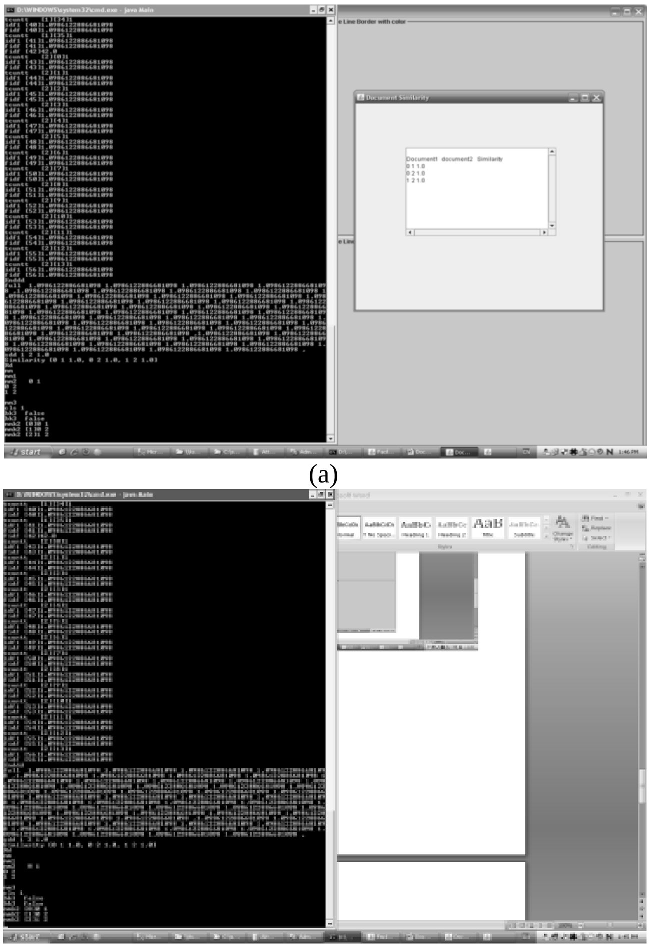
```
try {
bhhh = new BufferedWriter(new FileWriter("Sim1"));
System.out.println("Start");
n = nm.length;
file = new String[nm.length];
System.out.println("Size"+file.length);
file = nm;
ta.append("\n"+"Document1"+"            "+"document2"+"
"+"Similarity"+"\n");
ph = phase;   tcuntt=count;
for(int n3=0;n3<file.length;n3++){
System.out.println("file"+file[n3]);
sb.append(file[n3]+"\n");
System.out.println("fin"+file[n3]);
in1 = new FileInputStream(file[n3]);
bs1 = new Buffered Reader (new InputStreamReader (in1));
while ((line1=bs1.readLine( )) !=null)}
```

Table.1. Similarity Measure Calculation

Document1	Document 2	Similarity
0	1	1.0
0	2	1.0
0	3	1.0000000000000002
0	4	1.0
0	5	1.0
0	6	1.0
1	2	1.0

1	3	1.0000000000000002
1	4	1.0000000000000002
1	5	1.0
1	6	1.0000000000000002
2	3	1.0000000000000002
2	4	1.0000000000000002
2	5	1.0
2	6	1.0000000000000002
3	4	1.0
3	5	1.0000000000000002
3	6	1.0
4	5	1.0
5	6	1.0

The sample screenshots of the similarity measure between documents is given in Fig.3(a) and 3(b). The chart comparison for measuring similarity measure is given in Fig.4.



```
Document1  document2  Similarity
0 1 1.0
0 2 1.0
1 2 1.0
```

(a)

(b)

Fig.3. Executed results of Similarity Measure

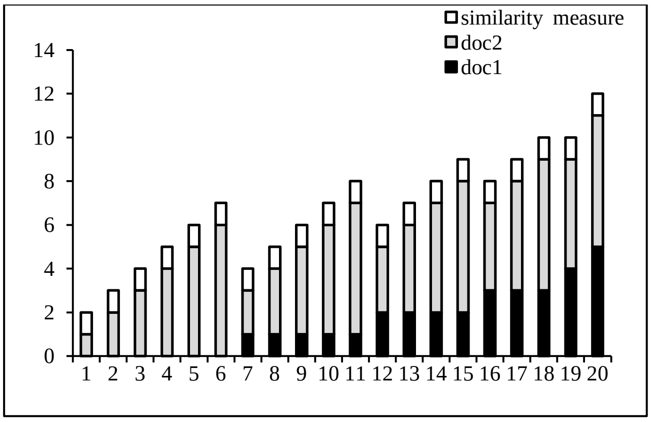


Fig.4. Chart comparison for similarity measure

4.4 CONCEPT BASED DOCUMENT CLUSTERING

Clustering is performed based on the result of the concept based similarity measure. Hierarchical clustering algorithms are either top-down or bottom-up. Agglomerative clustering techniques, also known as bottom-up approach, starts with as many clusters as there are objects, with each cluster having only one record. Then the pairs of clusters are successively merged until the number of clusters reduces to *k*. At each stage, the pairs of the clusters that are merged are the ones nearest to each other. The working procedures are given below [8],

Step 1: Initially each item $x_1, \ldots, x_n$ is in its own cluster *C*1, . ,*Cn*.

Step 2: Repeat until there is only one cluster left:

Step 3: Merge the nearest clusters, say *Ci* and *Cj*. Let d_{ij} = distance between item *i* and item *j*.

Step 4: Search the distance matrix for the nearest pair clusters. Denote the distance between these most similar clusters *U* and *V* by d_{UV}.

Step 5: The maximum distance between elements of each cluster is presented by Max $\{d(x,y) : x \varepsilon A, y \varepsilon B\}$

Step 6: The minimum distance between elements of each cluster is given by Min $\{d(x,y): x\varepsilon A, y\varepsilon B\}$

Step 7: The mean distance between elements of each cluster is given by, $1/|A|.|B| \sum_{x\varepsilon A} \sum_{x\varepsilon B} d(x,y)$

Step 8: Merge clusters *U* and *V* into a new cluster, labeled *T*.

Step 9: Repeat steps 2 and 3 a total of *N* – 1 times.

4.5 SESSION TRACKING

Based on the profile the users may view the results based on `basic', `intermediate', or `advanced' reading level. This approach has as its advantages both simplicity of user interaction and transparency of search behavior. The purpose of session identification is to allow a web application to identify related incoming requests as such.

A common way of session tracking is the use of cookies. A cookie is information that's stored as a name/value pair and transmitted from the server to the browser. Cookies containing unique user information can be used to tie specific visitors to information about them on the server [11]. The Java Servlet specification provides a simple cookie API that allows you to write and retrieve cookies. The following code shows how to create a new cookie,

```
Cookie user = new Cookie ("user","Prema");
user.setMaxAge(1500);
response.addCookie(user);
```

This code creates a cookie with a name of "user" and a value of "Prema". The cookie's expiration date is set with the setMaxAge( ) method to 1,500 seconds from the time the browser receives the cookie. The following code demonstrates how you would retrieve the value for a specific cookie,

```
String user = " ";
Cookie[ ] cookies = request.getCookies( );
if (cookies != null)
    {
    for (int i = 0; i < cookies.length; i++)
        {
        if (cookies[i].getName().equals("user"))
        user = cookies[i].getValue( );
        }
    }
```

4.6 SNIPPNET GENERATION

A result snippet should be self-contained so that the users can understand it [7]. Snippet helps the user to understand the core concept of the web page. An automated tooltip is designed so that if the user is placing a cursor on the document a window with self contained is displayed.

5. BOOKSHELF DATA STRUCTURE

Bookshelf data structure [9] as in Fig.5 has been introduced for community information which stores the inverse indices of the WebPages. Structures define the group of contiguous fields, such as records or control blocks. A structure is a collection of variables grouped together under a single name. It provides an elegant and powerful way for keeping related data together.This data structure is formed by combining a matrix and list with dynamically allocated memory. This is an extended data structure of hash table and bi-partite core [9], which is used to store base domain and sub-domain indices of various communities. The information retrieved from the database is stored in the each shelf separately for easy retrieval. Users are not satisfied if they didn't get results in the first page itself. So a modified vision based approach is proposed in this paper to present the web search results in visual mode.

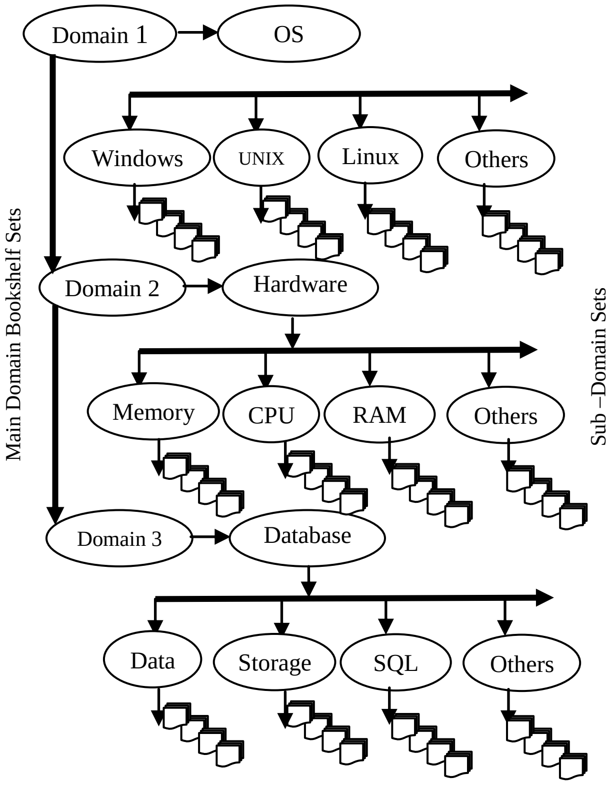


Fig.5. Bookshelf Data structure

6. RESULTS AND DISCUSSION

In the proposed work web search result personalization is focused. From the Table.2 it is clear that the user can get the efficient results based on their domain. Concept based Browsing provides increased proficiency as in Fig.6.The databases are stored in SQL server as in Fig.7. Through ODBC (Open Database Connectivity) the data as in Fig.8 are linked to the Source code written in Java.

```
Nmain.ta.append("\n Links are"+"\n");
db = new database_conn( );
db.st.executeUpdate("if object_id('Links') is not null delete from Links");
String str = ard;
URL url = new URL(searchstr)
URLConnection conn =  url.openConnection( );
conn.setRequestProperty("User-Agent","Mozilla/6.0);
BufferedReader in = new BufferedReader(new InputStreamReader(conn.getInputStream()));
BufferedWriter bwd = new BufferedWriter(new FileWriter("web"));
while ((str1=in.readLine( )) != null)
{
html+ = str1+"\n";
bwd.write(str1.trim( )+"\n");
}
in.close( );
bwd.close( );
```

Table.2. User Profile Based Browsing

Domain	Number of queries	Response efficiency	
		Common Login (%)	User Profile based Login (%)
Science	50	52	56
Education	40	68	72
Research	42	45	62
Software	38	67	70
Total	170	232	260

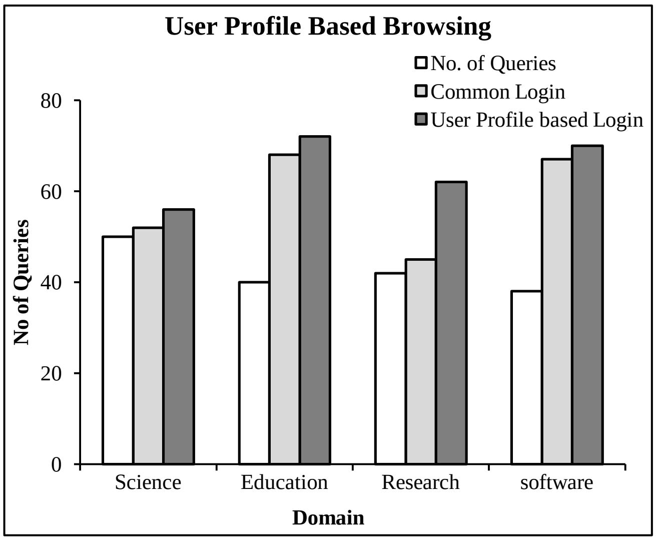
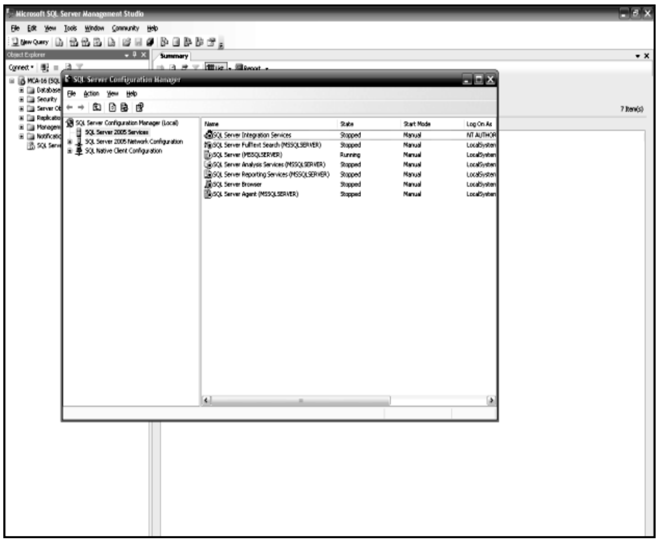
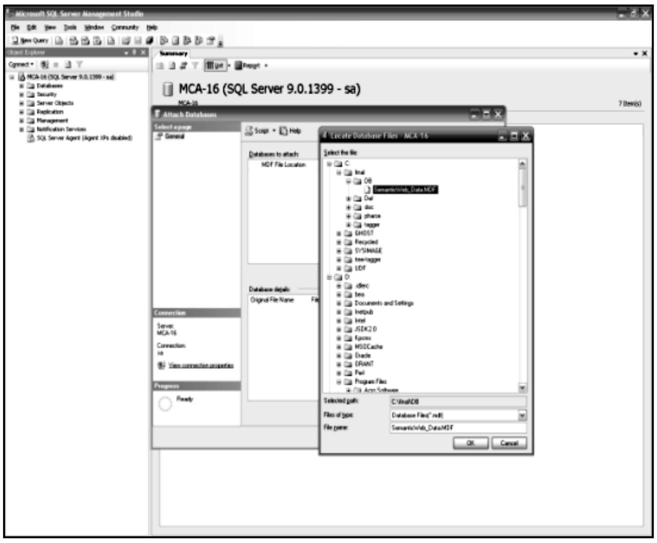


Fig.6. User Profile Based Browsing


Microsoft SQL Server Management Studio
File Edit View Tools Window Community Help
New Query
Object Explorer
Summary
SQL Server Configuration Manager
File Action View Help
SQL Server Configuration Manager (Local)
SQL Server 2005 Services
SQL Server 2005 Network Configuration
SQL Native Client Configuration

Name	State	Start Mode	Log On As
SQL Server Integration Services	Stopped	Manual	NT AUTHOR
SQL Server FullText Search (MSSQLSERVER)	Stopped	Manual	LocalSystem
SQL Server (MSSQLSERVER)	Running	Manual	LocalSystem
SQL Server Analysis Services (MSSQLSERVER)	Stopped	Manual	LocalSystem
SQL Server Reporting Services (MSSQLSERVER)	Stopped	Manual	LocalSystem
SQL Server Browser	Stopped	Manual	LocalSystem
SQL Server Agent (MSSQLSERVER)	Stopped	Manual	LocalSystem

Fig.7.Database



MCA-16 (SQL Server 9.0.1399 - sa)

Server: MCA-16

Selected path: C:\final\DB

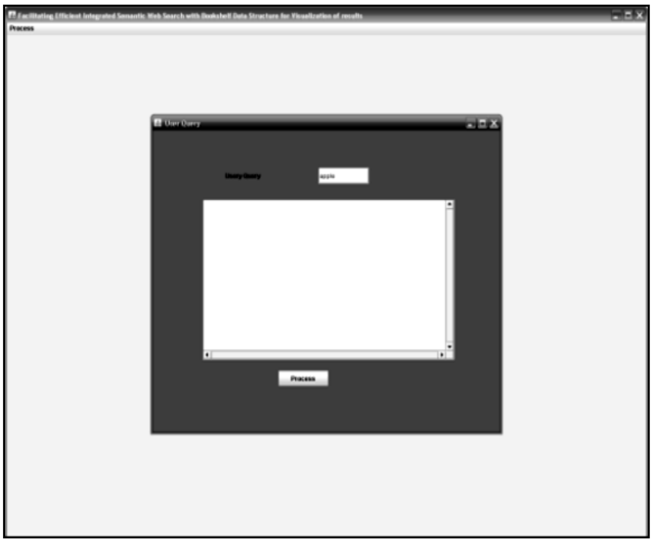
Files of type: Database Files(*.mdf)

File name: SemanticWeb_Data.MDF

Fig.8. Database Connectivity

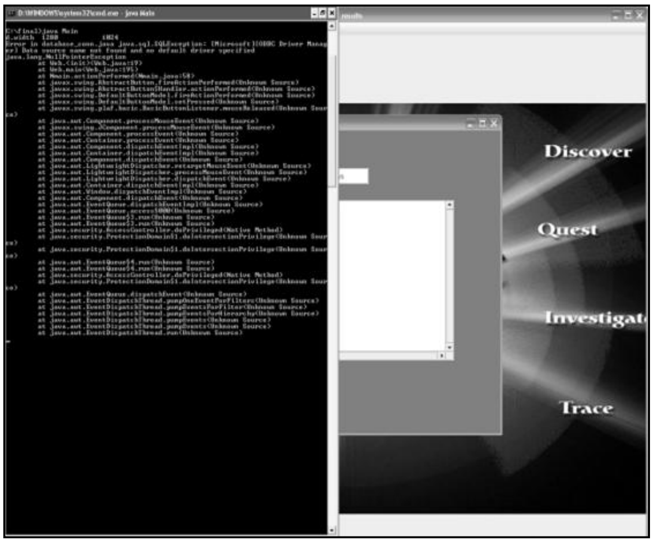
Based on the user query the related links are displayed. Normally the query is in the form of keywords as in Fig.9. The keywords which match with the documents are indexed as in Fig.10. The lexical meaning of the keyword is analyzed through Word Net. Tree Tagger is for annotating text with partofspeech and lemma information .Tokenization is the process of breaking a stream of text up into words phrases, symbols or other meaningful elements called tokens. The list of tokens becomes input for further processing such as parsing.

The related documents are captured and arranged in the folder. The similarity between the documents is calculated. The documents with more similarity measures are clustered and arranged in each shelf of BookShelf for easy access of information in visual mode.


Facilitating Efficient Integrated Semantic Web Search with Bookshelf Data Structure for Visualization of results
Process
User Query
Process

Usery Query: apple

Fig.9. Home Page



```
E:\final>java Main
d.width  1280        1024
Error in database_conn.java java.sql.SQLException: [Microsoft][ODBC Driver Manag
er] Data source name not found and no default driver specified
java.lang.NullPointerException
        at Web.<init>(Web.java:19)
        at Web.main(Web.java:195)
        at Mmain.actionPerformed(Mmain.java:58)
        at javax.swing.AbstractButton.fireActionPerformed(Unknown Source)
        at javax.swing.AbstractButton$Handler.actionPerformed(Unknown Source)
        at javax.swing.DefaultButtonModel.fireActionPerformed(Unknown Source)
        at javax.swing.DefaultButtonModel.setPressed(Unknown Source)
        at javax.swing.plaf.basic.BasicButtonListener.mouseReleased(Unknown Sour
ce)
        at java.awt.Component.processMouseEvent(Unknown Source)
        at javax.swing.JComponent.processMouseEvent(Unknown Source)
        at java.awt.Component.processEvent(Unknown Source)
        at java.awt.Container.processEvent(Unknown Source)
        at java.awt.Component.dispatchEventImpl(Unknown Source)
        at java.awt.Container.dispatchEventImpl(Unknown Source)
        at java.awt.Component.dispatchEvent(Unknown Source)
        at java.awt.LightweightDispatcher.retargetMouseEvent(Unknown Source)
        at java.awt.LightweightDispatcher.processMouseEvent(Unknown Source)
        at java.awt.LightweightDispatcher.dispatchEvent(Unknown Source)
        at java.awt.Container.dispatchEventImpl(Unknown Source)
        at java.awt.Window.dispatchEventImpl(Unknown Source)
        at java.awt.Component.dispatchEvent(Unknown Source)
        at java.awt.EventQueue.dispatchEventImpl(Unknown Source)
        at java.awt.EventQueue.access$000(Unknown Source)
        at java.awt.EventQueue$3.run(Unknown Source)
        at java.awt.EventQueue$3.run(Unknown Source)
        at java.security.AccessController.doPrivileged(Native Method)
        at java.security.ProtectionDomain$1.doIntersectionPrivilege(Unknown Sour
ce)
        at java.security.ProtectionDomain$1.doIntersectionPrivilege(Unknown Sour
ce)
        at java.awt.EventQueue$4.run(Unknown Source)
        at java.awt.EventQueue$4.run(Unknown Source)
        at java.security.AccessController.doPrivileged(Native Method)
        at java.security.ProtectionDomain$1.doIntersectionPrivilege(Unknown Sour
ce)
        at java.awt.EventQueue.dispatchEvent(Unknown Source)
        at java.awt.EventDispatchThread.pumpOneEventForFilters(Unknown Source)
        at java.awt.EventDispatchThread.pumpEventsForFilter(Unknown Source)
        at java.awt.EventDispatchThread.pumpEventsForHierarchy(Unknown Source)
        at java.awt.EventDispatchThread.pumpEvents(Unknown Source)
        at java.awt.EventDispatchThread.pumpEvents(Unknown Source)
        at java.awt.EventDispatchThread.run(Unknown Source)
```

Fig.10. Indexing of Documents

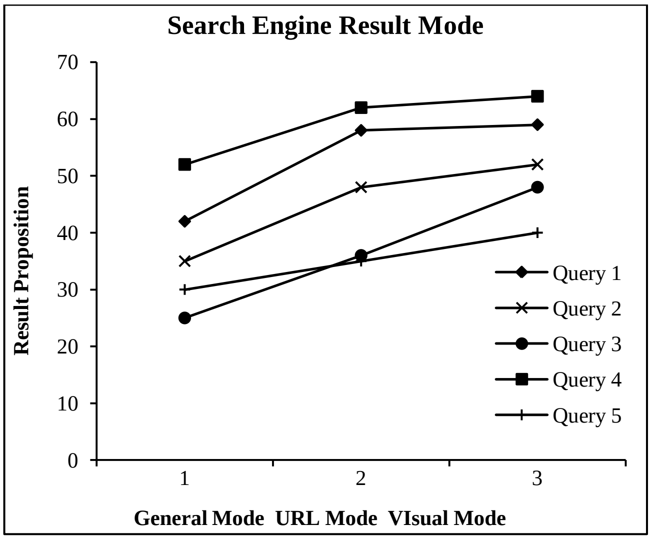


Fig.11. Result Mode

Web users normally prefer the web search results in visual mode as in Fig.11.

7. CONCLUSION

Clustering is a well-organized way of grouping relevant information from raw data. In the existing work the web search results are displayed in URL mode. In this research work web search result personalization and presenting the report in visual mode is proposed. The architecture is composed of TF-IDF analysis between the documents, sentence based concept analysis and document clustering based on similarity measure. Displaying the link based on user query provides easy access to the user. This semantic web search engine efficiently minimizes the time and cost of the user by uniquely identifying the service objects. The Snippet generation minimizes the browsing time of the user. Finally the clustered documents are arranged in Bookshelf Data Structure for easy retrieval and the results are displayed in visual mode.

REFERENCES

[1] Kevyn Collins-Thompson, Paul N. Bennett, Ryen W. White, Sebastian de la Chica and David Sontag, “Personalizing Web Search Results by Reading Level”, *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, pp. 403-412, 2011.

[2] Filip Radlinski and Susan Dumais, “Improving Personalized Web Search using Result Diversification”, *Proceedings of the 20th annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 691-692, 2006.

[3] Kazunari Sugiyama, Kenji Hatano and Masatoshi Yoshikawa, “Adaptive web search based on user profile constructed without any effort from users”, *Proceedings of the 13th International Conference on World Wide Web*, pp. 675-684, 2004.

[4] Jaime Teevan, Susan T. Dumais and Eric Horvitz, “Beyond the commons: Investigating the value of personalizing web search”, *Proceedings of Workshop on New Technologies for Personalized Information Access*, pp. 84–92, 2005.

[5] Yi Zhang, Jamie Callan and Thomas Minka, “Novelty and redundancy detection in adaptive filtering”, *Proceedings of the 25th annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 81–88, 2002.

[6] S. Prema and S.K. Jayanthi, “Web Search Results Visualization Using Enhanced Branch and Bound Bookshelf Tree Incorporated with B3-Vis Technique”, *International Journal of Machine Learning and Computing*, Vol. 2, No. 5, pp. 644-647, 2012.

[7] Yu Huang, Ziyang Liu and Yi Chen, “Query Biased Snippet Generation in XML Search”, *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp. 315-326, 2008.

[8] David M. Blei, “Hierarchical clustering”, Lecture Notes, Princeton University, 2008.

[9] S.K. Jayanthi and S. Prema, “CIMG-BSDS: Image Clustering Based on Bookshelf Data Structure in Web Search Engine Visualization”, *Communications in Computer and Information Science*, Vol. 269, pp. 457-466, 2012.

[10] http://en.wikipedia.org/wiki/Tf%E2%80%93idf.

[11] en.wikipedia.org/wiki/HTTP_cookie.