Design of a User Controlled Parser based Desktop Search Engine

Nitin Sharma
Cognizant Technology Solutions, Pune, India

Prerika Agarwal
Department of Computer Science
Ajay Kumar Garg Engineering College, Ghaziabad, India

Abstract:
The user's computer is contains millions of electronic collections that often contains high quality information. The user's disk also includes the removable disks and external drives as well which can have data of more than 1TGB. To deal with this explosion of the digital data in size and complexity many imperatives are to be faced. Due to the massive growth of the disks, search engines have emerged as a fast and efficient tool to find relevant information from the user's computer. These search engines are huge and retrieve a lot of information. They also contain some filters and ranking algorithm that results output in the best possible form. Disks can be of type single user as well as multiple users in which multiple users can save their information as in case of a large organization. Disks are very dynamic in nature since documents are updated every minute of the day, as users are saving more documents and uploading their work. Moreover, a user is allowed to interact with the search engine through the given interface for query purpose only. He is not allowed to update the Desktop Search Engine side database. In this paper, the user is provided an authority to update the Desktop Search Engine database. It parses the text, HTML pages and document files by using a basic parser. The ranking mechanism also has been changed and is based on the occurrence of each word on the document and then indexes it accordingly.

Keywords: Desktop Search engine, Parser, User controlled, Page Rank

1. INTRODUCTION

Desktop search is the name for the field of search tools which search the contents of a user's own computer files, rather than searching the Internet. These tools are designed to find information on the user's PC, including web browser histories, e-mail archives, text documents, sound files, images and video. One of the main advantages of desktop search programs is that search results arrive in a few seconds.

Desktop search is emerging as a concern for large firms for two main reasons: untapped productivity and security. A commonly cited statistic states that 80% of a company's data is locked up inside unstructured data the information stored on an end user's PC, the files and directories they've created on a network, documents stored in repositories such as corporate intranets and a multitude of other locations.

Desktop search engine for text files is a desktop search technique. All the information can be easily accessed by using the technique and specific search results are given in a time of one or two seconds.

The system provides users to manually scan the hard disk for text files, docx files, pdf files, html files and to parse text files for keywords. User can search for files by their name and also with the contents of text files.

Since the size of user's disk is increasing in a tremendous way it has increased from 20GB in 1999 to 1TGB in 2011. To search any relevant information over the hard disk is always a challenging task. To search through hard disk to

get the most specific information desktop search engines are needed. It also helps them to save their time and energy when they want to search files on user's hard disk.

The digital information that is created, captured or replicated in digital form was 1 GB in 1992 [9]. There has beenunbelievable growth in the number of documents and it is estimated that in 2011, the amount of digital information produced in the year should nearly equal 1000GB, or ten times that produced in 2000 (Fig. 1). The compound annual growth rate between now and 2011 is expected to be almost 60% [9]. To deal with this explosion of the digital universe in size and complexity many imperatives are to be faced.

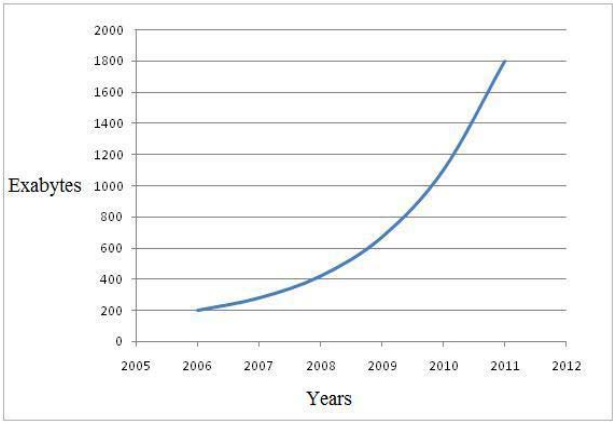


Figure1. Growth of Digital Information

Due to the massive growth of the disks, applications called "Desktop search engines" have emerged as a fast and efficient tool to find relevant information from the desktop [2, 5]. Information on almost any imaginable subject can be rapidly found using search engines. And as information and knowledge continues to grow, search engines will become increasingly important to our daily life.

2. RELATED WORK

Search engines [7] operate as a link between users and documents in our desktop (Fig. 2). Without desktop search no information can be retrieved on time or when it is needed.Since the size of the disks is increasing and to remember the path of each and every file into the systemis not an easy task. Desktop Search searches the database for the desired keyword, ranks it according to the similar content and then returns the required information with the best possible solution.

Different types of search engines available are Crawler based search engines, Human powered directories, Meta search engines, and Hybrid search engines.Crawler based search engines create their listings automatically with the help of web crawlers. It uses a computer algorithm to rank [13] all pages retrieved.These search engines are huge and retrieve a lot of information. It also contains some filters and ranking algorithm rank results in the best possible form.

Human powered directories are built by human selection i.e. they depend on humans to create the repository. These directories are organized into subjects, and pages are grouped under subjects.Meta search engines accumulate search and screen the results of multiple primary search engines. Meta engines are relatively slow engines and do not crawl the web by listings. The results are sent to several search engines and blend their results into one page.Hybrid search engines combine the spider search engines and dictionaries. These search engines favour one type of listings over another. In the Desktop search the crawler based search engine is modified to search the desktop.

Three main parts of a parser based search engineareParser*, Indexer and Searche*r [12]. The parser follows documents across the hard disk collecting information from different documents. Starting from a basic drive on the desktop, and recursively follow all the files and folders to other documents. This makes it possible to reach most of the hard drive in a relatively short time.

The indexer takes the web pages collected by the parser and parses them into a highly efficient index. In the index, the terms are given an importance weighting by the search engine's ranking [13] algorithm. The searcher (query

engine or retrieval engine) returns results of a query to the user. Because of these different term weighting and document selection methods, bias is introduced in the search engines. Different search engines also have different ranking algorithms and apply run-time filters to their results.

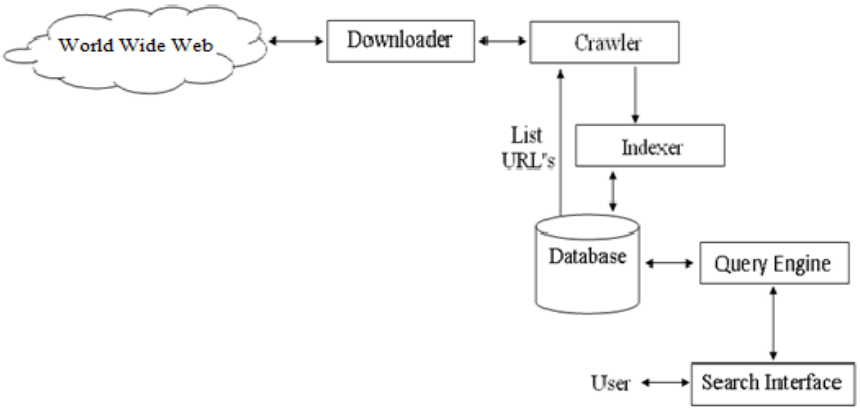


Figure 2. Architecture of a typical web search engine

3. PARSING PROCESS

Parsing (Fig. 3) is the process by which desktop search engine discovers new or updated documents; to be added to the database. The database created; is used to search all the queries related to the search process.

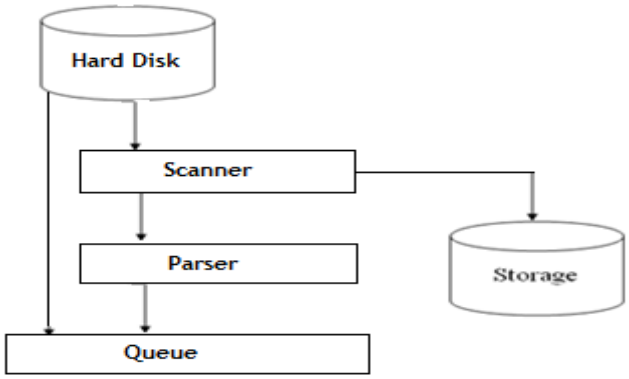


Figure 3. General architecture of a web Parser

Various problems related to the most famous desktop search engines are that many a times the results shown are not up-to-date or not proper. User is not allowed to update the Search Engines database, and he can interact with the search engine through the given interface for query purpose only. The results shown are biased or improper i.e. first few results shown do not solve our purpose. Sometimes the file which exists on the desktop is still not shown in the results as the database and indexing is not proper.

In this paper, the process of crawling is implemented using a new ranking mechanism.The user is provided an authority to update the Search Engine side database. He is allowed to submit the computer's drive which he wishes to scan. It parses the Text, html and word (.docx) files by using a basic parser. The ranking mechanism also has been changed and is based on the occurrence of each word on the document and then indexes it accordingly.

4. PROPOSED WORK

In the proposed desktop search engine the crawler is given a hard disk of the user from which it can extract documents one by one,parse and analyze these pages. On the other hand it provides a handle to the user to add a drive for extraction. It also extracts all the embedded URLs found in the page and add them to the list of URLs to be extracted. It uses in-place updation to maintain freshness of the database. A good indexing technique is used for searching the database with minimum possible time.

The main working of proposed search engine is shown in Figure 4 using a '0 level' data flow diagram.The user submits query to the search engine and it searches for that query in the database of the crawler, and displays the result.The working of User operations and Admin operations is shown in Figure 5 using '1 level' data flow diagram. In Admin Operation it shows scanning hard disk for files, parsing the text files from the drive and updating database. In User operations it shows submitting the Query and output operations. It also provides a handle to the user to select a hard disk for scanning as well as parsing the text files which are already there in the database of the search engine.

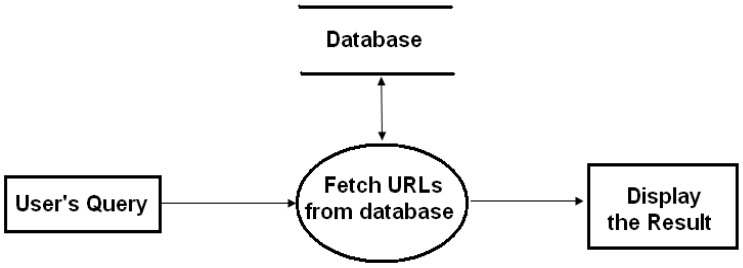


Figure 4. 0 level DFD of proposed Desktop Search Engine

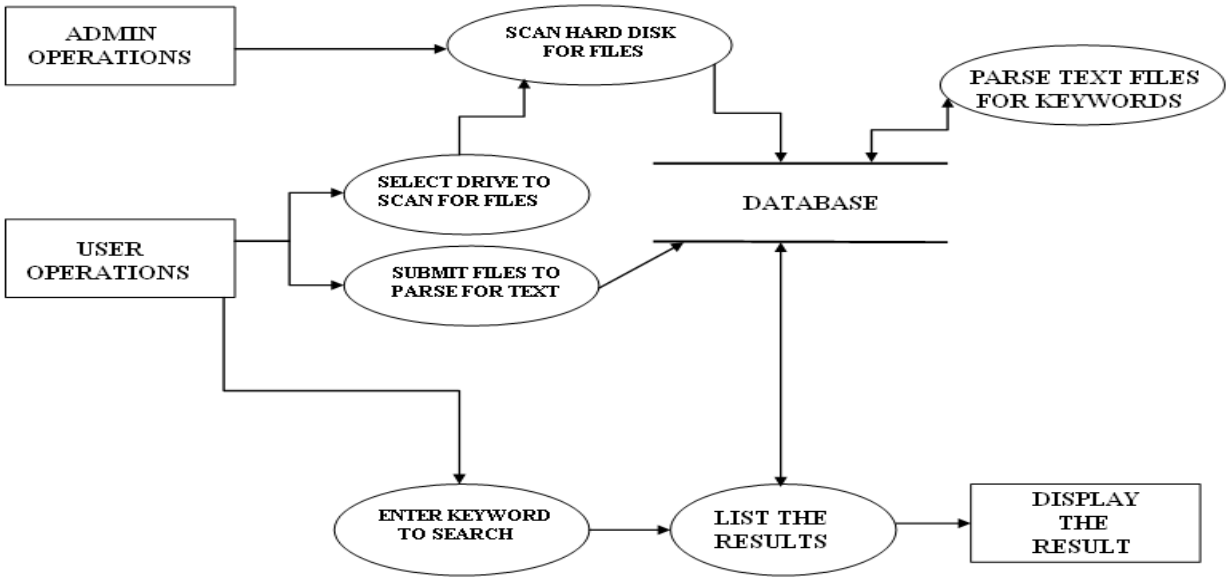
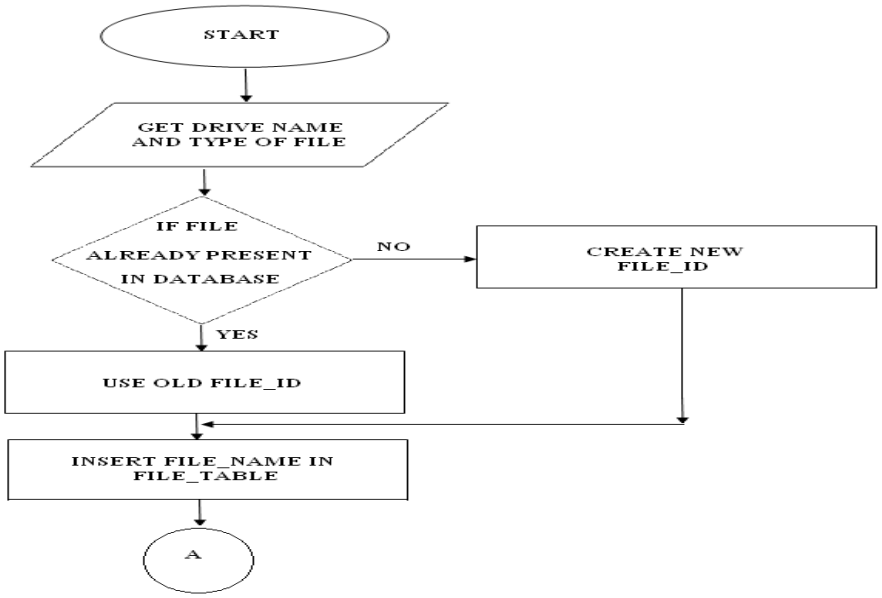


Fig. 5. DFD representing Administration operations and User's operations

The working of the crawler for proposed search engine is shown in Figure 6.It starts with the base drive of the computer i.e. C Drive. First, a document is fetched from list of documents and checks for availability of that document in the database. If document is in the database then it uses the old *page_id* otherwise creates a new *page_id*. Now, the document is parsed for the text. The titles are inserted in the page table and the words of the page are extracted and placed in *word table*.If word is already in *word table* use the same *word_id* otherwise create a new*word_id*. Place the occurrences of the words in the *occurrence table* till all the words present in the *file table* are

registered. These words are used in ranking. Fetch another document from FILE table. If the link already present in *file table,* then delete it from FILE table otherwise more links from FILE table are extracted if more links are present then they are extracted and whole process is repeated otherwise the crawling process is stopped.

The database of the proposed search engine consists of three tables i.e.*File table, word tableand occurrence table* (Fig 7). File table holds all the documents, word table holds all of the words extracted from the documents. The rows in occurrence table correlate words to their containing pages. Each row represents one occurrence of one particular word on one particular page. While page table and word table hold actual data, occurrence table acts only as a reference table. By joining occurrences with page and word, we can determine which page contains a word as well as how many times the word occurs in a particular page and the number of occurrences of a particular word in a particular row determines the ranking technique of the proposed search engine. An efficient ranking and indexing technique is used to place searched web pages in a perfect order of search query so that the user gets the desired output of the searched query.



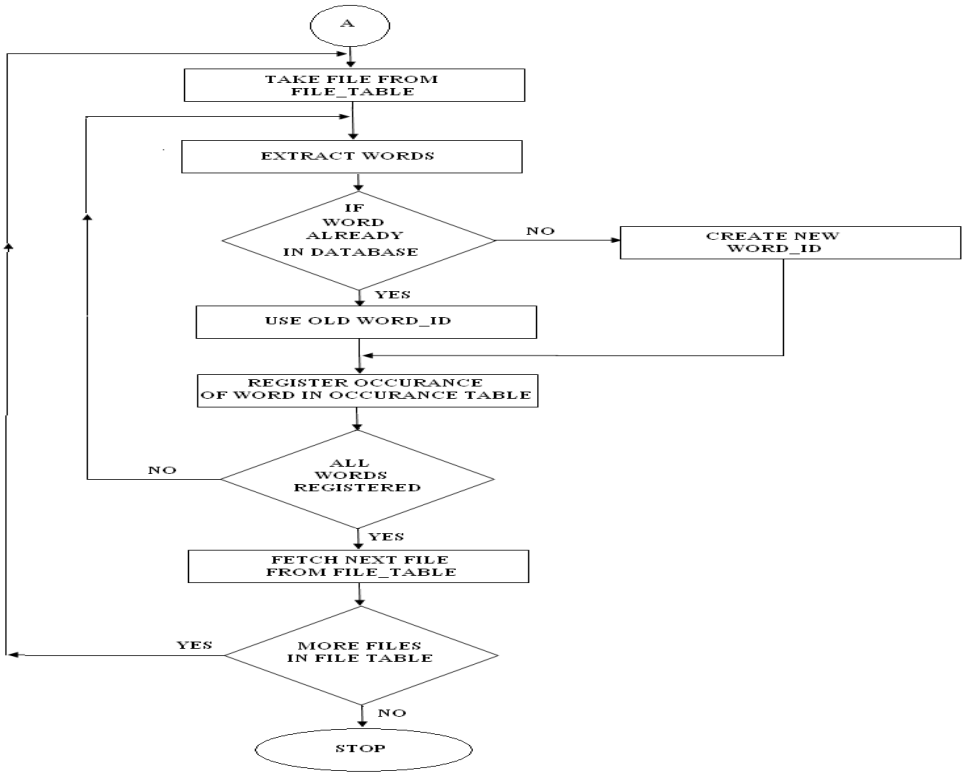


Fig 6. Flow chart of crawling process of proposed Search Engine

File Table

file_table

Field Name	Data Type
file_id	Number
file_name	Text
file_path	Text

Word Table

word_table

Field Name	Data Type
word_id	Number
word_word	Text

Occurance Table

occur_table

Field Name	Data Type
occur_id	Number
word_id	Number
file_id	Number
occur_no	Number

Tables Organized after the Data is inserted.

file_table

file_id	file_name	file_path	Add New Field
1	harsh.txt	D:\hdc\harsh.txt	
2	nitin.txt	D:\hdc\nitin.txt	

word_table

word_id	word_word	Add New Field
2	nitin	
3	is	
4	a	
5	good	
6	boy	
7	harsh	
8	also	
9	kaisan	
10	ho	
11	babu	
12	bhai	

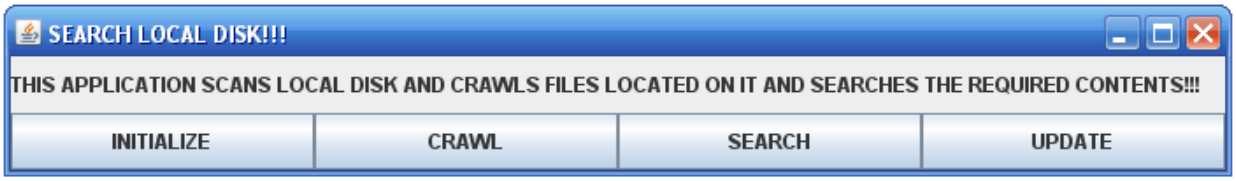
occur_table

occur_id	word_id	file_id	occur_no
1	2	1	1
2	3	1	2
3	4	1	2
4	5	1	2
5	6	1	2
6	7	1	1
7	3	1	2
8	8	1	1
9	4	1	2
10	5	1	2
11	6	1	2
12	9	2	1
13	10	2	1
14	11	2	1
15	2	2	1
16	12	2	1

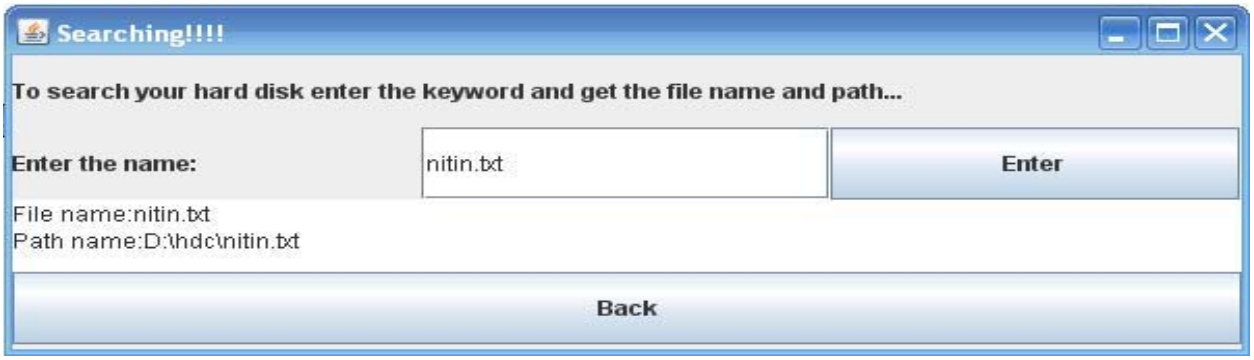
Figure 7: Database of the Search Engine

The search engine has been designed using Java and MS Access. Output screens of the search engine are shown in Figure 8. Main page of the search engine can be reached by running the application (Fig. 8a). Now on searching for a given keyword, it shows output for the searched query appears (Fig 8b).

To add a drive to the database of the search engine, click on the "Initialize" tab and select the desired driveto be crawled as shown in Figure 8c. On submitting the drive to the engine it first searches all the files and then parses every word (Fig. 8d) and documents on the drive are added to the database of the search engine and show the results of the crawling process as under.

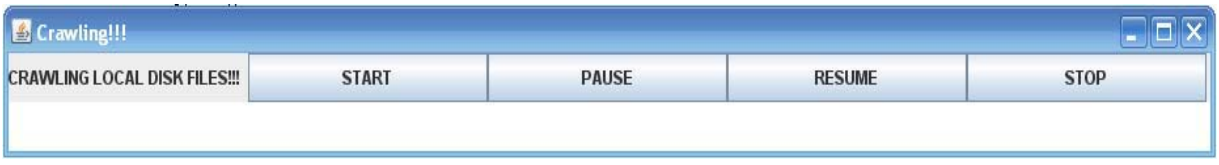


a. Main page of the search engine



b. Output of Query Searched

c. Submitting a URL to be crawled



d. Words crawledfrom the page

Figure 8.Interacting with the search engine

5. CONCLUSION AND FUTURE WORK

In this paper, we conclude that size of the electronic information on the desktop is increasing at a very fast rate. Better searching and updating techniques are needed to extract data from this huge source of information, and to

update desktop search engine database.Efficient ranking and indexing techniques are needed to provide the best possible result to the user, containing the most relevant information about the query given to the search engine.

The proposed system is using techniques for parsing text. This system may be upgraded in the future and may also include many more features for existing system. In future this system may also search for audio and video files. This search engine may parse for word files, pdf files and html files etc. Indexing may be improved to make the results even quicker.

6. REFERENCES

1. AlexandrosNtoulas, Junghoo Cho, Christopher Olston, "What's new on the Web? The Evolution of the Web from a Search Engine perspective.",In Proceedings of the World-Wide Web Conference (WWW), May 2004.
2. ArvindArasu, Junghoo Cho, Hector Garcia-Molina, Andreas Paepcke, SriramRaghavan, "Searching the Web.",ACM Transactions on Internet Technology**,** 1(1): August 2001.
3. Barry M. Leiner, Vinton G. Cerf**,** David D. Clark**,** Robert E. Kahn**,** Leonard, Kleinrock**,** Daniel C. Lynch**,** Jon Postel**,** Larry G. Roberts**,** Stephen Wolff**, "**A Brief History of the Internet", www.isoc.org/internet/history.
4. Brian E. Brewington and George Cybenko. "How dynamic is the web.", In Proceedings of the Ninth International World-Wide Web Conference, Amsterdam, Netherlands, May 2000.
5. Dirk Lewandowski, "Web searching, search engines and Information Retrieval, Information Services & Use", 25 (2005) 137-147, IOS Press, 2005.
6. Franklin, Curt., "How Internet Search Engines Work", 2002, www.howstuffworks.com
7. Grossan, B., "Search Engines : What they are, how they work, and practical suggestions for getting the most out of them," February1997, www.webreference.com.
8. Heydon A., Najork M., "Mercator: A scalable, extensible Web crawler.", World Wide Web, vol. 2, no. 4, pp. 219-229, 1999.
9. IDC White Paper, "An Updated Forecast of Worldwide Information Growth through 2011", March 2008.
10. SinghalNiraj, Dixit Ashutosh, "Web Crawling Techniques : A Review", in proceedings of National Seminar on "Information Security : Emerging Threats and Innovations in 21st Century", Ghaziabad, March 2009, pp 34-43.
11. SinghalNiraj, Dixit Ashutosh, "Retrieving Information from the Web and Search Engine Application", in proceedings of National conference on "Emerging trends in Software and Networking Technologies (ETSNT'09)", Amity University, Noida, India, April 2009, pp 289-292.
12. SinghalNiraj, Dixit Ashutosh, "Need of Search Engines and Role of a Web Crawler", in proceedings of National Conference on "Recent Trends in Computer and Information Technologies Century (RTCIT-2009)", Panipat, Haryana, India, April 2009, pp 26.
13. Sergey Brin, Lawrence Page, "The anatomy of a large-scale hyper textual Web search engine", Proceedings of the Seventh International World Wide Web Conference, pages 107-117, April 1998.