MIT International Journal of Computer Science & Information Technology, *Vol. 3, No. 2, August 2013, pp. 53–57*
ISSN 2230-7621 ©MIT Publications

Search Engine Techniques: A Review

Shivanshu Rastogi
Assistant Professor
Department of CS&IT
Moradabad Institute of Technology
Moradabad, U.P., INDIA
Email: rshivanshu1145@gmail.com

Zubair Iqbal
Assistant Professor
Department of CS&IT
Moradabad Institute of Technology
Moradabad, U.P., INDIA
Email: zubairiqbal17@gmail.com

Prabal Bhatnagar
Assistant Professor
Department of CS&IT
Moradabad Institute of Technology
Moradabad, U.P., INDIA
Email: prabal_bhatnagar@yahoo.com

ABSTRACT

The World Wide Web is a medium of sharing the information globally which results in a huge amount of availability of data over the web. The amount of data being shared grows without bound. In order to extract information that we are interested in, we need a tool to search the Web. The tool required for this purpose is called a search engine. This review covers major aspects of various search engines available worldwide, their working and new features that are being incorporated in these search engines. This paper focuses on the optimal information retrieval through various search engines.

1. INTRODUCTION

The World Wide Web (also known as“WWW” or “Web”) is the world of network-accessible information, the embodiment of human knowledge. It allows people to share information globally. That means it allows anyone toread and publish documents freely. The World Wide Web hides all the detail of communication protocols, machine locations, and operating systems from the user. It allows users to point to any other Web pages without any restrictions. The Web is accessible to anyone via a Webbrowser. Search engines answer tens of millions of queries every day (Brin, 1998). The amount of information on the Web grows exponentially.

2. ELEMENTS OF A WEB SEARCH ENGINE

The various elements of a Web search engine are schematically shown in Fig.1. It consists of following main components:

2.1 Crawler Module

As compared to traditional document collections which reside in physical warehouses [1] such as the college’s library, the information available on WWW is distributed over the Internet. In fact, this huge repository is growing rapidly without any geographical constraints. Therefore, a component used crawler [2] is employed by the search engine which visits the Web pages, collect them and categorize them.

2.2 Page Repository

The downloaded Web pages are temporarily stored in a local storage of search engine, called page repository. The new pages remain in the repository until they are sent to the indexing module, where their vital information is used to create a compressed version of the page.

2.3 Indexing Module

The indexing module takes each new uncompressed page from the page repository extracting suitable descriptors, creating a compressed description of the page. The compressed version of the page is stored in the database, accessible through appropriate interface. Thus, the indexing module is like a black box that takes the uncompressed page as input and outputs a compressed version of the page.

2.4 Indexes

The indexes hold the valuable compressed information for each web page. Three types of indexes are possible. The first is called the content index. Here the content, such as keyword, title, and anchor text for each web page, is stored in a compressed form using an inverted file structure. This link information is stored in compressed form in the structure index. The crawler module sometimes accesses the structure index to find uncrawled pages. Special-purpose indexes are the final type of index. For example, indexes such as the image index and pdf index hold information that is useful for particular query tasks.

Crawlers constantly crawls the Web, brings back new and updated pages to be indexed and stored. In fact, the four modules discussed above and their corresponding data files operate independent of users and their queries as shown in

Fig. 1. It may be noted that, these modules have been separately circled and labelled indicating them to be query-independent. The query module is query-initiated i.e. when a user enters a query in form of keywords, the search engine responds by providing the result pages.

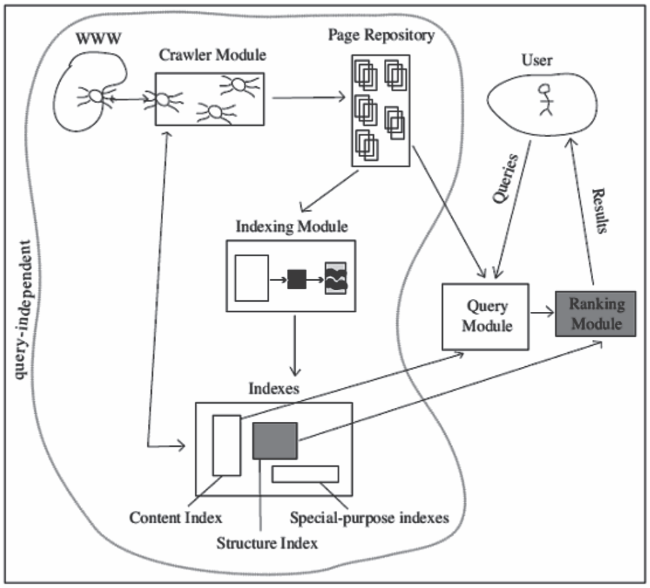

Fig. 1: Elements of a search engine

2.5 Query Module

The query module converts a user's natural language query into a language that the search system can understand and consults the various indexes in order to answer the query. For example, the query module consults the content index and its inverted file to find which pages contain the query terms i.e. the relevant pages, which are passed to the ranking module.

2.6 Ranking Module

The ranking [3] module takes the set of relevant pages and ranks them according to some criterion such as popularity score, content score etc. The output of this module is an ordered list of webpages such that the pages on the top of the list are having the highest rank. The ranking module is the most important component of the search process because the output of the query module often results in too many (thousands of) relevant pages that the user otherwise must sift through. The ranking of a page is computed using rules by combining two scores, the content score and the popularity score. For example, many web search engines give pages, using the query word in the title, as a higher content score as compared to the pages containing the query word in the body of the page. The popularity score is determined from analysis of the Web's hyperlink structure. The content score is combined with the popularity score to determine an overall score for each relevant page. The set of relevant pages resulting from the query module is then presented to the user in order of their overall scores.

2.7 Google Crawler

The Google search engine uses multiple machines for crawling. The crawler works as follows. The crawler consists of five functional components which run indifferent processes. A URL server process reads URLs out of a file and forwards them to multiple crawler processes. Each crawler process runs on a different machine, which is single threaded. It uses asynchronous I/O to fetch data from up to 300 Web servers inparallel. The crawlers transmit downloaded pages to a single Store Server process,which compress the pages and store them to disk. Then the indexer process reads pages from disk. It extracts links from the pages and saves them to a different disk file. A URL Resolver process reads the link file, analyzes the URLs contained therein, and saves the absolute URLs to the disk file that is read by the URL server.

3. TYPE OF DATA RETRIEVED BY SEARCH ENGINE

3.1 Distributed Data

Data is distributed widely over the WWW. It is located at different sites and platforms. The communication links between computers vary widely. Also, there is no topology for data organization.

3.2 High Percentage of Volatile Data

Documents can be added or removed easily in the World Wide Web. These Changes to the documents are usually unnoticed by users.

3.3 Large Volume

The growth of data over the WWW is exponential. It poses scaling issues that are difficult to cope with.

3.4 Unstructured and Redundant Data

The Web is not exactly a distributed hypertext. It is impossible to organize and add consistency to the data and the hyperlinks. Web pages are not well structured. Semantic redundancy can increase traffic.

3.5 Quality of Data

A lot of Web pages do not involve any editorial process. That means data can be false, inaccurate, outdated, or poorly written.

3.6 Heterogeneous Data

Data on the Web are heterogeneous. They are written in different formats, media types, and natural languages.

3.7 Dynamic Data

The content of Web document changes dynamically. The content can be changed by a program such as hit counter that keep tracks of number of hits.

4. INFORMATION RETRIEVAL

As Web is massive, much less coherent, changes more rapidly, and is spread over geographically distributed computers. This requires new information retrieval techniques, or extensions to the old ones, to deal with the gathering of the information, to make index structures scalable and efficiently updateable, and to improve the discriminating ability of search engines.

Before we can understand search engines, we need to understand Information Retrieval (IR), because Web searching is within the field of information retrieval. Before the Internet was born, information retrieval was just index searching. For example, searching authors, title, and subjects in library card or computers. Today, among other things, IR includes modelling, document classification and categorization, systems architecture, user interfaces, data visualization, filtering, and languages. IR deals with the representation, storage, organization of, and access to information items. The user should easily retrieve information of what interests him/her.

There is a difference between information retrieval and data retrieval. In data retrieval, the result of a query must be accurate, it should return the exact match tuples of the query, no more and no less. If there is no change to the database, the result of a query executed at different times should be the same. On the other hand, information retrieval can be inaccurate as long as the error is insignificant. The main reason for this difference is that information retrieval usually deals with natural language text which is not always well structured and could be semantically ambiguous. Data retrieval deals with data that has a well-defined structure and semantics (e.g. a relational database). In addition, data retrieval cannot provide a solution given a subject or topic, but information retrieval is able to do so.

5. USER PROBLEMS

There are some problems when users use the interface of a search engine.

- The users do not exactly understand how to provide a sequence of words for the search.
- The users may get unexpected answers because he/she is not aware of the input requirement of the search engine. For example, some search engines are case sensitive.
- The users have problems understanding Boolean logic: therefore, the user cannot perform advanced searching.
- Learning users do not know how to start using a search engine.
- The users do not care about advertisements, so the search engine lacks funding.
- Around 85% of users only look at the first page of the result, so relevant answers might be skipped.

In order to solve the above problems, the search engine must be easy to use and provide relevant answers to the query.

6. TYPES OF SEARCH ENGINES

From the starting of web, various search engines are developed which are being used. Some are inactive but some are still in use. Table.1 shows the list of search engines year-wise which are active or taken over by different companies due to lack of funding or some other reasons.

Table 1: List of Search Engines.

Year	Engine	Current Status
1994	WebCrawler	Active, Aggregator
	Go.com	Active, Yahoo Search
	Lycos	Active
1995	AltaVista	Active, Yahoo Search
	Excite	Active
	SAPO	Active
	Yahoo!	Active, Launched as a Directory
1996	Dogpile	Active, Aggregator
	HotBot	Active (lycos.com)
	Ask Jeeves	Active (rebranded ask.com)
1997	Yandex	Active
1998	Google	Active
	MSN Search	Active as Bing
1999	GenieKnows	Active, rebranded Yellowee.com
	Naver	Active
	Teoma	Active
2000	Baidu	Active
2002	Inktomi	Acquired by Yahoo!
2003	Info.com	Active
2004	Yahoo! Search	Active, Launched own web search
2005	AOL Search	Active
	Ask.com	Active
	GoodSearch	Active
2006	Quaero	Active
	Ask.com	Active
	Live Search	Active as Bing,
	Guruji.com	Active
2007	Blackle.com	Active
2009	Bing	Rebranded Live Search
	Yebol	Active
	Goby	Active
2010	Blekko	Active
	Yandex	Active, Launched Global
2011	Interred	Active as Interredu
	Yandex	Active, Launched Turkey Search
2012	Volunia	Active
	Interredu	Active

Table 2: Market Share of Search Engines

Search Engine	Market Share in May 2011	Market Share in Dec. 2010
Google	82.80%	84.65%
Yahoo!	6.42%	6.69%
Baidu	4.89%	3.39%
Bing	3.91%	3.29%
Yandex	1.70%	1.30%
Ask	0.52%	0.56%
AOL	0.30%	0.42%

7. SEARCH ENGINE ARCHITECTURES

Most search engines use centralized crawler-indexer architecture. The market share of various search engines is given in Table 2. As the implementations of many search engines are not available to the public. However, there are still some that can be found. They are Google, AltaVista[4], and Harvest [5].

7.1 Google Architecture

The word Google comes from the word googol, which means 10100. The Google search engine (www.google.com) heavily uses the structure present in hypertext. It claims that it produces better results than other search engines today. It references about billion of pages.

Google is mainly written in C/C++ for efficiency reasons. It can run on Solaris or Linux platforms. The architecture is shown in the Fig. 2.

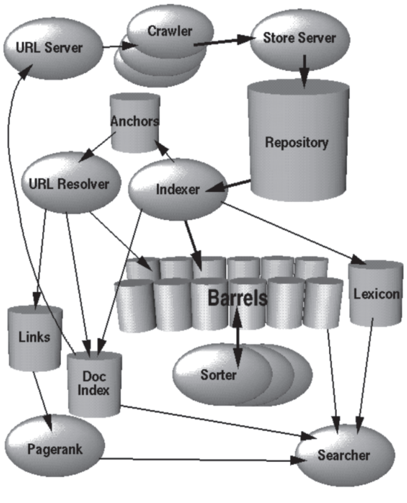

Fig. 2: Google Architecture

The URL Server sends lists of URLs to be fetched by the crawlers. The crawlers download pages according to the list and send the downloaded pages to the Store Server.The Store Server compresses the pages and stores them in the repository. Every Webpage has an associated ID number called a docID, which is assigned whenever a newURL is parsed out of a Web page. The index performs an indexing function [6]. It reads the repository, uncompressed the documents, and parses them. Each page is converted into a set of word occurrences called hits. The hits contain information about a word: position in document, an approximation of font size, and capitalization. The indexer distributes these hits into a set of "barrels" and creates a partially sorted forward index (like bucketsort). It parses out all the links in every Web page and stores important information about them in an anchors file. The anchors file contains information about where each link points from and to and the text of the link. After that, the URL resolver reads the anchors file and converts relative URLs into absolute URLs and in turn into docID. Itputs the anchor text into the forward index, associated with the docID. It generates alinks database for storing links and docIDs. The database is used to compute Page Ranks for all the documents. The Sorter takes the barrels and resorts them by wordID instead of docID in order to generate the inverted index. Also, the Sorter produces a list of wordIDs and offsets into the inverted index. A program called Dump Lexicon takes this list together with the lexicon produced by the indexer and generates a new lexicon to be used by the searcher. The searcher is run by a Web server and uses the lexicon built by Dump Lexicon together with the inverted index and the Page Ranks to answer queries.

7.2 Search by Image Feature of Google

Now we can explore the web in an entirely new way by beginning our Google search with an image. There are a few ways to search by image. We can visit images.google.com, or any Images results page, and click the camera icon in the search box. Enter an image URL for an image hosted on the web or upload an image from our computer. Search by image works best when the content is likely to show up in other places on the web. For this reason, it is likely get more relevant results for famous landmarks or paintings than personal images.

7.3 AltaVista Architecture

This section discusses the AltaVista search engine as an example for demonstrating how this architecture works. The crawler's duty is to run on a local machine and sends requests to remote Web servers. The index is used in a centralized fashion to answer queries from users. The Fig. 3 shows AltaVista's software architecture. It canbe divided into two parts. The first part consists of the user interface and the query engine. The second part contains the crawler and the indexer. In 1998, AltaVista was running on 20 processors. All processors have 130 GB of RAM and over 500 GB of hard disk space. Only the query engine uses more than 75% of these resources. (Baeza-Yates, 1999)There are two problems with this architecture. The first problem is data gathering in the dynamic Web environment, which uses saturated communication links, and high load at Web servers. The second problem is the volume of the data. The crawler-indexer architecture does not cope with Web growth in the near future.

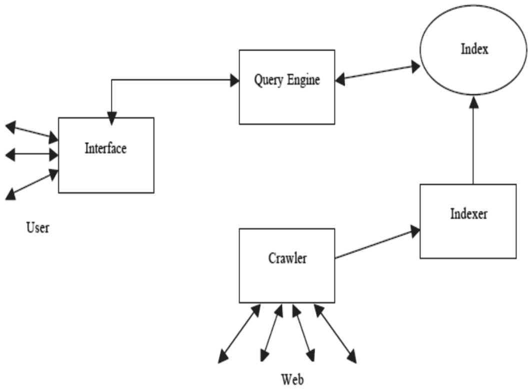

Fig. 3: The typical crawler- indexer architecture (AltaVista)

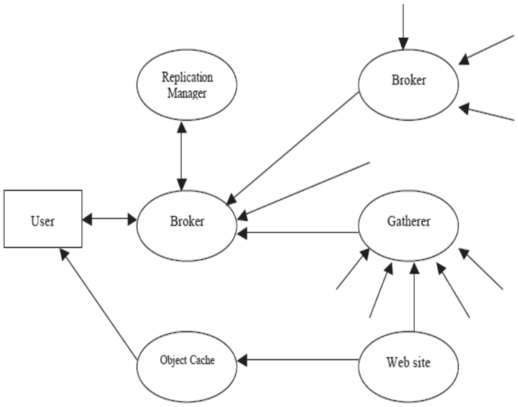

Fig. 4: Harvest Architecture

7.4 Harvest Architecture

There are several variants of the crawler-indexer architecture. One of the variants is called Harvest. Harvest is the most important variant that uses distributed architecture togather data and distribute data. It is used by CIA, NASA, the US National Academy of Sciences, and the US Government Printing Office (Baeza-Yates, 1999). In addition, Netscape's Catalog Server is a commercial version of Harvest and Network Appliances' cache is a Commercial version of the Harvest Cache. As shown in Fig. 4, Harvest introduces two main elements: gatherers and brokers. The job of gatherers is to collect and extract indexing information from one or more Webservers. Gathering times are specified by the Harvest system. The times are periodic as suggested by its name, Harvest. The job of brokers is to provide the indexing mechanism and the query interface to the data gathered. Brokers receive information from gatherers or other brokers to update their indices. Also, brokers can filter information and send itto others, so that other brokers are saved time. Depending on the configuration of gatherers and brokers, server's workload and network traffic can be balanced. The harvest system builds topic-specific brokers and focuses the index contents there by avoiding many of the vocabulary and scaling problems of generic indices. In addition,the system provides a replication manager (to replicate servers for enhance user-bases calability) and an object cache (to reduce network and server load).

8. RANKING

Ranking is the heart of the search engine. In order to produce a good search engine, weneed to know how to rank pages properly for the result documents. There is not much information available about this in the public. Today, most search engines use variations of the Boolean or vector model to do ranking. Recall that search engines do not allow access to the text, but only the indices, because it is too expensive in terms of time and space. So, when searching, ranking must use indices while not accessing the text. Besides that, there are also other difficulties as well. There might be too many relevant pages for a simple query. Also, it is difficult to compare two search engines, because of their continuous improvement.

9. CONCLUSION

This review describes the overview of Web search engines. The goal of this paper is to help people perform Web searching easily and effectively. It discusses the different components of search engines such as architectures, user interfaces, ranking algorithms,Web crawlers, meta searchers and indices. Also, it investigates other issues such as information retrieval, characteristics of the Web, different types of search engines,searching guidelines and possible future research. It provides reasons why we need to study search engines, and it provides relevant references for readers to proceed further. More important, the readers should try out different search engines that are available today.

REFERENCES

[1] Labio, W.J., Quass, D., Adelberg, B.,"Physical database design for data warehouses", Data Engineering, 1997. Proceedings.13th International Conference.

[2] Gupta, P., Johari, K., "Implementation of Web Crawler" Emerging Trends in Engineering and Technology (ICETET), 2009 2nd International Conference.

[3] Kumar, G., Duhan, N., Sharma, A.K., "Page ranking based on number of visits of links of Webpage" Computer and Communication Technology (ICCCT), 2011 2nd International Conference.

[4] Spink, A.,"Multitasking Web search on Alta Vista"

Information Technology: Coding and Computing, 2004. Proceedings. ITCC 2004. International Conference.

[5] Qureshi, P.A.R., Memon, N., Wiil, U.K. Arampelas, P., Sancheze, J.I.N., "Harvesting Information from Heterogeneous Sources", Intelligence and Security Informatics Conference (EISIC), 2011 European.

[6] Stark L., Bowyer K., "Indexing function-based categories for generic recognition", Computer Vision and Pattern Recognition, 1992. Proceedings CVPR '92., 1992 IEEE Computer Society Conference.