Using WebSQL to Query the World Wide Web[1]

Ramzi A. Haraty and Fadi Osman
Lebanese American University
P.O. Box 13-5053
Beirut, Lebanon

Abstract:

The Internet is growing faster than ever, in an extremely decentralized (some would say anarchic) way, spanning over 130 countries and millions of users, covering every imaginable topic. Unfortunately this has caused some trouble (such as the "lost in hyperspace" syndrome), thus some researchers felt the need to uncover a structure of the World Wide Web (the major part of Internet), that would ease the search process.

This paper discusses two well-known solutions that attempt to alleviate the trouble caused by searching the World Wide Web. It then notes an observation of a slight weakness in these solutions and provides its own solution by merging the two existing ones.

Keywords:

Authorities, CLEVER, HITS, Hubs, Hypermedia, Sherlock, World Wide Web.

Introduction:

The world has reached the state of "information overload". The proliferation of books, research, and universities had only a minor impact on this overload, when compared to the Internet, and specifically to World Wide Web (WWW), which has grown enormously, gaining a wide global success due to its ease of use, and a "natural" presentation of information [5].

People are just saturated with the amount of information available to them, and tend to get lost, especially when using current Internet search facilities (such as Alta Vista, Yahoo, Infoseek, etc.). These Internet indexes are the only available tools to search the WWW, and to add to the confusion of users, each uses its own parameters requiring different input.

Furthermore, the results they return frequently contain a vast amount of irrelevant information. The only alternative is to browse-search, which does not help much! This paper provides a brief overview of hypermedia. It then discusses two well-known solutions to the search problem, namely the HITS and CLEVER techniques. The paper then introduces our solution by combining these two methodologies to produce even better search results.

Hypermedia Overview:

The traditional linear presentation of information has long been considered as a problem. As soon as the papyrus roll needed some effort to read (imagine a roll ten meters long!), man invented a new way to present information - the book. By cutting the papyrus roll into several parts, and

[1] Proceedings of the International Conference on Scientific Computations. Beirut, Lebanon. March, 1999.

classifying these into chapters, information retrieval was greatly improved. The book gave birth to indexes and thesauri. Unfortunately, information was still presented in a “unnatural way”, forcing the associative human mind to work linearly.

In 1945, Bush described a machine, the Memex, that could be used to “browse and make notes in a voluminous online text and graphics system. It was to have several screens, and a facility for establishing a labeled link between any two points (nodes) in the library”[5]. Due to technological limitations of the time, the Memex was never built, but it may well be the first description of hypertext ever (Bush did not call it “hypertext”).

The term hypertext originated in 1965, while Nelson was working on his Xanadu project [5], an ambitious term project whose ideas are still exploited. The concept of hypertext was then applied to graphics, sounds and other media, resulting in what is known as hypermedia, vastly used nowadays.

Applications of Hypermedia:

The results of the experimental hypertext systems (Memex, Xanadu) lead to a worldwide explosion of interest. But we had to wait till the middle of the 80’s, for the development of commercial systems, specifically targeted towards the large public. Building on their image of user friendliness, Apple Macintosh presented in 1987 its HyperCard, bundling it with their computers. Integrating graphics, sounds and text HyperCard gained popularity, and by 1989 there were hundred of thousands of HyperCard Stacks. A major characteristic of HyperCard was its use of hypertext to present the help topics.

The flexibility of hypermedia lead to the belief that anything could be done. Relying on its solid implementation with objects, hypermedia was used as representation of knowledge that was to change the world.

Researchers began exploring ways to represent relational databases, objects, facts and rules as hypermedia, which would solve many problems caused by heterogeneous databases. People even started thinking of ways to convert all existing printed material to hypertext, creating a huge database. With its inherent indexing capabilities, hypermedia seemed like the ideal way to store and present this huge amount of information. However, limitations started appearing, caused mainly by this freedom. Instead of helping the user to focus and gain information on his requested topic, hypertext lead to loss in focus.

Researchers began exploring ways to represent relational databases, objects, facts and rules as hypermedia, which would solve many problems caused by heterogeneous databases. People even started thinking of ways to convert all existing printed material to hypertext, creating a huge database. With its inherent indexing capabilities, hypermedia seemed like the ideal way to store and present this huge amount of information. However, limitations started appearing, caused mainly by this freedom. Instead of helping the user to focus and gain information on his requested topic, hypertext lead to loss in focus.

These problems were greatly emphasized by the birth of the HyperText Markup Language (HTML) and the WWW. Following the success of WWW, almost everything is now represented in hypertext. Desktop computers are following the trend to blend their environment with the Internet where everything could be found. It took us, however, a lot of time to be able to localize the information we wanted (80% of total project time). This shows the difficulty of searching the Internet.

Apple Macintosh tried to solve this difficulty by presenting Sherlock, a kind of unified interface to index engines (see Figures 1 and 2), described as the intelligent way to search the Internet. However, Sherlock is just an "interface", and still relies heavily on the indexes it queries, although having advanced natural language recognition, and presenting the search results in a rather appealing way.

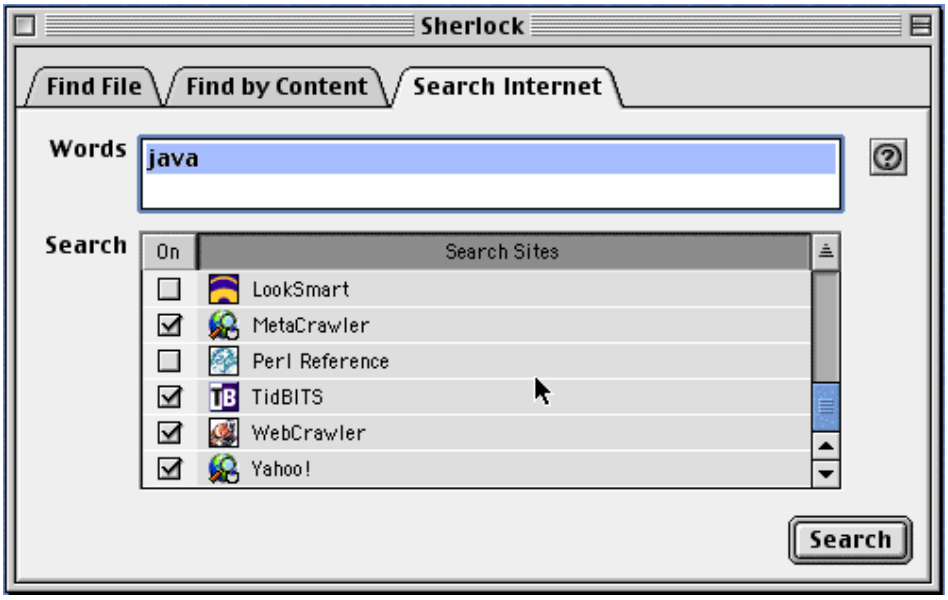

Figure 1. Sherlock Interface.

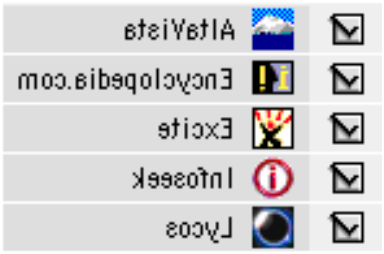

Figure 2. Sherlock List of Search Engines

WebSQL - a WWW Querying Language:

Motivations behind WebSQL:

As mentioned earlier, current WWW search engines are rather limited. Their biggest limitation is that their search process occurs in two separate phases. They would first navigate through the Internet, building their indexes according to the contents of web pages, then search these indexes in response to queries, thus ignoring additional information which belong to web documents. One important information is the location of a document. For example, consider a keyword search on 'IBM', 'personal computers' and 'price'. A search robot, 'Metacrawler', returned 92 references, including "I bought a Mac" T-shirt [3]. Clearly, if we were to indicate we want to search www.ibm.com for these info, we would have found what we wanted, a list of IBM PC prices.

Another issue is the cost of connection that is whether a site is local or remote. Clearly, we would prefer to access pages that have a faster download rate.

Overview of WebSQL:

WebSQL was the result of Mihaila's [4] effort to design a structured model of the web. He observed that when communicating with the web clients (browsers), web servers sent information concerning the modification date of a document, its type, length, title, and finally text. Similarly, every link on a web page could be represented by its label, the document it originates from and its target. Thus he proposed a "minimalist relational approach", consisting in associating every web object with the following tuples:

- URL[protocol, server, file, reference]
- Document[URL, title, text, type, length, modif].
- Anchor[base, href, label].

He then studied thoroughly the model, establishing for it a Formal Calculus, and thus was able to derive an SQL-like language enabling him to formulate such queries as:

Query 1. "Find all HTML documents about 'hypertext'":

SELECT d.url, d.title, d.length, d.modif
FROM Document d
SUCH THAT d MENTIONS "hypertext"
WHERE dtype = "text/html";

Query 2. "Starting from the Department of Computer Science in Toronto, find all documents that are linked through paths of length two or less containing only local links. Keep only the documents containing the string 'database' in their title."

SELECT d.url, d.title
FROM Document d
SUCH THAT http://www.cs.toronto.edu=|∏|∏∏ d
WHERE d.title CONTAINS "database";

Such a language means the possibility of writing applications with embedded WebSQL.

Drawbacks of WebSQL:

One obvious drawback is its limitation on performing nested queries and its limited set of "link predicates"[5].

It also presents a slight weakness, which is its reliance on the results returned by the search engines (although it greatly improves on them). In fact, search engines return a somehow large collection of pages, which may verify our conditions, but still are poor references to what we seek (Note this drawback only appears when we do not give WebSQL a controlled navigation, as in Query 1).

Authoritative Sources: a Solution:

Description of Authorities and Hubs:

This approach was conceived while trying to find a relative structure underlying the WWW. Studies by Kleinberg [2] shows that the WWW is far less chaotic than it is thought to be. In fact he observed that a group of documents concerning a certain topic usually contained a couple of sets: Authorities and Hubs (see Figure 3). Authorities are the pages most frequently referred to in this group, while Hubs are the pages that refer most to Authorities. Kleinberg conceived an algorithm he called HITS to get these two sets. HITS first queries the search engines, then expands the resulting set by all pages pointed to by this set. Then it traverses these pages while maintaining a table of counts of references and then repeats the process a couple of rounds. It was shown that this actually takes some time but produces pretty accurate results.

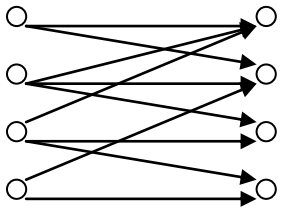

Figure 3. Hubs and Authorities

IBM researchers improved on HITS by adding another level of traversal [5], and looking at the text around the links. They also filter out the sites frequently referenced by practically all pages such as Netscape, Microsoft, and Apple.

Authorities, Hubs, and WebSQL:

The idea is to add to WebSQL the possibility to use HITS or CLEVER to get a set of starting pages to the queries that do not specify ones. WebSQL uses definitions to propose an index server, context attributes, and link predicates. We could, for example, specify HotBot as index server instead of the default AltaVista by

DEFINE INDEX = “HotBot”;

We could add to WebSQL the possibility to choose a distillation method:

DEFINE DISTILLATION = “none”;
DEFINE DISTILLATION = “authority”;
DEFINE DISTILLATION = “hub”;
DEFINE DISTILLATION = “community”;

where “community” = both “authority” and “hub”.

By choosing “none”, we would choose the default behaviour of WebSQL, “authority”, “hub”, or “community” would mean to start the query with the URL resulting from the distillation algorithm (HITS or CLEVER) in the FROM clause:

SELECT xxx

FROM Document d
SUCH THAT << authority | hub | community >> ….
WHERE …

Note that "hub" and "community" are included because some it was observed that sometimes Hubs were better starting places than Authorities. [5]

This method of distillation considerably increases networks access and thus would cause a loss in the speed of the query. However, sometimes we are willing to loose sometime to get better results.

Conclusion:

WebSQL proposes a method of structuring the WWW by proposing a relational view to web documents. It works pretty good in all cases but results of directed queries, where we specify the starting documents, were better than general queries. We proposed an extension to WebSQL, allowing it to take advantage of HITS or CLEVER, therefore improving the general queries.

However, all of these methods heavily rely on search engines, and if software companies continue on improving the search algorithms (everybody would buy something that could help search the Internet), we could get to a point where search engines will refuse queries coming from CGI, and thus kill all our efforts (remember that these tools allow us to short-circuit all the publicity, which keep index servers running). Another alternative would be queries only allowed by registrations!

References:

[1] – Chakrabati, Dom, Raghavan, Rajagopalan, Gibson, Kleinberg, "Automatic Resource Compilation by analyzing Hyperlink structure and associated text", http://decweb.ethz.ch/WWW7/1898/com1898.html.

[2] – Kleinberg, "Authoritative Sources in a Hyperlinked Environment", IBM RJ 10076, 1997.

[3] – Mendelzon, Mihaila, Milo, "Querying the World Wide Web", SPDIS, 1996.

[4] – Mihaila, "WebSQL – an SQL-like Query Language for the World Wide Web", Computer Science Ms. Thesis, University of Toronto, 1996.

[5] – Parsaye, Chignell, Khoshafian, Wong, "Intelligent Databases", Wiley, 1989.

Biography

Ramzi A. Haraty is an Assistant Professor of Computer Science at the Lebanese American University in Beirut, Lebanon. He received his B.S. and M.S. degrees in Computer Science from Minnesota State University - Mankato, Minnesota, and his Ph.D. in Computer Science from North Dakota State University - Fargo, North Dakota. His research interests include database management systems, artificial intelligence, and multilevel secure systems engineering. He has well over 35 journal and conference paper publications.