December 2001

A New Architecture for Web Meta-Search Engines

Zheng Li
New Jersey Institute of Technology

Yuanqiong Wang
New Jersey Institute of Technology

Vincent Oria
New Jersey Institute of Technology

A NEW ARCHITECTURE FOR WEB META-SEARCH ENGINES

Zheng Li
CIS Department
New Jersey Institute of Technology
zxl8078@njit.edu

Yuanqiong Wang
CIS Department
New Jersey Institute of Technology
yxw9836@njit.edu

Vincent Oria
CIS Department
New Jersey Institute of Technology
oria@cis.njit.edu

Abstract

Web search engines have become the necessary tools for Web users to search useful information. However, users often face problems such as how to find the right information promptly with the least effort. This paper examines the limitations of current Web search engines and proposes a new Web meta-search engine architecture. The proposed architecture uses a multi-agent approach to process the user queries with greater personalization functionality and higher result quality than regular meta-search engines. Thus, the propose method improves the usability and efficiency of current search engines.

Keywords: Search engine, meta-search engine, information retrieval, user interface design, multi-agents

Introduction

More and more information is available on the Internet, accessing the right information has become a critical problem. The value of the Internet resides on how the use of the rich information it provides. And human efforts to deal with Web information retrieval can be traced back to a decade.

In early 1990s, before browsers came into the world, there was the WAIS and its XWAIS version, that were using a specific format on a server to retrieve information on the Web. GOPHER as an index also appeared around 1991. The earliest search engine, Lycos, was introduced around 1994. John Leavitt's spider was linked to an indexing program by Michael Mauldin to make the Lycos work. Also in 1994, Yahoo — a catalog style tool became available (Grossan 1997).

As time goes by, more and more search engines have emerged and their number is estimated to be over 160 (Lawrence et al. 1999). Among them, there are about over 50 so called "meta search engines" (Barker 2000). Some general-purpose search engines can find information on all kinks of categories, while many others are limited to specific categories, such as education, finance, kids etc. In general, the search engines have greatly ease the pain of Internet users when they are trying to grab useful information on the Web.

However, according to Lawrence et al. (1999), most of the search engines cannot completely satisfy user's requirements. For example, most search engines can only index and process a very small part of the web pages on the Web; A lot of the indexing methods are based only on meta-tags within the documents; the updating period is quite long, etc. All these problems may regretfully result in incomplete information retrieval. That is, users can only get very small proportions of the information they want.

Moreover, the poor interface design in many search engines hinders the full use of their advanced functions. And the search results are often very inaccurate and irrelevant (Lawrence et al. 1999; Pollock et al. 1997; Grossan 1997).

Many researchers have been working on evaluating and improving current Web search engines (Henzinger 2000; Winship 2001; Barker 2000; Pollock et al. 1997; Zorn et al. 1996; Kingoff 1997; Grossan 1997). Steve Lawrence et al. (1999) did an intensive work on evaluating Web search engines. Monika Henzinger systematically studied how "Google" works on Web information retrieval (Henzinger 2000). Glover et al. have proposed a new architecture (Glover et. al., 1999). Yang et al. proposed a solution on natural language processing on Web search (Yang et al. 1997).

In this paper, we analyze the problems of current Web search engines, justify the need of a new design, present some ideas on how to improve current Web search engines, especially in the user interface (UI) designs, and then propose a new architecture for Web meta-search engines with a multi-agent support in the UI to make search engines work more efficiently while user-friendly.

What is a Web Search Engine?

The evolving tools to find and access the rich Multi-media Web beyond FTP archives are called search engine (Grossan 1997). Search engines utilize indexing software agents called “robots” or “spiders”. These agents are programmed to constantly "crawl" the Web in search of new or updated pages. They essentially go from URL to URL until they have visited every Web site on the Internet.

There are various types of search engines. As we mentioned earlier, some search engines are for general-purpose searches. These general-purpose search engines can further be divided into two sub-categories: direct and indirect. Direct search engines are those that have their own database of web pages and indexes. Example include AltaVista, Excite, Google, Infoseek, Lycos, etc. Indirect search engines, also called Meta-search engine, comprise MetaCrawler, DogPile, AskJeeves, and InvisibleWeb. They usually do not have their own database, but send queries to several other direct search engines before combining the results.

Some search engines provide hierarchical directories “Yahoo!”, and all the portals on the Internet are of this kind of search engine. They are also called "directory".

There are also some specialized search engines that provide services in some specific areas. For instance, Ahoy is a home page finder; Jango and Junglee are shopping robots, Applet finders, etc.

Some search engines can search by example. For instance, Alexa can search related information by using "What's related" criteria; Excite use "More like this" as their criteria to search; Google use "Googlescout", etc.

Search engines like Firefly and GAB use collaborative filtering to improve their performance. Some others use Meta-information, which includes "search engine comparison results", "query log statistics", and so on.

Basic components of a search engine include a spider, an indexer, and a search interface. The spider, also called crawler, is in charge of collecting the documents on the web. The indexer processes and represents the data. The search interface gives the answer to the queries.

Search engines always use spiders to ‘comb” the Internet looking for documents and their Web addresses. The documents and Web addresses are collected and sent to the search engine's indexing software. The indexing software extracts some information from the documents, and stores it in a database. The kind of information indexed depends on the particular search engine. Some search engines index every word in a document; others index the document title only. When users perform a search by entering keywords, the database is searched for documents that match the user query. The search engine assembles a web page that lists the results as hypertext links.

Some of Current Search Engines

In order to understand the current state of the Web search engines, we will describe some of the most representatives with their features (Winship 2001).

Alta Vista (http://www.altavista.com) is among the famous search engines. In addition to its own database, it uses Usenet, sounds and pictures as resources. For multi-word input, implied "OR" relationship is supported. It allows users to use "+" or "-" in their query. In advanced search, it allows "AND", "OR", and "NOT" in its logic. It can search title, URL, text, etc. It uses truncation. For phrases, it uses adjacency, which means two words are next to each other. It provides search in proximity by using "NEAR" in the query.

Excite (http://www.excite.com) uses Usenet, news, email addresses in addition to its own database as its resource. It also uses implied "OR". It supports "+" or "-" logic in its query, users can also use "AND", "OR", and "NOT" to control the results. It has no specific fields for querying, which means any place that contains the information will be checked. It also uses adjacency in phrases; Truncation and proximity are not allowed.

Google (http://www.google.com) uses only its own database. It uses "AND" as its implied logic when users input several words in the query. It allows "+" and "-" logic, but it doesn't support "AND", "OR", "NOT" in the user input. It doesn't search in specific fields, either. Neither truncation nor proximity is supported. It uses adjacency in phrase searching.

Hotbot (http://www.hotbot.com) also uses Usenet, sounds, and pictures as its additional resource besides its own database. It uses implied "AND" relationship. It can search title, domain specifically. It doesn't support proximity, but it supports "+", "-", "AND", "OR", "NOT" logic in its queries. It uses truncations and adjacency.

Infoseek (http://www.infoseek.com) uses Usenet, email address, news, and pictures as its additional resource. It provides specific searches in the title and the URL fields. It uses implied "OR" and supports "+" or "-" logic. It doesn't support "AND", "OR", "NOT", and proximity in its queries. It does not truncate results, either.

Lycos (http://www.lycos.com) as the first search engine, uses sounds and pictures as its additional sources. It does not use truncation. But it does support "+" or "-", "AND", "OR", "NOT" logic. It uses "NEAR" to support proximity. It supports adjacency in phrase searches.

NorthernLight (http://www.northernlight.com) is also called a research search engine. It usually uses Journal articles as its resource. It uses implied "AND", supports "+", "-", "AND", "OR", "NOT", adjacency, and proximity search. It provides search in title, text, and URL specifically.

Many researches have been carried out on comparing and evaluating current Web search engines. According to these works, despite all kinds of effort that have been made on Web search engines, the result is very disappointing. The well-known and most frequently cited evaluation is Lawrence and Giles's survey taken on 1999 (Lawrence et. al., 1999). The following five problems are often identified in current search engines:

1. ***Search engine coverage has decreased*** – search engines' indexing capability is increasingly falling behind the fast growth of the Web. Relative to the estimated size of the publicly indexable web contents, search engine coverage has decreased substantially since December 97. The size of the Web is estimated to be more than 800 million indexable pages, encompassing about 15 terabytes of information or about 6 terabytes of text after removing HTML tags, comments, and extra white space. However, no single search engine coverers more than about sixteen percent of the total indexable pages on the Web.

2. ***Unequal access*** – Web search engines typically index a biased sample of web. Search engines either follow the links (i.e. URL) to find new pages, or by analyzing user registration. They are typically more likely to index sites that have more links to them (more 'popular' sites). They are also typically more likely to index US sites than non-US sites (AltaVista is an exception), and more likely to index commercial sites than educational sites.

3. ***Out of date*** – By looking at the percentage of documents reported by each engine that are no longer valid (because the page has moved or no longer exists), and age of new documents found that match given queries, it was found that the mean age of a matching page when indexed by the first of the nine engines is 186 days.

4. ***Low metadata use*** – Many search engines index Web information based on web page defined metadata. However, the simple HTML "keywords" and "description" meta-tags are only used on the homepages of 34% of sites. Only 0.3% of sites use the Dublin Core meta-data standard.

5. ***Information distribution*** – Information distribution on the Web is uneven. There are about 83% of Web sites with commercial content and 6% with scientific or educational content. Only 1.5% of Web sites contain pornographic content.

Meta-Search Engine

A meta-search engine is the kind of search engine that does not have its own database of Web pages. It sends search terms to the databases maintained by other search engines and gives users the results that come from all the search engines queried.

The reason of being of the meta-search engines is clear. First of all, since each single search engine can only deal with less then 16% of the total information on the web, and each of them covers different scales, combining several of the search engines should result in a higher recall rate, at least a better recall than using a single search engine. According to Barker (2000), results from submitting very comparable query to different search engines can differ widely (about 40%), but also contain some of the same sites (about 60%). Lawrence and Giles found that combining the results of 11 major search engines increases the coverage to about 42% of the estimated size of the publicly indexable web (Lawrence et al. 1999). This means by properly using meta-search engines, the information coverage can be improved.

Second, by applying filters or improved algorithms towards the directive query results from the general search engines, the precision of the meta-query is expected to significantly improve.

The mechanism and algorithms that meta-search engines employ are quite different. The simplest meta-search engines just pass the queries to other direct search engines. The results are then simply displayed in different newly opened browser windows as if several different queries were posed. Some improved meta-search engines organize the query results in one screen in different frames, or in one frame but in a sequential order. Some more sophisticated meta-search engines permit users to choose their favorite direct search engines in the query input process, while using filters and other algorithms to process the returned query results before displaying them to the users.

Problems often arise in the query-input process though. Meta-Search engines are useful if the user is looking for a unique term or phrase; or if he (she) simply wants run a couple of keyword. Some meta-search engines simply pass search terms along to the underlying direct search engine, and if a search contains more than one or two words or very complex logic, most of them will be lost. It will only make sense to the few search engines that supports such logic.

In order to get a quicker response, the meta-search engines spend a short period of time in each database and it is estimated that they retrieve only about 10% of the results in each of the databases queried (Lawrence et. al., 1999). Meta-search engines usually do not retrieve all the data related to the user query.

In the following we compare three relatively powerful meta-search engines with some direct search engines like “Alta Vista” and “Yahoo!”.

Dogpile (http://www.dogpile.com) uses About.com, Alta Vista, Dogpile Open Directory, GoTo.com, Infoseek, Lycos, Lycos' Top 5%, Thunderstone and “Yahoo!” as its underlying search tools for Web search queries. It is customizable and supports OR, NOT, and "()". The default logic between different query words is "AND". Dogpile does not sort the query results received from the search engines. The output is a concatenation of the different results with the ranking by each search tool; therefore, there may have duplicates in Dogpile's output. The results are displayed sequentially.

Inference Find (http://www.infind.com) is another Meta-search engine. It uses Alta Vista, Excite Search, Infoseek, Lycos, WebCrawler, and “Yahoo!” as its search tools for Web search queries. It’s not customizable. Similar to Dogpile, it uses “AND” as its default logic and supports “OR”, “NOT” and "()" logic. It sorts the results into clusters of words or phrases found. Duplicates are eliminated.

MetaCrawler (http://www.metacrawler.com) uses Lycos, About.com, Alta Vista, Excite, Infoseek, Looksmart, Thunderstone, WebCrawler, and “Yahoo!” as its tools for Web search queries. It is customizable, and also provides "Power Search". It consolidates results in one large list, which are ranked by a "vote" score. It doesn't support Boolean logic in its queries. It supports “ALL”, “ANY” or exact PHRASE to, and allows the use of "+/-", "" around phrases.

Improvements of Web Search Engines

There are different ways to improve the performance of web search engines. Generally speaking, there are three main directions:

I. Improving user interface on query input

II. Using Filtering towards the query results
III. Solving algorithms in web page spying and collecting, indexing, and output

Method III is the fundamental solution for any direct search engines to deal with the problems of unequal accessing, out-of-date information, and low metadata using, as well as information coverage. It is also very important to look at the user interface issues that method I and II are dealing with — How to handle user queries effectively and present the results efficiently.

A survey conducted on over 22,000 Internet search engine users in 1997 yields the following features as very important to make a search engine successful:

- Ease of use
- Speed of loading and response
- Reliability and accuracy of results
- Organized and up-to-date information

Beside the algorithms used by a search engine to provide reliable and accurate information, the user interface is another important components of a search engine. Non-computer science users usually feel very frustrated when dealing with different user interfaces of different search engines. Even for computer professionals that are not quite familiar with some specific search engines, it is sometimes difficult to point out what is the proper input logic, and what kind of input format is allowed etc. Both kinds of users are always annoyed by the high percentage of "irrelevance" in the search results, and they often loose their patience and do not go through all the result pages. This may lead to serious loss of useful information that is ranked lower in the result.

In this paper, we concentrate on improving the user interface of Web meta-search engines. Our objective is to design a better user interface to help users understand the functions of a search engine and reduce their frustration; confine the search categories and speed up the response; provide more relevant and accurate information in an organized way, and greatly improve the recall and precision rate of the web information retrieval.

A New Architecture

In analyzing some of the most popular meta-search engines or powerful direct search engines, we found that there are several very good features in the user interface design that should be kept in future search engines or meta-search engines:

- ***Customizable***: Some search engines are customizable. Users can choose the way to do their search. For example, there are search options, such as, simple-search /powerful-search (Northern Light), faster/smarter /custom search (C4 Total Search Technology) etc. In the later one, faster searches use fewer search engines and spend less time qualifying and sorting the results. Search smarter uses more search engines, spends more time qualifying and sorting the result, removing duplicates, and verifying the status of the link. Custom search can choose searches from different categories, like search the web/family search/news search etc., and can have different preferences on which search engines to choose, how the results should be returned etc.

- ***User profile***: Some search engines provide ways for setting up user profiles. Normally the use of this function is not free. The user profile contains both the user search preferences and personal information. In the preference items, the user can choose the number of items he(she) wants to be returned, the number of items he(she) wants to be displayed, the preferred search engines, etc. The user can change the search preferences and the personal information whenever at any time.

- ***Categorizing***: Many advanced search engines support categorized searches. For example, Northern Light optimizes its search by limiting it to several main categories—Simple-Search, Power-Search, Business-Search, Investment-Search, and Stock-Quotes. Business-Search is specialized in searching industry-focused Web pages, market research, economic analysis, and company reports for business professionals. Before submitting their searches, users can select from all sources or business only, limit industry to Insurance/media/music-industry etc., and limit documents to Business Week /Fortune /Press-release etc. Users can also select the date and time range and how the results should be sorted.

- ***Help information***: Many search engines put little piece of search hints on their search page. Some also provide instant feedback on how to submit a correct search, what is the search status etc. This kind of information leads to a better use of the search engine.

Often these features are highly interwoven, and do not necessarily work alone. But they make search engines very flexible and easy to use.

From the analysis of several Web meta-search engines in the previous section, we know that a meta-search engine must make decision on:

- How to verbalize a query,
- Which sources to query,
- How to modify the submitted query to best utilize the underlying search engines,
- How to order and display the results.

Most current meta-search engines allow users to decide on part of these factors, but not on all of them. In order to improve the performance of a meta-search engine, all these factors should be thought about.

In this section, we propose a new architecture for Web meta-search engines, which integrates not only the features discussed above, i.e. customizable, user profile, categorization, and help information, but also many other good features of current Web search engines. In our architecture, we give users the freedom to decide on all the search parameters by defining their own preferences.

The new architecture proposed is shown in Figure1. It is an agent-based approach. It contains a natural language parser, a query customizer, a page retriever, a page filter, a page ordering module, and a user preference agent.

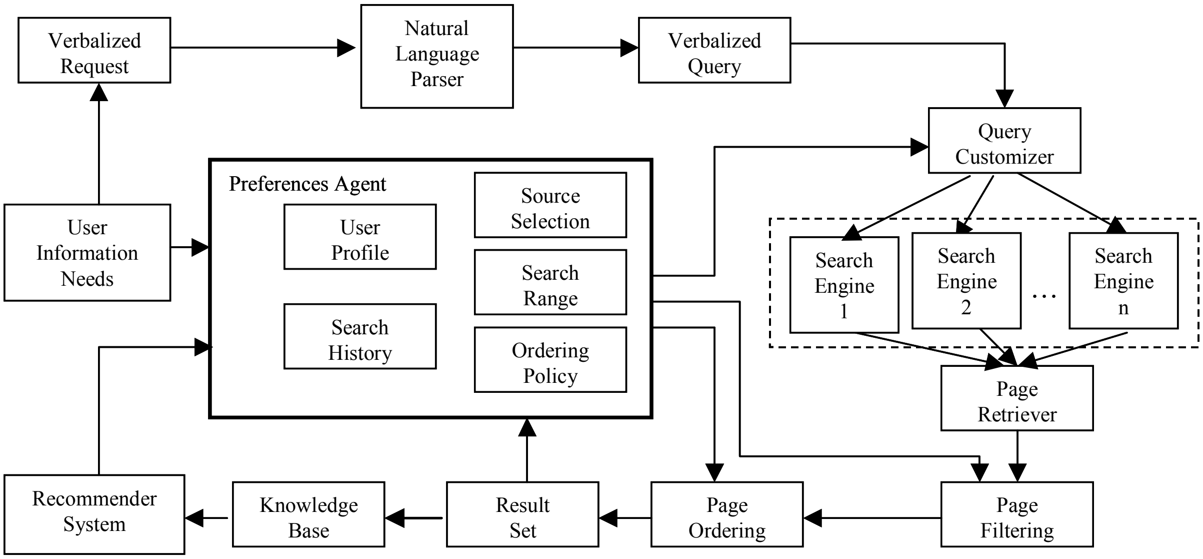

Figure 1. Architecture for Web Meta-Search Engines

A consistent user interface is very important for meta-search engines. From this interface, users can get meaningful results from several different places, while no need to know all about the different search engines used. When users have some information needs, what they have to do is to form their verbalized request, and define their preferences on the search strategy.

Users are allowed to input either keywords or a query in a natural language to describe their requirements. In the user interface, a user can choose any input format -- Either single keyword, several keywords, or natural language. Due to the diversity of the users, some users may not know the exact message that they want. The feature of allowing users to input natural language gives them the possibility to access that information they need. An agent called natural language parser will be able to understand the natural language query. We start with a simple natural language query template and fix the language to be used (e.g. English). The parser first generates an analysis of the phrase structure and parts of speech in the input sentence. It consists of three parts: the segmentation that identifies individual words and some proper names, the tagging that determines the part in the speech information using a hand-coded probabilistic grammar; the phrasing that determines (sometimes overlapping) phrase boundaries based on sentence tags. After the analysis, the information needed will be specified into verbalized query.

In the user preferences part, users can specify the profiles, the sources they want to use, the range of the search, and the ordering policy. Users can change the search preferences at any time to customize the way the search engine searches the Web. Due to the diversity of the users, some default settings are provided for users who may not know how to set different preferences or who may only want simple searches.

Users can update their personal information at any time. In the source selection, users can choose which direct search engines they want to use to run the search from a list of available search engines. And they can choose what kind of source sites they want, such as ".edu", or ".com", or general etc.; or select the source sites from a list of specified sites, such as the sites defined in the bookmarks. For users periodically following certain sort of information, this is convenient.

One of the problems of meta-search engines is their dependence on the underlying search engines to provide a reasonable set of results. As we have mentioned before, meta-search engines can only allow limited number of results to be returned to the user. To enhance the precision of the results, and deal with the problems caused by search engines result limits, our architecture allows users to modify the query by defining the search range – by using search engine specific options, such as sort by date & time period, constrain to a language or geographical areas, specify how many items they want to be returned (number of hits), or how many items they want to be displayed, etc.

In the search range, we allow users to specify which categories they want to search by selecting from a list of pre-defined categories. We also allow users to specify the "type" of the messages they want, for instance, when they specify the "type" as "research paper", we can add constraints on the query with "abstract", "introduction" and references, as most research papers contain those sections.

The selection of the search engine sources, the query modification settings, combined with the verbalized query will form a customized query processed by an agent called query customizer. The customized query will be sent to the underlying search engines. An agent called page retriever will analyze the results from each search engine and pass the results to a page-filtering agent. The constraints of search range, and the sites of search source specifications will form the filtering policy and affect the filtering process. The duplicate and out of range information will be removed by the filtering agent, the "good" pages will be sent to the page-ordering agent.

How to order the results is one of the most important decisions made by a meta-search engine. By using an ordering policy defined by the user's preference agent, different users, even with the same query and the same set of documents, will have results presented in an order meaningful to their individual need. The results can be ordered by time, degree of relevancy, or other indexing criteria chosen by users.

For sorting the results, we will use a dynamic interface that inserts each result as it is scored. As a new result is downloaded and scored, it is immediately available for the user to see, thus improving over many search engines that force users to wait before seeing results.

The reliability and accuracy of results is important to users. The nature of Internet leads to a global set of unmanaged resources. Search engine users should be able to learn what information resources the results are based on, what is the scope of the resources, how reliable are the resources, the age of the information, and the relevancy of the information. The page filtering agent will collect and present these information to users.

A feedback mechanism is adopted in our architecture. When search results are successfully presented to the user, the search history is also recorded in the user preferences and in the system knowledge base. Learning techniques will be used to improve the search selection. A recommander system is running to analyze the patterns of the using sources of a specific user, and user's performance history, thus give users feedback on their search choices.

Help information is always available in the process. And it is customizable. That is, users can choose whether to use it or not.

By allowing users to use flexible query input, and customizable preferences, such as choice of resources they want to use, the capability to modify the query, and specify ordering policy, and the feedback mechanism, our architecture provides more efficient and user-friendly Web based meta-search engines than others.

Summary and Future Work

We have explained the concepts and work mechanisms of Web search engines and meta-search engines, comparing their functions and performance by analysis of several examples, described their limitations and good features, and presented a new meta-search engine architecture, which can improve search engine performances. This architecture uses multi-agents to process user queries, allows greater personalization and higher quality results than a regular meta-search engine, because it gives users more flexibility to help the system make decisions.

After proposing this architecture, our future work will be to implement the new meta-search engine based on the new architecture.

Acknowledgment

Special thanks to professor Murray Turoff and Eli Rohn for inspiring us with some new ideas. Thanks to all IS-seminar participants who help us refine our works.

References

Barker, J. “Meta-search Engines”, *Teaching Library Internet Workshops University of California, Berkeley,* April 2000, http://www.lib.berkeley.edu/TeachingLib/Guides/Internet/MetaSearch.html

Glover, E.J., Lawrence, S., Birmingham, W.P., and Giles, C.L. "Architecture of a metasearch engine that supports user information needs", *Proceedings of the eighth international conference on Information knowledge management*, Kansas City, MO, USA, November 1999, pp. 210 - 216.

Grossan, B. “Search Engines: What they are, how they work, and practical suggestions for getting the most out of them,” February 1997. http://www.webreference.com/content/search

Henzinger, M. “Web Information Retrieval”. At 16th International Conference on Data Engineering, IEEE Computer Society, San Diego, CA, USA February 29 -March 3, 2000, http://www.henzinger.com/monika/

Kingoff, A. “Comparing Internet Search Engines”. *Computer* (30:4), April 1997, pp.117-118.

Lawrence, S. and Giles, C.L. “Accessibility of information on the Web”, *Nature*, Vol 400, July 1999, pp.107-109.

Pollock, A., Hockley, A. “What’s wrong with Internet Searching”, *D-Lib Magazine*, March 1997.

Winship, I. “Web Search Service Features, ” February 2001. http://www.unn.ac.uk/central/isd/features.htm

Yang, M.H., Yang, C.C., and Chung, Y.M., "A Natural Language Processing Based Internet Agent", *IEEE*, 1997, pp.100-105.