To Evaluate the Performance of Metasearch Engines: A Comparative Study

SATINDER BALL¹, RAJENDAR NATH²
¹Assistant Professor, Department of Computer Science and Applications, Vaish College of Engineering, Rohtak-124001, Haryana, India
²Reader, Department of Computer Science and Applications, Kurukshetra University, Kurukshetra-136119, Haryana, India
Email: ¹ satinder_bal@yahoo.com, ² rmath_2k3@rediffmail.com

Abstract:-The explosive growth of information source on the Web and in turn continuing technological progress of searching the information by using relevant tools like search engine poses many problems for the user to know which one is best for their query. At this time metasearch engine comes in to play by reducing the user burden by dispatching queries to multiple search engines in parallel and refining the results of these search engines to the best by doing superior job on their side. These metasearch engines do not own a database of Web pages. They send search terms to the databases maintained by search engine companies, get back results from all the search engines queried and then compile the results to be presented to the user. In this paper, we describe the working of a typical metasearch engine and then present a comparative study of different metasearch engines on the basis of different parameters to show which one is better.

Keywords: Metasearch engine, Search Engine, Query, Web pages, World-Wide-Web.

1. Introduction

With the explosive growth of World-Wide-Web (WWW), publishing document on Internet has become more popular. But how to locate what we need in the ocean of information is an increasingly important and urgent problem. To simplify the problem of getting relevant results based on the search query, the Internet search engines were created that allowed searching a lot of information from the World-Wide-Web in the form of Web pages [1]. Search engines are among the most successful application on the Web today. They act as a system for searching the information available on the Web by automatically searching the contents of other systems and creating a database of the results [2]. The most famous search engines include AltaVista, Infoseek, Google, and MSN. They provide good searching ability by indexing more pages on the Web and maintaining the updated indices in their databases. Despite so many search engines are available to help user in finding the information of their interest, searching on the Web is not an easy task. The problem is due to the vast amount of data on the Web and its rapid updation and growth [3]. The coverage of each search engine is limited. Even the largest and strongest database can index only 5 billion pages [4], which is a fraction of the total Web size. This causes information search to be difficult and irrelevant and some times incomplete information from a single search engine. It points to the importance of searching the Web with multiple engines to find more top ranked results. Obviously it will bring inconvenience and difficulty to the user [5]. This provide basis to the necessity of increasing search coverage via combining results of multiple search engines [3]. Metasearch engines are designed to address these problems [2].
Metasearch engines are powerful tools that send user query simultaneously to several search engines, Web directories and sometimes to deep Web and their databases of Web pages. Within a few seconds, you get back results from all the search engines queried [6]. Metasearch engines do not compile a physical database or catalogue of the Web pages. Instead, they take a user's request, pass it to several other heterogeneous databases and then compile the results in a homogeneous manner based on a specific algorithm and create virtual database [7,8]. Many metasearch engines are available now such as Metacrawler, Savvysearch, Cyber, Dogpile, Profusion, Ixquick, Vivisimo etc.

It was found that it is better to query metasearch engines to obtain the most relevant result. Metasearching is an excellent approach for broad and shallow searches. For keywords of an unfamiliar subject a better way to discover search terms is to see how they appear in a cross section of documents across the Web. Also metasearch engine is an excellent way to get to know about different search engines, their strengths, weaknesses, and types of queries they handle best [6].

Observing the importance and need of metasearch engines, in this paper different metasearch engines are compared on different parameters. As no two metasearch engines are alike, some search only the most popular search engines while others also search lesser-known engines, newsgroups, and other databases. They also differ in how the results are presented and the number of engines that are used. Some metasearch engines list results according to search engine or database. Others return results according to relevance, often concealing which search engine returned which results. This benefits the user by eliminating duplicate hits and grouping the most relevant ones at the top of the list [8].

In our study we surveyed articles; examined test experiments to analyze the quality of the various metasearch engines. From these articles we observed that Dogpile, Vivisimo, Ixquick are the most popular and Metacrawler is the oldest metasearch engine in the industry.

So, for the study, we have chosen these four metasearch engines.

2. Related Work

The Internet grows astronomically in last few years. This gave origin to search engine in 1990. The first search engine was Archie created by Alan Emtage, a student at McGill University in Monteral [9]. In 1992, Veronica appeared on the scene. Soon after another user interface name Jughead appeared with the same purpose as Veronica in 1993. Both of these were used for files sent via Gopher [10, 11].

At this time, there was no World-Wide-Web. The most common way people shared data was via File Transfer Protocol (FTP). In the meantime, Tim Berner Lee created World-Wide-Web. By 1993, the Web was beginning to change. Rather than being populated mainly by FTP sites, Gopher sites, and E-mail servers, Web sites began to proliferate. In response to this change, Matthew Gray introduced World-Wide-Web Wanderer (WWW). ALIWEB was developed, which is the Web page equivalent to Archie and Veronica. The next development in cataloging the Web came late in 1993 with spiders. Like robots, spiders scoured the Web for Web page information. These early versions looked at the titles of the Web pages, the header information, and the URL as a source for key words [9].

The first popular search engine, Excite, was developed in these early days of Web cataloging. It was released for general use in 1994. The first full-text search engine was WebCrawler. Again in 1994, Yahoo! was developed and became the first popular searchable directory. The next search engine to appear on the Web was Lycos. The next major player in the search engine wars was Infoseek. The Infoseek search engine itself was unremarkable and is almost same as WebCrawler and Lycos. What made Infoseek unique was that it becomes Netscape browser's default search engine replacing Yahoo.
By 1995, Digital Equipment Corporation (DEC) introduced AltaVista. This search engine contained some innovations that set it apart from the others. First, it ran on a group of DEC Alpha-based computers. At the time, these were among the most powerful processors in existence. It was also the first to implement the use of Boolean operators (and, or, but, not) to help refine searches. Then HotBot appeared and became the most powerful search engine, by indexing more than 10,000,000 pages.

In 1995, a new type of search engine was introduced - The metasearch engine. The concept was simple. It would get key words from the user either by the user typing key words or a question and then forward the keywords to all of the major search engines. These search engines would send the results back to the metasearch engine and the metasearch engine would format the hits all on one page for concise viewing. The first of these search engines was Metacrawler. There are also other major metasearch engines like ProFusion, Dogpile, Ask Jeeves, and C-Net's Search.com. Ask Jeeves combines many of the features such as natural language queries with the ability to search using several different search engines. C-Net's claims to use over 700 different search engines to obtain its results [9].

3. Working

Searching distributed databases requires three problems to be solved. They include 1) identification and characterization of primary search services whose results are to be merged; 2) selection of a subset of available search engines based on their efficiency or effectiveness; 3) translation of the searcher's request into the relevant query language of each primary search service and getting/parsing results. Metasearch engines can conceptually be dissected into a number of cooperating software components to overcome these problems as shown in Fig. 1 [8, 12, 13]. The functionality served by the various components of the metasearch engine is briefly discussed in the following subsections:

3.1 User Interface

The user interacts with the metasearch engine through user interface. The user submits query to the metasearch engine using this user interface. It enables the user to get the processed results provided by the metasearch engine. This interface provides the user with various advanced features or options supported by different metasearch engines that helps the user to search more specific information. It also provides options for the user to select from the available search engines for his query.

3.2 Database Selector

It is software component that decides which group of search engines will be asked to search a given user query. The goal is to identify as many potentially useful sources as possible while minimizing the inclusion of useless ones. The existing solutions to the database selection problem have been categorized into the following approaches: Rough representative approach, Statistical representative approach, and Learning-based approach [9]. The problem of selecting the information sources to be used were considered in a broader scope by including both search engines and databases, hence the term database selection is most commonly used. The main criterion used for determination of inclusion of Web search engines is the availability of some API to access that search service. Web service providers provide search API’s to enable application developers to leverage the dynamic information generation capability afforded by search engines in the applications that they develop. Metasearch engines will not leave any of the search engines, it is on the user to choose or leave search engine according to their preference and search query.

3.3 Document Selector

This component determines which results shall be retrieved from each of the databases so as to maximize useful documents while minimizing unrelated documents and
network traffic. The criteria used for document selection can be specified by the user, based on statistical-based information, or learning-based information. Document selection, is concerned with the issue of determining how many documents are to be selected and how is the quota distributed among the component search engines. The types of issues that arise in this process are commonly caused by the existence of a large number of component engines to choose from, and that some of them work with more focused datasets [8].

3.4 Query Dispatcher
It establishes connection with each individual search service, forward the query string, and wait to collect the results. This can be implemented by HTTP protocol using GET and POST methods, or via other Web service protocols such as SOAP or REST. The dispatcher connects the metasearch engine to the search engines.

3.5 Result Merger
This component merges the result from different search engines by applying its merging algorithm and then presents the result to the user. It presents the result in two ways. One way is to simply list ten or more results from each engine queried with no additional post-processing. Dogpile works this way. Other metasearch engines analyze the results and then rank them according to their own rules, combining results from multiple engines into a single, unified list.IxQuick, Metacrawler and Vivisimo are examples of this type of result aggregating metasearch engines. When results are retrieved separately from each of the search engines, a single ranked list needs to be assembled for presentation. This is the task of the result merger. The main complexity associated with the merging process is determining how to re-rank the results in the presence and absence of their ranking scores assigned by their respective search engines viewing [14]. Each individual component search engine will return results ranked according to their own ranking algorithm. The only information that can be derived is the relative ordering among the documents in each result. Metasearch engine knows neither the exact ranking algorithm used nor the actual scores associated with each document. Therefore, merging algorithm must be designed to work without either piece of information. Even if the scores are available somehow, the result merger is still faced with the issue of comparability among the scores. Duplicate results also contribute to the complexity of ranking. Some or all of the component search engines in response to a query may return a document. Result merging is concerned with presenting a single consistent result set to the end-user in response to a query. The design of the ranking algorithm is complicated by the fact that a duplicated result can be ordered differently in different search engines. It requires reconciling two or more rankings of the same result [7].

4. Comparative Study
After having given the overview of the Metasearch engine, in this section, a comparative analysis between four most popular Metasearch engines is presented [6,9,10,15,16,17,18]. The parameters that have been identified for comparative study are shown in Table 1. These parameters can act as a measure of effectiveness for any metasearch engine. In the study, we show how each single parameter is crucial in determining the effectiveness of metasearch engines.
Table 1. Parameter List

<table>
<thead>
<tr>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Search Engines Used</td>
</tr>
<tr>
<td>Coverage</td>
</tr>
<tr>
<td>Result Relevancy</td>
</tr>
<tr>
<td>Query Response Time</td>
</tr>
<tr>
<td>Ranking Algorithm</td>
</tr>
<tr>
<td>Advance Display Option</td>
</tr>
<tr>
<td>Advanced Search Options</td>
</tr>
<tr>
<td>International Phone Directory</td>
</tr>
<tr>
<td>Search Images, Mp3</td>
</tr>
<tr>
<td>Language Option</td>
</tr>
<tr>
<td>Approximate Amount of Results</td>
</tr>
</tbody>
</table>

Table 2. shows the comparative analysis done by the authors of this paper. In the following, the results of comparative analysis are discussed point wise.

4.1 Number of Search Engines Used
The number of search engines that are used varies dramatically - the smallest number of search engines referred by a metasearch engine is half a dozen, while the largest number referred is more than one thousand search engines, or database front ends. This factor is not the real indicator of quality of results. The quality of results depend on the search engines that are used (and also the variety) rather than the sheer number. The number of search engines used is an acceptable criterion to use when evaluating the effectiveness of metasearch engines. The more is not necessarily better, but less could certainly be considered worse.

4.2 Coverage
Metasearch engines offer the potential to search a larger portion of the Web as more the search engines used more the Web covered and to some extent in the deep Web. In this parameter, Vivisimo bypass all other three metasearch engines as it uses 35 different search engines which are approximately double or treble the number of search engines used by other three which results in more coverage area for Vivisimo.

4.3 Result Relevancy
Any search service would prove to be beneficial to the user only if it is able to provide relevant and to the point information to the user. Thus, result relevancy can be considered as the most dominating criteria for any information retrieval system. Duplicate results are one factor for irrelevancy. Dogpile and metacrawler might give duplicate results, which cause frustration to the user, but Vivisimo eliminate this by using clustering technique.

4.4 Query Response Time
Query response time is the time taken by any search service in providing the response to the user so as to equip the user with desired information. It is one of the major factors for assessing the performance of any search service. Vivisimo and Metacrawler
equip the user in real time i.e. these two provide the result faster than Dogpile and Ixquick that have no response time limit and can take large time quantum to respond to the user.

4.5 Ranking Algorithm
Metasearch engines are totally dependent upon their ranking algorithm for providing relevant result. Dogpile simply collect the result from different search engines and displays it to the user. Vivisimo collect the result and use clustering algorithm to eliminate duplicate result and for providing best result at the top position. Ixquick use Star rating method for providing relevant results. Metacrawler eliminate duplicates and rank the results by relevance.

4.6 Advance Display Option
Many metasearch engines allow different display modes like brief, normal or verbose. User choice is likely going to depend upon his needs from the search service, and these display modes can be quite helpful in some circumstances. The various other types of display options provided by metasearch engines are like number of results shown per page, number of response from which search engine, rollback option.

4.7 Advanced Search Options
Advance search option enables the users to meet their specific requirement by specifying the importance of each keyword in the query to be submitted. Unlike other three metasearch engine of our study Ixquick doesn't support advance search option. Metasearch engines also provide various features like selecting language in which results to be displayed, feature for searching images, audio file, video file etc.

4.8 International Phone Directory
This option is used to search people, business, and an international reverse phone number. This option is valuable as it provides search for phone numbers all over the world. Only Ixquick provides this option.

4.9 Search Images, MP3
Some Metasearch engines also provide the facility to find Images, pictures or MP3 files. The Vivisimo does not have this facility. The other metasearch engines have this facility.

4.10 Language Option
This is the option through which users can find results in their native language. The Iquick provides 18 languages in which a user can submit query and get results from the search engine. The other metasearch engines have only one option i.e. English.

4.11 Approximate Amount of Results
The results provided by these metasearch engines vary. While Vivisimo provides more than 200 top results received from various search engines with the listing about the number of results taken from each search engine. The other metasearch engines provides less than 100 results received from the various search engines.

5. Result and Conclusions
In this work, four most popular metasearch engines have been studied by the authors and have found that no two metasearch engines are alike. They all differ in the way they handle the query. They also differ in the way they collect, process and display the data. Some metasearch engines list results according to search engine or database. Others return results according to relevance and generally hides the information about, which search engine returned which results. Different metasearch engines provide various advanced search options and features and they vary from one metasearch Engine. Each metasearch engine has strength and weakness.

From the comparative study, it has been observed that Vivisimo has appeared to be a better choice if the larger coverage, high result relevancy, lesser response time are of main concern. When it comes to
optional facilities then Vivisimo lacks behind Ixquick as it provides some of the features like International phone directory, Image and MP3 search which are not provided by Vivisimo. Ixquick stands unique from other three metasearch engines by providing the facility of international phone directory. Dogpile and Metacrawler are also unique in their own strengths.

6. Future Work

We have already developed a metasearch Engine named Snifferidia. The metasearch Engine is already online having URLname “snifferidia.com”. It sends the queries to Two search engines, Google and MSN and Then presents the results to user after re-ranking and eliminating the duplicate results.

The metasearch engine's functionality is crippled by the slow response rate. We need it running much faster in order for it to be usable. Currently, the metasearch engine sallow the user to define the priority of the search engines. It would be much more interesting to determine the priority using some statistical information or learning-based methods. We will investigate and incorporate these modifications in our metasearch engine to address the issues of faster response rate and automatic determination of priority.

We are investigating and developing a metasearch engine based on mobile agents. These mobile agents will take the query to different search engines and collect the results produced by these search engines and after eliminating the duplicate results it will compress them and send it to the metasearch engine for presentation to the user.

References:


### Table 2. Comparison of Various Metasearch engines search

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Dogpile</th>
<th>Vivisimo</th>
<th>Ixquick</th>
<th>Metacrawler</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>No. of Search Engines Used</strong></td>
<td>It uses 14 engines</td>
<td>It uses 35 engines</td>
<td>It uses 10 engines</td>
<td>It uses 5 engines</td>
</tr>
<tr>
<td><strong>Coverage</strong></td>
<td>Searches average portion of the Web</td>
<td>Searches large portion of the Web</td>
<td>Searches average portion of the Web</td>
<td>Searches small portion of the Web</td>
</tr>
<tr>
<td><strong>Result Duplicacy</strong></td>
<td>Might give duplicate results.</td>
<td>Provides no duplicate results.</td>
<td>Provides no duplicate results.</td>
<td>Might give duplicate results.</td>
</tr>
<tr>
<td><strong>Response Time</strong></td>
<td>Large response time.</td>
<td>Display results quickly.</td>
<td>Large response time.</td>
<td>Display results quickly.</td>
</tr>
<tr>
<td><strong>Ranking System</strong></td>
<td>Simply display the result provided by different search engines.</td>
<td>Use clustering algorithm for ranking and to eliminate duplicacy.</td>
<td>Use star rating system for ranking and merging of result.</td>
<td>Eliminate duplicates and rank the results by relevance.</td>
</tr>
<tr>
<td><strong>Varied Display Options</strong></td>
<td>Does not support the varied display options.</td>
<td>Supports the varied display options.</td>
<td>Does not support the varied display options.</td>
<td>Supports the varied display options.</td>
</tr>
<tr>
<td><strong>Advanced Search Options</strong></td>
<td>Supports advanced search options.</td>
<td>Supports advanced search options.</td>
<td>Does not support advanced search options.</td>
<td>Supports advanced search options.</td>
</tr>
<tr>
<td><strong>International Phone Directory.</strong></td>
<td>Does not provide international phone directory.</td>
<td>Does not provide international phone directory.</td>
<td>Provides International phone directory.</td>
<td>Does not provide international phone directory.</td>
</tr>
<tr>
<td><strong>Language Option</strong></td>
<td>Does not provide the option for language.</td>
<td>Does not provide the option for language.</td>
<td>Provides the option for language (18 language).</td>
<td>Does not provide the option for language.</td>
</tr>
<tr>
<td><strong>Approximate Amount of Results</strong></td>
<td>50-70 results</td>
<td>200-250 results</td>
<td>60 - 90 results</td>
<td>50-100 results</td>
</tr>
</tbody>
</table>