Volume 4, Issue 4, April 2014 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

# A Survey on Database Selection in Metasearch Engine

**Kawaljeet Kaur**
M.Tech Student, RGEC Meerut
India

**Richa Bansal**
Asst. Prof., CSE Deptt, RGEC Meerut
India

***Abstract: Search engines are the most useful and high-profile resources on the Internet. A Meta Search Engine acts as an agent for the participant search engine. It receives queries from users and redirects them to one or more of the participant search engines for processing. The various algorithms are used for search engine selection and result merging that provide relevant information according to the user. In this paper, we focus on the technical challenges of meta searching, namely search engine selection, by providing different algorithms.***

***Keywords: Metasearch engine, Personalization, Information retrieval, Algorithm***

## I. INTRODUCTION

Searching the Internet has never been an easy task, even though it is one of the most common tasks performed on the Web. This task is becoming even more difficult with the continued growth of the amount of information posted on the World Wide Web. Not only is there an enormous amount of information, but also the lack of structure makes the search for relevant information a difficult and sometimes very time-consuming procedure. Information Retrieval (IR) systems are software tools that help users find documents contained in a specific corpus or database; these tools are becoming ubiquitous. They are currently used for finding scholarly information as well as for news dissemination, shopping, and many other recreational activities [1].

Search engine is a tool that retrieves web pages that contain information relevant to a specific subject described with a set of keywords given by the user[4]. It is inconvenient and inefficient for an ordinary user to invoke multiple search engines and identify useful documents from the returned results.

To support unified access to multiple search engines, a metasearch engine can be constructed. When a metasearch engine receives a query from a user, it invokes the underlying search engines to retrieve useful information for the user [5]. A MSE (multi-threaded SE) such as Dogpile, Savvy Search, Metacrawler, Profusion and Inquire, is a search tool that sends a query simultaneously to several SEs and consolidates all results, thereby saving time [2].The main goal of Metasearch over the single search engine is increased coverage and a consistent interface to ensure that result from several places can be meaningfully combined [8].

The rest of the paper is organized as: In Section II Web Search engine, In Section III Parts of a search engine, Section IV Meta search engine (MSE), Section V discusses about Components of MSE, Section VI discusses Database selection, Section VII describes Database selection approaches, Section VIII gives Learning based database selection approaches and Section IX present Summary of the work.

## II. WEB SEARCH ENGINE

It is basically a type of program that uses keywords to search for documents that relate to these keywords and then puts the results found in their order of relevance to the topic that was search for.Web search services have very different implementation methods and search strategies from each other. Over the years, search technology has improved significantly. The search plan facilitates user control of parallel searching. Different search engines with different search depths, behaviour, and speed have emerged. However, all the available search engines can be categorized in various architectures:

**Crawler based search engines**

Crawler-based search engines use sophisticated pieces of software called spiders or robots to search and index web pages. These spiders are constantly at work, crawling around the web, locating pages, and taking snapshots of those pages to be cached or stored on the search engine's servers. They are so sophisticated that they can follow links from one page to another and from one site to another. Google is a prominent example of a crawler-based search engine.Yahoo features both directories and crawler-based search engines.

**Directory Search Engine**

**A directory** such as Yahoo [26] depends on humans for its listings. It is similar to newspaper-classified advertisements. The Web author submits a short description of the site to the directory, or editors write descriptions for sites they review. A search looks for matches only in the descriptions submitted.Unlike search engines, which use special software to locate and index sites, directories are compiled and maintained by humans. Directories often consist of a categorized list of links to other sites to which you can add your own site. Yahoo, LookSmart and Encyclopedia Britannica Online are examples of directory search engine.

 

**General Purpose Search Engine**

General-purpose search engines aim at providing the capability to search all pages on the Web. the existence of a large number of unrelated documents in the general-purpose search engine may hinder the retrieval of desired documents.

**Special Search engine**

Special-purpose search engines focus on documents in confined domains such as documents in an organization or in a specific subject area. For documents on the Web, the databases in different special-purpose search engines are natural clusters.

**Hybrid search engine**

Hybrid search engines will present both crawler-based results and human-powered listings. Usually, a hybrid search engine will favor one type of listings over another.

## III. PARTS OF A SEARCH ENGINE

**• Spider, crawler or robot**

The first part of a search engine is called the spider. The spider (sometimes called a crawler or robot) is a program that moves around the World Wide Web visiting websites. It reads the web pages it finds and follows the links further down into the website. The spider returns from time to time and checks for changes. The pages that it finds are placed into the catalog.

**• Index, catalog or database**

The second part of a search engine is called the index, catalog, or database. This index contains a copy of each page that was collected by the spider. A spidered page must be indexed to become a search result.

**• Search engine software**

When a user requests keywords from a search engine, the search engine software sifts through all the indexed pages to find matching keywords, then returns the results/hits to the user.

## IV. META SEARCH ENGINE (MSE)

A meta search engine is a tool that helps to locate information available via the WWW. It provide a single interface that enables users to search many different search engines, indexes and databases. Thus Meta search engines are capable of searching several search engine databases at once. Metasearch engines reduce the user burden by dispatching queries to multiple search engines in parallel [6]. Metasearch engine would collect the result from each engine, after comparing, analyzing, consolidating and deleting the repeat information, finally returns to users with certain format [3]. For each search engine selected by the database selector, the component document selector determines what documents to retrieve from the database of the search engine [1]. The top most documents having higher global similarity in the ranked list are returned to the user through the interface. In this survey, we concentrate on the search of text data. Query format is a list of keywords, called ‘‘terms’’ which provides the semantic to the documents. Ranking of the relevance documents is based on the weight of the query. There are a number of reasons for the development of a metasearch engine and we discuss these reasons below [12].

**Increase the search coverage of the Web:** A recent indicated that the coverage of the Web by individual major general-purpose search engines has been decreasing steadily. By combining the coverages of multiple search engines through a metasearch engine, a much higher percentage of the Web can be searched.

**Solve the scalability of searching the Web:** the problems associated with employing a single general purpose search engine will either disappear or be significantly alleviated. The size of a typical special-purpose search engine is much smaller than that of a major general-purpose search engine. It is also much easier to build the necessary hardware and software infrastructure for a special-purpose search engine. As a result, the metasearch engine approach for searching the entire Web is likely to be significantly more scalable than the centralized general-purpose search engine approach.

**Facilitate the invocation of multiple search engines:** If a metasearch engine on top of these local search engines is built, then the user only needs to submit one query to invoke all local search engines via the metasearch engine. A good metasearch engine can rank the documents returned from different search engines properly.

**Improve the retrieval effectiveness:** Suppose that there is a special-purpose search engine for this subject area and there is also a general-purpose search engine that contains all the documents indexed by the special-purpose search engine in addition to many documents unrelated to this subject area. It is usually true that if the user submits the same query to both of the two search engines, the user is likely to obtain better results from the special-purpose search engine than the general-purpose search engine. This method has been shown to improve the retrieval effectiveness of the system As a result, if for any given query submitted to the metasearch engine, the search can be restricted to only special purpose search engines related to the query, then it is likely that better retrieval effectiveness can be achieved using the metasearch engine than using a general-purpose search engine.

## V. COMPONENTS OF MSE

The architecture of a Metasearch engine, as in [14, 22], is shown in Figure (1) The Metasearch engine consists of four components namely Search Engine Selector, Document Selector, Query Dispatcher and Result Merger.

 

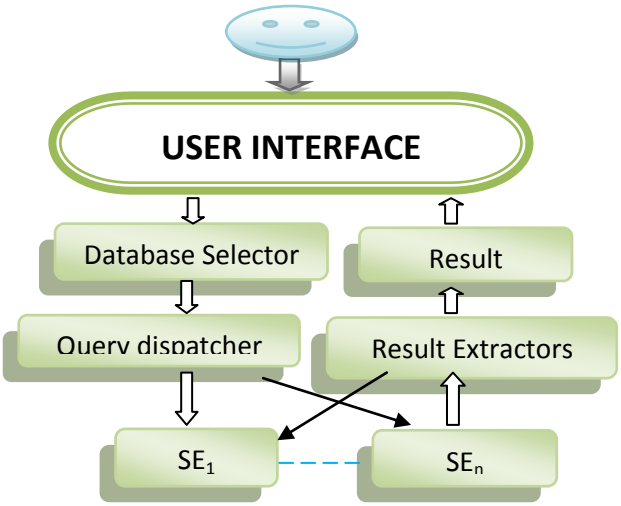


**Fig. 1. MetaSearch software components architecture**

***Search Engine Selector*:** The search engine selector selects the appropriate underlying search engine with respect to the user query. A good search engine selector should correctly identify search engines while minimizing identifying irrelevant search engines. The approaches for selecting search engines are discussed later in this paper.

***Document Selector:*** The document selector determines what documents to retrieve from the selected search engines. The aim is to retrieve more relevant documents with few irrelevant documents. To find out the relevant information different similarity measure is used which estimate the relevance between document and user query. The similarity is measured based on a pre-defined threshold value. The high similarity value shows that the information is more relevant with respect to the user query.

***Query Dispatcher:*** The query dispatcher has a mechanism to establish a connection of a server with each selected search engine in order to dispatch query to each of these search engines. In general, the user query will be sent to the search engine after preprocessing. Every search engine may or may not have the same query as posed on the Metasearch engine.

***Result Merger:*** The result merger merge document retrieved from the selected search engines. The result merger combines all the result into a single ranked list and arranges the documents in descending order with their global similarity with respect to the user query. The topmost documents having higher global similarity in the ranked list are returned to the user through the interface.

## VI. DATABASE SELECTION

To enable search engine selection, some information that can represent the contents of the documents of each component search engine needs to be collected first. Such information for a search engine is called the representative of the search engine [11]. The representatives of all search engines used by the MSE are collected in advance and are stored with the MSE. During search engine selection for a given query, search engines are ranked based on how well their representatives match with the query. Different search engine selection techniques often use different types of representatives [11]. A simple representative of a search engine may contain only a few selected key words or a short description.

## VII. DATABASE SELECTION APPROACHES

The search engine selection approaches can be classified into four categories as in [8].

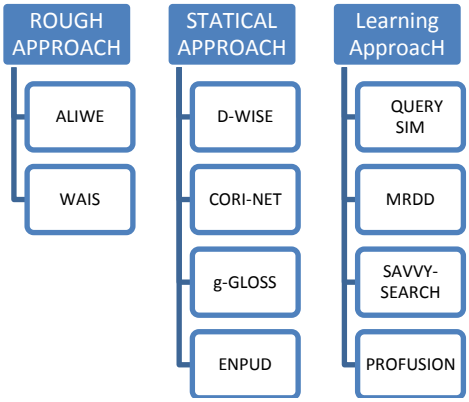


**Fig 2: Search Engine Selection Approaches**

**Rough representative approaches**

In these approaches, the contents of a local database are often represented by a few selected key words or paragraphs. Such a representative is only capable of providing a very general idea on what a database is about, and consequently database selection methods using rough database representatives are not very accurate in estimating the true usefulness of databases with respect to a given query. Rough representatives are often manually generated [5].

 

**Statistical representative approaches:**

These approaches usually represent the contents of a database using rather detailed statistical information. Typically, the representative of a database contains some statistical information for each term in the database such as the *document frequency* of the term and the average weight of the term among all documents that have the term. Detailed statistics allow more accurate estimation of database usefulness with respect to any user query. Scalability of such approaches is an important issue due to the amount of information that needs to be stored for each database[5].

## VIII. LEARNING BASED DATABASE SELECTION APPROACHES

In these approaches, the knowledge about which databases are likely to return useful documents to what types of queries is learned from past retrieval experiences. Such knowledge is then used to determine the usefulness of databases for future queries. The retrieval experiences could be obtained through the use of training queries before the database selection algorithm is put to use and/or through the real user queries while database selection is in active use. The obtained experiences against a database will be saved as the representative of the database.

**Savvy Search Approach**

In SavvySearch Approach [5], the ranking score of a component search engine with respect to a query is computed based on the past retrieval experience of using the terms in the query. Savvy- Search [6] is a metasearch engine employing the dynamic learning approach.

For each search engine, a weight vector ($w_1$…….$w_m$) is maintained by the database selector, where each $w_i$ corresponds to the $i^{th}$ term in the database of the search engine. Initially, all weights are zero. When a query containing term $t_i$ is used to retrieve documents from a component database D, the weight $w_i$ is adjusted according to the retrieval result. If no document is returned by the search engine, the weight is reduced by 1/k, where k is the number of terms in the query. On the other hand, if at least one returned document is read/clicked by the user (no relevance judgment is needed from the user), then the weight is increased by 1/k. Intuitively, a large positive $w_i$ indicates that the database D responded well to term $t_i$ in the past and a large negative $w_i$ indicates that D responded poorly to $t_i$ .

Performance of each search engine in terms of *h*, the average number of documents returned for the most recent five queries, and *r*, the average response time for the most recent five queries sent to the component search engine. If *h* is below a threshold $T_h$ (the default is 1), then a penalty $P_h = \left(\frac{(T_h - h)}{T_h}\right)^2$ for the search engine is computed. Similarly, if the average response time *r* is greater than a threshold $T_r$ (the default is 15 seconds), then a penalty $p_r = \frac{(r - T_r)^2}{r_0 - T_r)^2}$ is computed, where *ro* = 45 (seconds) is the maximum allowed response time before a timeout. For a new query *q* with terms $t_1$,…… , $t_k$, the ranking score of database *D* is computed by

$$r(q, D) = \frac{\sum_{i=1}^{k} wt_i \,.\, \log(N/f_i)}{\sqrt{\sum_{i=1}^{k} |w_i|}} - (p_h + p_r)$$

where log(*N* $f_i$) is the *inverse database frequency weight* of term $t_i$ , *N* is the number of databases, and $f_i$ is the number of databases having a positive weight value for term $t_i$ .

**Profusion Approach**.

ProFusion is a metasearch engine employing the combined learning approach. In ProFusion thirteen preset categories are utilized in the learning process. The 13 categories are "Science and Engineering," "Computer Science," "Travel," "Medical and Biotechnology," "Business and Finance," "Social and Religion," "Society, Law and Government," "Animals and Environment," "History," "Recreation and Entertainment," "Art," "Music," and "Food." A set of terms is associated with each category to reflect the topic of the category. For each category, a set of training queries is identified. The reason for using these categories and dedicated training queries is to learn how well each component database will respond to queries in different categories.

ProFusion combines static learning and dynamic learning, and as a result, overcomes some problems associated with employing static learning or dynamic learning alone. Some short Cummings that the static learning part is still done mostly manually, i.e. selecting training queries and identifying relevant documents are carried out manually. Second, the higher-ranked documents from the same database as the first clicked document will remain as higher-ranked documents after the adjustment of confidence factors although they are of no interest to the user [5].

**QuerySim Approach**

Since there tends to be many similar queries [7] in a real world federated search system, the valuable information of past queries can help us provide better resource selection results. In this section, we propose a novel algorithm, which is called *qSim*, to utilize the valuable information to guide the decision of resource selection. In the algorithm [3, 8] $rel\left(s_j|q\right)$, means it is more appropriate search engine contain more relevant information for the user query. The value of $rel\left(s_j|q\right)$ depends on $rel\left(s_j|p_i\right)$ and $sim\left(p_i|q_i\right)$where $rel\left(s_j|p_i\right)$is the relevance between search engines and past queries and $sim\left(p_i|q_i\right)$is the similarity between all past queries with the user query. The search engines with higher value of $rel\left(s_j|q\right)$being selected by the Metasearch engine.

 

**MRDD Approach**

The MRDD (Modeling Relevant Document Distribution) approach [10] is a static learning approach. During learning, a set of training queries is utilized. Each training query is submitted to every component database. From the returned documents from a database for a given query, all relevant documents are identified and a vector reflecting the distribution of the relevant documents is obtained and stored. Specifically, the vector has the format $<r_1, r_2, : : : , r_s>$, where $r_i$ is a positive integer indicating that $r_i$ top ranked documents must be retrieved from the database in order to obtain *i* relevant documents for the query. With the help of cosine distance similarity function it finds the similarity between user query and all training queries and identifies the k-most similar training query and find the average relevant document distribution vector over k vector corresponding to the k-most similar training queries. Finally, average distribution vector is used to identify the appropriate search engines.

## IX. SUMMARY

This paper provide a study and understanding of database search engine, Meta Search Engines, components of MSE and different approaches of searching database. Our survey seems to focus on the better solutions for learning based database selection approaches.

**References**


[1] DANIEL DREILINGER, ADELE E. HOWE *" Experiences with Selecting Search Engines Using Metasearch"* ACM Transactions on Information Systems, Vol. 15, No. 3, Pages 195–222, July 1997.

[2] MANOJ M AND ELIZABETH JACOB *"Information retrieval on Internet using meta-search engine: a review"* Journal of Scientific & Industrial Research, Vol 67, pp. 739-746October, 2008.

[3] SULEYMAN CETINTAS, LUO SI, HAO YUAN *"Learning from Past Queries for Resource Selection"*ACM CIKM'09, November 2–6, 2009.

[4] H. JADIDOLESLAMY " Search Result Merging and Ranking Strategies in Meta-Search Engines: A Survey" IJCSI International Journal of Computer Science Issues, ISSN (Online): 1694-0814, Vol. 9, Issue 4, No 3, July 2012.

[5] WEIYI MENG , CLEMENT YU , KING-LUP LIU *"Building Efficient and Effective Metasearch Engines"*ACM Computing Surveys, Vol. 34, No. 1, pp. 48–89,March 2002.

[6] Adele E. Howe and Daniel Dreilinger "*A Metasearch Engine That Learns Which Search Engines to Query"* American Association for Artificial Intelligence, AI Magazine Volume 18 Number 2, 1997.

[7] LUO SI AND JAMIE CALLAN, *"Relevant Document Distribution Estimation Method for Resource Selection" SIGIR '03*, July 28-Aug 1, 2003, Toronto, Canada. Copyright ACM, 2003.

[8] R.Kumar, A.K Giri*"Learning Based Approach for Search Engine Selection in Metasearch"* IJEMR Volume-3, Issue-5, ISSN No.: 2250-0758, Pages 82-88, October 2013.

[9] WEIYI MENG , CLEMENT YU , KING-LUP LIU *"Building Efficient and Effective Metasearch Engines"*ACM Computing Surveys, Vol. 34, No. 1, pp. 48–89,March 2002.

[10] G.TOWELL, E.M. VOORHEES, N.K. GUPTA, B.J LAIRD *"Learning Collection Fusion Strategies for Information Retrieval"* Appears in Proceedings of the Twelfth AnnualMachine Learning Conference, Lake Tahoe, July 1995.

[11] HOSSEIN JADIDOLESLAMY *"INTRODUCTION TO METASEARCH ENGINES AND RESULT MERGING STRATEGIES: A SURVEY"* International Journal of Advances in Engineering & Technology, ISSN: 2231-1963, Nov 2011.

[12] YUAN FU-YONG, WANG JIN-DONG*"An Implemented Rank Merging Algorithm for Meta Search Engine"* International Conference on Research Challenges in Computer Science,IEEE, 2009.


 