

Metasearch Engine: A Technology for Information Extraction in Knowledge Computing



P. Vijaya and Satish Chander

Abstract The increasing number of the Web data due to the increased amount of the digitalized standards, electronic mails, images, multimedia, and Web services, the World Wide Web rises as the cost-effective resource for releasing the data and for discovering the knowledge. Thus, the open challenges for the information retrieval efforts have received much attention among the researchers to deem the browsing as an appropriate searching procedure. To facilitate the most relevant information, it is necessary to develop a unique and extensive structural framework that offers a platter and a navigational proxy to the clients and servers. Consequently, the search engines play a vital role in contributing the users in providing the related Web pages from the Web. A Metasearch engine, in essence, is a search mechanism that sends the user query to a number of modern search engines autonomously and provides the combined outcome through their own unique page ranking technique. This chapter intends to discuss the necessity of metasearch engines, starting with a series of definitions of search engines and its classification. Further, a summary of metasearch engine is provided with the architecture and the result-merging methods. It also states several criteria that validate the stability of metasearch engines and, finally, conclude the chapter explaining the future work.

Keywords Knowledge computing · Search engine · Metasearch engine
Page ranking · Result merging

P. Vijaya (✉) · S. Chander
Waljat College of Applied Sciences, P.O Box 197, 124 Rusayl,
Muscat, Sultanate of Oman
e-mail: pvvijaya@gmail.com

S. Chander
e-mail: dimrisatish@gmail.com

1 Introduction

The world, which we live now, has a knowledge repository of information choices—print, spatial, visual, sound, numeric, and so on [1]. Knowledge, which is a facilitator of various capabilities like learning, is transformed into economic value, called knowledge economy, to meet certain organizational goals [2]. Knowledge economy deals with sharing knowledge-based services and products among different users and providers. To facilitate “understanding,” which refers to the process of linking new information and knowledge to the existing one, standardization is required for the knowledge obtained from various information sources. For precise computations, knowledge has to be handled or managed effectively. Knowledge management can be considered as a self-organization that adapts to changing environments depending on dynamic evolution, known as knowledge ecosystem. Enhancement of autonomous behavior of knowledge based on Semantic Web standards is known as executable knowledge, and the corresponding process of management of executable knowledge is termed as knowledge computing [3]. The World Wide Web (abbreviated WWW or the Web) is a knowledge management system. The tremendous growth of the World Wide Web in the last decades has promoted searching [4] as one of the most prominent issues in the field of Web research [5]. The World Wide Web consists of a number of Internet servers that protect the formatted documents designed in a Markup Language known as Hyper Text Markup Language (HTML). Moreover, it sustains the link to several documents, with graphics, audio, and video files.

The Users refer the World Wide Web as “the Web,” that is a segment of the Internet. In general, Web enables the user to jump from one document to another, through a click on the hot spots. However, all the Internet servers need not be necessarily a part of the www family. The size of the Web has grown exponentially since 1990, and it is estimated to contain not less than 14.27 billion Web documents that are publicly accessible and are distributed all over the world via several thousand Web servers. Searching and obtaining relevant information from a massive collection of Web documents is a complex task, as the Web pages are neither managed as books in a library nor cataloged fully at a central location. Hence, there is a need to design an efficient information retrieval approach for obtaining the relevant information from the Web. An important tool used for accessing online information available on the Web is a search engine. It is a synchronized set of programs, which can access every searchable Web page, by creating an information index that is compared to the query requested by the user, thereby returning the results to the user [6].

According to the studies made by American Life Project, the search engine use has become a regular online activity. The increasing popularity and the influence of search engines generate questions regarding their role in determining the global information order [7]. The word “search” refers to the attempt of finding something, which is either existing or novel [7]. The search engine processes the keyword query received, providing a list of Web pages, called Search Engine Results Page

(SERP) [8]. The results of the search engines are ranked displaying the most relevant results first. In the case of generating irrelevant results, the users must filter their queries to discover the required information [9].

This chapter deals with the basic concepts of knowledge extraction using metasearch engines by discussing the definitions of search engines and its classifications and thereby explains the architecture of metasearch engine [10, 11]. Moreover, it summarizes the ranking concepts that the metasearch engines perform by describing the result-merging techniques. Then, the validation criteria regarding the stability of the metasearch engines are explained with respect to the user queries to show the effectiveness of the metasearch engines. The organization of the chapter is as follows: Sect. 2 provides few definitions of search engines stated by the researchers. The classification of search engine is explained in Sect. 3, along with the architecture of metasearch engine, describing each of its components. In the same section, different merging methods employed in the metasearch engine are demonstrated. Section 4 shows the development criteria, while the conditions to validate the queries are explained in Sect. 5. Common issues to be addressed are discussed in Sect. 6, and Sect. 7 provides the conclusion and the future work.

2 Definitions of Search Engines

Search engines are effective in identifying keywords, quotes, phrases, and information that are available in the entire content of Web pages. Search engines permit the user to search the keywords that are stored in a huge database, which can be retrieved when needed. Some of the definitions of search engines explained by the researchers are as follows:

According to [12], a “World Wide Web search engine” is termed as “a retrieval service, consisting of a database (or databases) describing mainly resources available on the World Wide Web (WWW), search software, and a user interface also available via WWW.”

According to [13], a search engine is referred as “a program that is accessible by any average user, capable of accepting user input which defines the information it produces as output to this user.”

According to [7], a domain-specific search engine is “an information access system that allows access to all the information on the Web that is relevant to a particular domain.”

According to [9] a search engine is stated as, “a search tool that allows one to find specific documents through keyword searches and menu choices, in contrast to directories, which are list of Web sites classified by topic.”

3 Classifications of Search Engines

The classification of search engines can be made in many ways. Herein, the classification is done on the basis of two concepts: working principle and properties. Based on the working principle, the search engines can be categorized into three, as follows.

3.1 *Based on Working Principle*

Crawler-based Search Engines: In crawler-based search engines [6], the listings for the index or catalog are created automatically using a software tool, named “Web crawlers” or “spiders,” by sending the spiders to crawl across the Web, following the links on one Web page to the next. Here, the search engine utilizes a computer algorithm for the ranking of all the retrieved pages. These search engines are massive, retrieving a large volume of information from the Web. The search engine is efficient in providing relevant information for the user who has a specific search topic [6]. For general searches, the crawler-based search engines may provide irrelevant responses, where the keyword may be found only once in the document [13]. It permits the user to search based on the results of the previous search for complicated searches, enabling the user to filter the search results. This kind of search engine has the complete text of Web pages which are connected so that the user can obtain the pages by matching the words in the desired pages [6]. One of the main challenges in this search engine is the dynamic nature of the Web, where it requires keeping the database up-to-date [9]. Google, Teoma, Altavista, and Lycos are some of the crawler-based search engines.

Human-powered directories: These are based on human editors who create listings to construct the directories. Human-powered directories [6] are structured into subject categories, and the pages are classified based on the subjects. They do not hold full text of the Web page to the link they are connected. The Webmasters provide a short description and the URL to the directory for their Web sites. These descriptions edited manually constitute the search base. Hence, a change in individual Web page will not affect the pages listed in the results. The directories are capable of providing much more relevant results than that offered by the search engines. During the search, the directory site focuses only on the matches that are within the descriptions submitted, rather than the information on the Web pages. The user has to submit an online update to the Webmaster of the search engine to update the description of the Web site. Even though the search topic is relevant and accurate, this is not efficient for searching when the user has a specific search topic in his mind. Few human-powered directories are MSNSearch, Yahoo, AskJeeves, Look Smart, and Open Directory.

Metasearch engines: Search engines [13] can filter the pages matching explicit queries effectively. However, they require huge memory resources to store the Web index and extreme network bandwidth to construct and refresh the index. As they receive millions of queries a day, the CPU cycles dedicated to satisfy each query are curtailed. This result in information overload that the mandatory intelligence fails to combat at most of the time [1]. To overcome the issues of search engines and to solve the information overload problem over the Web, metasearch engines are introduced [1]. It is likely to attain better results by submitting a query to various search engines, as their databases do not overlap completely. Metasearch engines, such as Mamma, Metacrawler, and Dogpile, perform this automatically. Actually, metasearch engines do not own a database; instead, they submit the query automatically to multiple conventional search engines and obtain the results [14]. Thus, it is possible to acquire more responses, as the combined coverage of multiple search engines is appreciably larger than the result of any single search engine [9]. A searching tool that utilizes the results of other search engines, irrespective of user's preferences, is known as a metasearch engine [1].

A Metasearch engine [15] characterizes an in order retrieval agent which is configured at the apex of parallel search engines. The Metasearch engine receives the query that forwards the queries to a number of individual search engines in parallel. Consequently, the Metasearch engine integrates the results from all the search engines. The numerous models of the Metasearch engines (MSE) are the Metacrawler, Profusion, Savvy search, and the MetaSEEk. There are a lot of challenges in dealing with the Metasearch engine. In contrary, the combined result from various search engines is the hectic challenge in utilizing the Metasearch engine. Upon the arrival of the specific query, the individual principal search engine will re-examine the result, which reasonably leads to a subset of the ultimate post-processed upshot of a Metasearch engine. In case there are a large number of hits of a search engine, the users check only the top-listed documents of the results, and the order of documents in the ultimate outcome assumes added significance.

3.2 Based on Properties Included

The above three categories of search engines have three common properties, such as general, personalization-based, and semantic-based. A hierarchical diagram is presented in Fig. 1, regarding the categorization of the search engine.

General search engines: General search engines are those search engines intended to search any general search topic. Google, Bing, and Yahoo are general search engines, where images, maps, news, and much more, can be searched. Specialized keywords and symbols can also be used in general search engines to obtain specific information. These search engines are efficient in finding Web sites with relevant information.

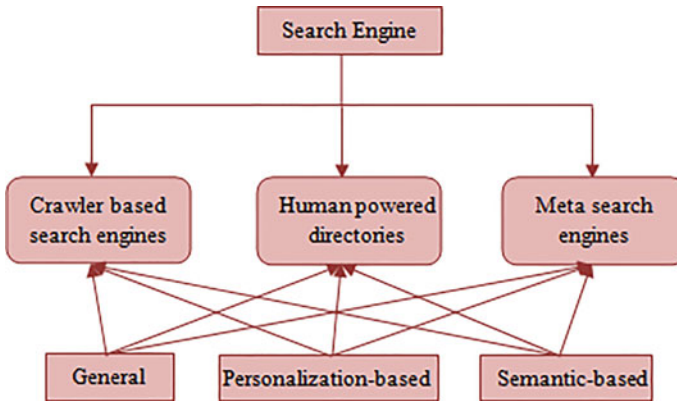


Fig. 1 Hierarchy of search engine

Personalization-based search engines: [16] Personalization-based search engines track the search history of a registered user and then regulate the search results according to the preferences of the users. For example, a person searching for “jaguar” will get the results for the animal if his/her past searches were based on the information regarding animals, and car if the past searches were related to automobiles. The results are obtained taking the benefits of information on the Web sites based on the analysis of the behavior of the user in association with the data taken from the Web context. Some of the benefits of personalization-based search engines are as follows: Personalized search can save the user’s time by avoiding the repetitive search. The search engine eliminates the tasks that are no longer required from the search results. Thus, the search engine can offer an efficient and faster means to obtain the results as per the requirements of the user. Personalization-based search engines provide the user more satisfactory and accurate result.

Semantic-based search engines: The search process of the conventional search engines is improved based on semantic search. Hence, the researchers try to develop metasearch engines based on the idea of including the semantic information. This kind of search includes the context of search, intent, variation of words, specialized queries, location, and so on, to offer relevant results [1]. Major Web search engines, such as Google and Yahoo, incorporate few semantic search elements to enhance the search. LinkedIn, which is a social networking service, provides semantic search for searching the job to recognize and standardize the entities in terms of queries and documents, such as companies, skills. Depending on the entities, several entity-aware features are constructed. This search engine is aware of the context being searched and thereby provides smart and relevant results based on the queries requested. Earlier, the search engines use page ranking approach to provide ranking to the particular link for the relevant search. Conversely, a semantic search engine utilizes ontology for more meaningful and precise search results to be retrieved in minimum time. It ensures that most of the

relevant results are returned based on a word’s meaning and relations, rather than a specific keyword. The search engine maintains semantic identification regarding Web resources in a way to solve complex queries. By integrating technologies related to Semantic Web into the search engine, semantic offers improved search results, evolving future generation of search engines that are built on Semantic Web.

3.3 Architecture of Metasearch Engine

Metasearch engines are constructed on other search engines without collaborating with these search engines explicitly. Even though designing the programs may not be a difficult task for an experienced programmer, preserving their validity can be a serious issue as the connection parameters and the format of result page visualization of the search engines may change. Moreover, in the case of applications, which require connecting several thousand search engines, maintaining these programs is costly and time-consuming. Figure 2 shows the architecture of a typical metasearch engine.

The components of the metasearch engines are explained as follows,

Database Selector: Identifying potentially useful databases for a given query is important in metasearch engines. This is the responsibility of the software component database selector. Selecting local databases makes a search engine useless, as it is wastage of effort.

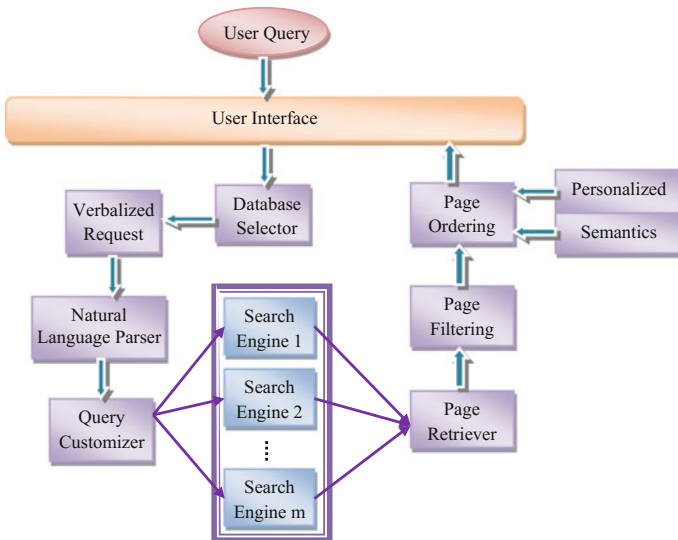


Fig. 2 Architecture of metasearch engine

Natural Language Parser: The users who have information needs create verbalized requests, define what they prefer on the search strategy. They can input their requests either as keywords or a query in a natural language to explain their requirements. The user can select any input format using the user interface. Hence, the user who is unaware of the exact message can access the desired information, with the utilization of natural language parser that has the ability to recognize the natural language query. Thus, the information required will be formatted into a verbalized query.

Query Customizer: Query customizer is an agent that progresses the customized query that is determined by the fusing the process of selecting the search engine, query modification settings, and the verbalized query. To facilitate search engine selection process, the information having the document contents of individual component search engine is to be collected. This informative content of a search engine is termed as the search engine representative. The metasearch engine stores the representatives of all the collected search engines in advance. To select a search engine for a particular query, they are ranked according to the matching of the representatives with the query. Various search engine selection methods are being developed using various types of representatives.

Page Retriever: The function of a page retriever is to analyze the search results obtained from every search engine and forward the results to the succeeding agent, i.e., page-filtering agent.

Page Filtering: The users of search engine must learn few important aspects, such as the selected information resources based on which the results are retrieved, the reliability of the resources, the scope of the resources, information relevancy, and the time when the information is taken. The information gathered by the page-filtering agent is send to the users of the search engine. Moreover, the filtering agent eliminates the out of range and irrelevant information, sending the “good” pages to the following page-ordering agent.

Page-Ordering: Ordering the search results is one of the crucial decisions to be made by the Metasearch engine. Page ordering is also known as result merging, where the search results obtained from individual search engines are combined into a single list of rank. Most of the existing search engines used a numerical matching or similarity score in every search result returned, and the techniques for result merging were designed in such a way to “normalize” the scores obtained from various search engines. One of the metasearch engines that employ this technique is Inquirus. Offering a regular way to calculate the ranking scores is the key advantage of this technique such that the ranking provided makes sense. However, it has a limitation due to the delay produced while downloading and analyzing the result instantly, which lead to longer response time. Recent search engines visualize the title of every result returned with a brief summary, known as snippet. These two features of a result can offer better clues suggesting whether the result is relevant to a query or not. Various factors, such as the number of unique terms of the query appearing in the title or snippet and the proximity of those terms appearing in title or snippet, are used to measure a matching score, when the result merging is based on titles and snippets. Even though multiple search engines return same results, they

probably tend to be relevant to the user query. This is due to the fact that most of the ranking techniques attempt to offer the same set of relevant results, but various sets of irrelevant results [6]. Adding the ranking scores of the results obtained from several search engines, these results can be ranked higher in the rank-merged list so as to compute the absolute score for the search result.

Different learning methods can be used to enhance the search selection. Moreover, metasearch engines utilize a recommender system to analyze the patterns of the resources of a particular user, and his/her search history, so that the search engine can include user's preferences on their search choices. The process often makes the availability of help information, which is customizable, such that the users can make use of it when desired. Hence, the Metasearch engines are undoubtedly efficient and easily accessible allowing the users to input flexible queries. The customizable preferences like adaptability in using user-desired resources, ability to update the query, providing better ordering policy and the feedback mechanism, add to the benefits of metasearch engines than general search engines.

3.4 Merging Methods for Metasearch Engines

Several merging techniques utilized to order or merge the results of the search engines are summarized in this section, to provide better retrieval of search results for knowledge computing. The process of result merging refers to fusing of the results generated from a number of the search engines into a single ranked list. Following are few commonly used methods for ranking the search results:

- (i) Fusing the outcome of the multiple search engines by normalizing the scores into values within a common range with an objective to make them better so that the results obtained from valuable search engines are to be ranked higher.
- (ii) Downloading all the retrieved documents from their local servers and utilizing a similarity function to find their matching scores of the search engines used by the metasearch engine.
- (iii) Employing voting-based approaches.
- (iv) Adopting techniques that depend on titles, snippets and other similar features.

Based on the above concepts, various result-merging techniques [17, 18] employed in metasearch engines are described in detail as follows.

Let $E = \{E_i\}$, $1 \leq i \leq n$ represents the search engines present in a metasearch engine, where n denotes the selected number of search engines. Each search engine retrieves a Web page, W . Hence, the Web page retrieved by the i th search engine is represented as, W_{ij} ; $1 \leq j \leq m$, where m indicates the total number of Web pages recovered by the i th search engine. Let R_k be the ranked Web page of a metasearch

engine, where $k = 1, 2, \dots, p$, such that $p = m * n$ corresponds to the number of unique Web pages obtained in the result. Consider $Q = \{q^t\}; 1 \leq t \leq r$ represents the query, where q_t denotes a query term and r refers to the total number of query terms.

Simple merge algorithm: Simple merge algorithm is actually a metasearch engine, which collects the results from other search engines based on a multi-way merge approach. In realistic, users often prefer the first three pages of the search results. The remaining pages could not enhance user experience, rather could only provide completeness to the search results. This simple merge algorithm provides the results of all the search engine by unifying based on the search ranks. Therefore, the merge result provided by the simple merge algorithm is,

$$R^{SM} = \bigcup_{j=1}^m \bigcup_{i=1}^n W_{ij} \quad (1)$$

where R^{SM} is the merge result of the metasearch engine. W_{ij} is the j th Web page of i th search engine.

Abstract merge algorithm: Abstract merge algorithm is based on ranking, where the search results are ranked according to the relevance between the abstract information or snippet and the query of the results. Initially, the key terms are extracted from the query, and the relevance between the terms and the snippet are calculated. Then, the relevance between the query and the page is computed individually for each page. Based on the relevance calculated, the search results are returned to the users. The relevance between the query and the snippet is calculated as follows,

$$R^{AM}(W_k) = \sum_{t=1}^r \ln \left(\frac{l(s)}{\text{loc}(q^t, s)} \right) \quad (2)$$

where $R^{AM}(W_k)$ is the ranking of k th unique page in the search engine results by the metasearch engine. $l(s)$ denotes the length of the snippet k th unique page, which is denoted as s and $\text{loc}(q_{ij}^t, s)$ indicates the location of i th query term in the snippet.

Position merge algorithm: In position merge algorithm, the objective is to utilize the original information regarding the position entirely from each individual search engine. Few pages may occur in many result lists of various search engines for the same query, but their position may differ for each result list. As a solution to this contradiction, the position of the pages in each search engine is taken into account. For n search engines in the metasearch engine, the rank is obtained based on the position of the Web pages in the result of the search engine as,

$$R^{PM}(W_k) = \sum_{i=1}^n \left[\frac{1}{X(W_{ik}, S_i)} \times n - i + 1 \right] \quad (3)$$

where S_i shows the outcome of the i th search engine and $X(W_{ik}, S_i)$ means the position of W_{ik} in S_i . After the calculation of the relevant results between the query and each result record of the search engine, the outcomes are organized in descending order, returning the results to the users in the HTML format without overlapping.

Abstract/position merge algorithm: In abstract/position merge algorithm, the two aforementioned factors, namely abstract and position, are considered synthetically to return an integrated results to satisfy the needs of the user. The algorithm combines the ranking score of both abstract merge and position merge algorithm to obtain the final score, based on which the results are merged. Abstract/position merge algorithm helps in re-ranking the results retrieved from the search engine.

$$R^{AM}(W_k) = A * R^{AM}(W_k) + B * R^{PM}(W_k)$$

where A is the weight value of the abstract merging algorithm, and B is the weight value of the position merging algorithm.

Take the Best Rank: This algorithm intends to assign the score values of a Web page by choosing the best rank obtained among the search engine rankings, avoiding the clashes with the aid of search engine popularity.

$$R^{BR}(W_k) = \min\{R_1(W_{1k}), R_2(W_{2k}), \dots, R_i(W_{ik}), \dots, R_n(W_{nk})\} \quad (4)$$

where $R_i(W_{ik})$ indicates the rank of i th search engine for the Web page of W_k .

Borda's Positional Method: Here, the method computes the $L1$ -Norm of the ranks for the results obtained in various search engines to find the MetaRank of a Web page. Search engine popularity is adopted to avoid the clashes. The procedure to compute the score of a Web page based on this method is the average of the Web page ranking given by all the search engines.

$$R^{BR}(W_k) = \frac{R_1(W_{1k}) + R_2(W_{2k}) + \dots + R_i(W_{ik}) + \dots + R_n(W_{nk})}{n} \quad (5)$$

Weighted Borda-Fuse: In Weighted Borda-Fuse algorithm, the assumption of the search engines is that they are unequal; instead, their votes are computed along with the weights assigned. The vote computation is based on the reliability of the individual search engines. The users can assign the weights that are included to compute the votes, in their profiles. Hence, the votes obtained for the results rendered by i th search engine are given by,

$$R^{WB}(W_k) = A_i * \left(\max_i^n S_i - j + 1 \right) \quad (6)$$

where A_i corresponds to the weight of the i th search engine and S_i is the outcome of the i th search engine.

The Original KE Algorithm: KE Algorithm is a score-based approach utilizing the ranking result of the search engines and the number of its occurrences in the listings of the search engine. Since all the component engines are considered to be equal, they are treated as reliable. Equation (7) indicates the formula for ranking the retrieved results.

$$R^{KE}(W_k) = \frac{R_1(W_{1k}) + R_2(W_{2k}) + \dots + R_i(W_{ik}) + \dots + R_n(W_{nk})}{(c * n * (r/10 + 1) * c)} \quad (7)$$

where c represents the total number of search engines retrieved the Web page W_k , n denotes the specified number of search engines, and r represents the number of ranked outcome taken from every search engine by the algorithm. Hence, it is obvious that smaller the weight of a result score better is the ranking the result obtains.

Borda Count: This is a voting-based technique used for the fusion of the data. In Borda count, the results obtained are assumed as the candidates, while each search engine is considered as a voter. The candidates in the top rank list are assigned “ c ” points; for each voter, the candidates in the second top rank list are given “ $c - 1$ ” points, and so on. The candidates, which a voter did not rank, are those not returned by the associated search engine. Therefore, the remaining voter points are equally split among the candidates and the ranking of the candidates is done based on the received points.

Merging Based on Combination Documents Records (SRRs): The effective method among all the available merging techniques is the method developed by combining the document evidences, such as search engine usefulness, title, and snippet. In this method, the global similarity is computed by measuring the similarity between the query and its title and its snippet, which is merged linearly for each document. The weight is estimated for each query term in the component search engine using Okapi probabilistic model. The weights of the query term of the search engine are added to get the search engine score. At last, the relative deviation of the score of its source search engine is multiplied with the mean of all the search engine scores obtained, to adjust the global similarity of each result. For a given query, this method also has the possibility of returning the same document from different component search engines. In such case, the method combines their ranking scores. Several linear combination fusion functions have been designed to tackle this issue with the inclusion of min, max, average, sum, and so on.

TopD—Use Top Document to Compute Search Engine Score: TopD is an algorithm that merges the result based on the similarity between the query and the top-ranked document retrieved from a search engine. Taking the top-ranked document from the local server introduces a delay, which is negligible, as it requires fetching of only one Web page from the search engine. To estimate the similarity, two functions, namely Cosine function and Okapi function, are used. The similarity estimation using Okapi function [19] is given as follows,

$$\text{ST}(W_k, Q) = \sum_{t=1}^r W^o * \frac{(d_1 + 1) * f}{D + f} * \frac{(d_3 + 1) * q_f^t}{d_3 + q_f^t} \quad (8)$$

where $\text{ST}(W_k, Q)$ is the similarity between the Web page and query based on top results, W^o is the Okapi weight, given by, $W^o = \log \frac{b-b^t+0.5}{b^t+0.5}$, and $D = d_1 * \left((1 - \alpha) + \alpha * \frac{g}{g_{\text{avg}}} \right)$. f is the frequency of the query term u^T , q_f^t is the frequency of u^T within the query Q , b is the number of Web pages, b^T is the number of Web pages with q^t , g is the length of the Web page, g_{avg} is the average length of the Web page, and $d_1 = 1.2$, $d_3 = 1000$, and $\alpha = 0.75$ are the constants. This algorithm offers a ranking scores 1 for the top-ranked results obtained from all the search engines. For a Web page obtained from several search engines, the final ranking score is estimated by adding all the ranking scores.

TopSRR—Use Top Search Result Records (SRRs) to Compute Search Engine Score: TopSRR algorithm computes the search engine score based on the SRRs of the top results retrieved from each search engine, rather than the top-ranked Web pages. This makes the algorithm sensible as, for a given query, a useful search engine probably provides better results that are revealed in their SRRs. The algorithm merges all the titles of the top SRRs from each search engine to create a title vector and the snippets into a snippet vector. Following equation gives the formula of the search engine score obtained by combining the similarities between a query and a title vector, and between the query and the snippet vector,

$$\text{SRR}(W_k, Q) = P_1 * \text{Sim}(Q, V_i^T) + (1 - P_1) * \text{Sim}(Q, V_i^S) \quad (9)$$

where $\text{SRR}(W_k, Q)$ is the similarity between the Web page and query based on title and snippet, P_1 is a constant that takes a value 0.5, V_i^T is the title vector and V_i^S is the snippet vector, $\text{Sim}(Q, V_i^T)$ is a function to compute the similarity between the query and title vector.

SRRsim—Compute Simple Similarities between SRRs and Query: As each SRR is treated as the representative of the corresponding document, it is possible to rank SRRs obtained from multiple search engines. SRRsim algorithm [19] measures the similarity between a SRR and a query, defined as a weighted sum of the similarity between the title of SRR and the query and the similarity between the snippet of SRR and the query, as given below,

$$\text{SNT}(W_k, Q) = P_2 * \text{Sim}(Q, T^w) + (1 - P_2) * \text{Sim}(Q, S^w) \quad (10)$$

where $\text{SNT}(W_k, Q)$ is the similarity between the Web page and query based on title and snippet, $P_2 = 0.5$, T^w is the title of W_k , and S^w is the snippet of W_k . For a document with different SRR returned from several search engines, the similarity is measured between the SRR and the query and the maximum similarity will be taken as the final similarity value for ranking.

SRRRank—Rank SRRs Using More Features: The similarity measure designed in the SRRsim algorithm is not suitable in revealing the true matches of the SRRs for a given query. This measure did not consider either the proximity information, like the closeness of the query terms in the title and snippet of a SRR, or the order in which the query terms appear in the title and snippet. SRRRank algorithm overcomes those drawbacks of SRRsim algorithm by considering both the order and the proximity information, as they have a considerable impact on matching the phrases. The consideration of the information is based on five different features [20] defined based on the query terms as follows.

- N : The number of distinct query terms appears in the title and the snippet.
- N^T : Total number of times the query terms appears in the title and the snippet.
- X : The locations of the occurred query terms.
- A : Checking the occurred query terms if they are placed either in the same order as specified in the query or in a different order.
- w^s : The size of the window with distinct occurred query terms.

The information regarding these five features is obtained for every SRR of the result generated. The procedure involved in the SRRRank algorithm is given in the following steps:

- Initially, the algorithm groups all the SRRs depending on N and the groups that have more distinct terms than the other groups will be ranked higher.
- The SRRs in each group are grouped further to form three subgroups based on the feature regarding the position, denoted as X .
- Then, ranking is done by offering the highest rank to the SRRs having large N^T appearing in the title and the snippet within each subgroup. When two SRRs are with the same number of occurrences of query terms, the SRR with distinct query terms placed in the same order as they are given in the query will be ranked higher, after which, the one with least window size will be ranked. In the merged list, the result having the higher local rank is given a higher global rank. When a result is returned from several search engines, the one having the highest global rank will be kept.

SRRSimMF—Similarities between SRRs and Query Using More Features: SRRSimMF has a similar procedure of SRRRank, but differ in a way by quantifying the matches depending on the features identified in SRRRank such that the scores are merged into a numeric value. In the specified field of a SRR, S , for the given N , the matching score is given by the ratio of N to the total number of distinct terms in the query, denoted as N^D ,

$$C_N = \frac{N}{N^D} \quad (11)$$

For the N^T given, its matching score is computed as the ratio of N^T to the length of title, as

$$C_{N^T} = \frac{N^T}{g^T} \quad (12)$$

where g^T is the length of the title. Based on the order of the query terms and A , the matching score, $C_A = 1$, if the distinct query terms are in the same order, adjacently in the title, and $C_A = 0$, for the other case. The score obtained using the w^S of the distinct query terms in the title is $C_{w^S} = \frac{w^S}{g^T}$. By merging all the scores of these features into a single value, the similarity between the title and the query is given by,

$$\text{Sim}(T^S, Q) = C_N + (1/N^D) * (w_1 * C_A + w_2 * C_{w^S} + w_3 * C_{N^T}) \quad (13)$$

where w_1 , w_2 , and w_3 are the constants, which take values 0, 0.14, and 0.41, respectively. Hence, the final similarity is computed as,

$$\text{Sim} = \frac{N}{g^T} * (P_3 * \text{Sim}(T^S, Q) + (1 - P_3) * \text{Sim}(s, Q)) \quad (14)$$

where N is the total number of distinct query terms in title and snippet, g^T is the length of the query terms, $\text{Sim}(s, Q)$ is the similarity between the snippet and the query, and $P_3 = 0.2$.

4 Development Criteria of Metasearch Engine

Various criteria that tend to become a part of the development of metasearch engine are explained as follows.

4.1 Selection of Search Engine

Metasearch engine requires multiple search engines and its interfaces to connect with a new Metasearch engine. The result of the Metasearch engine completely depends on the results of the search engines which we have considered. So, the right selection of the multiple search engines plays a major role in the Metasearch engine. An ideal metasearch engine performs database selection, to recognize the most appropriate component search engines such that they access a given user query. A metasearch engine with a huge number of component search engines makes it inefficient, as it requires sending the user query to every component search

engine, which in turn inquires additional time to access each search engine and to process the user requests. Hence, for such cases, database selection should be carried out. A perfect database selector must be capable of correctly identifying all the possible databases that are potentially useful in such a way the possibility of identifying a wrong database is considerably reduced. The search engine selection is depending on the value of relevance computed between the user query and the search engine. Generally, database selection is performed based on the collected information that represents the main content of each search engine [20]. On receiving a user query, the usefulness of a search engine is measured using the representative information based on the query [17]. This helps in determining whether a search engine can be selected to process the particular query. The usefulness of the search engine is termed as search engine score [20]. However, there occurs a problem called database selection problem in metasearch engine. The difficulty regarding the identification of potentially useful databases for searching a particular query is termed as the database selection problem [17]. Hence, selecting suitable search engines to process the user query by handling database selection problem is a necessity and a challenging task.

4.2 Query Reformation for Search Results

Query applying through the search engine is highly sensitive to the results of the Metasearch engine. So, a user query can be reformulated into meaningful keywords to obtain the relevant information from the search engine. It would enhance the performance of the Metasearch engine. The process of modifying the original query based on its similarity to enhance the effectiveness of the search is known as query reformulation. This process requires the evaluation of a user's input, extending the search query to match similar documents. Add or remove words, acronym expansion, and word substitution are few query reformulation techniques. Earlier query reformation techniques mainly concentrated on the way of extracting useful patterns and ranking of the candidates with the useful patterns, whereas the candidate generation models are simple. This method through the usage of the word-by-word transformation can make the string transformation accurate and efficient. When a user inputs the query, the system returns all possible queries based on the original query. From the generated list of matching queries, the technique selects the top query that the user prefers [18]. Query reformation can also be done based on the keywords and their similarities in the query. The idea behind the usage of keywords in query reformation is to offer better results, instead of sending the original query to the search engine. Moreover, this can avoid the generation of more irrelevant results from the search engine to improve the effectiveness of retrieval [21].

The process of query reformation can be described as follows: For a given input query " u ," the most probable output queries are generated following a transformation based on a set of operators included in a chain. The input queries can be

sentences, strings of words, character, and so on. Every operator in the chain is considered as a transformation rule that replaces a query with another query with similar meaning. Query reformulation aims at solving the issue of maturity mismatch. This can be understood clearly from the given example. If a user submits a query “apjabdulkalam” to the search engine and the document available is only “Famous books by Dr. APJ Abdul Kalam,” retrieving results seem to be a difficult task, as the query and the document do not match and the document is not highly classified. In fact, the query reformulation technique tends to transform “apjabdulkalam” into the “Famous books by Dr. APJ Abdul Kalam,” and thus, form a better match between the document and the query. This also includes writing the original query or its similar words to match the original query or the similar words in the dictionary so that the search efficiency can be improved [18].

4.3 Bringing Semantic Richness

Before integrating the search results, the chief problem to be addressed is how to obtain the semantic results from the user query or reformulated query even though the intent of keywords is not presented in the user query. In semantic information retrieval applications, keyword similarity is not much useful, because of the difficulty in ensuring the user intention. In addition to keyword similarity measure, semantic similarity measure must be counted in for the information retrieval based on the semantics. Many interesting researches have been developed in semantic search associated with Web mining and ontologies like ODIX platform. ODIX is a metasearch engine offering search results that are structured based on the interests of the specific user and ontologies. The advantages of using ontology in semantic search are many: Ontology-based optimization can locate the Web sites in the search engines, filtering of results using ontology depending on the search concepts extracted from social Web sites to enhance the search, and it improves the relevance of search results based on search-term disambiguation. Semantic richness in the Web mining process that is provided by the ontologies to enable the realization of the outcomes retrieved and the entire performance of the search process. While browsing contents through the Internet using the search engines, such as Bing and Google, the search outcomes are obtained irrespective of the polarities, or they provide the scores depending on the sentiment information generated through the search. Hence, the techniques based on sentiment analysis can be used into a general-purpose sentiment search engine. In general-purpose semantic search engines, the search results are structured by comparing the meaning and not on the popularity of the search terms. Offering functionalities other than sharing the information with the users and search based on the keywords is a complicated task for the search engines, if the semantic information is not specified.

Metasearch engine identifies some new Web pages which are not in the top ten results of any other search engines. These search results are obtained through the semantic matching of keywords obtained through WordNet. Also, the ranking of

the Web pages is changed based on the significance and semantic richness. In [22], more user-oriented search results are obtained using the ontological approach in the clustering process. Here, an initial search is done on several search engines and the search results are preprocessed, transforming them into word vectors. WordNet lexical database is utilized to map the word vectors into concept vectors. Recently, with the utilization of text-based search engines, the commercial search engines can make a significant advance with better results for image searching. Even though the research on semantic image retrieval is growing significantly, it is still not widely employed for use due to the rapid index querying, which in turn leads to computational overhead. For the search requests associated with the images, there exist two categories of search requests, such as unique and non-unique queries. The queries that are satisfied based on the retrieval of a unique person, event, or object are known as unique queries, whereas all the other cases come under the category of non-unique queries. Recent search engines like Yahoo and Google utilize text descriptions adopting semantics. Thematically homogeneous groups can be constructed with the help of clustering process from the initial list generated using standard search mechanisms.

4.4 Designing of Merging Strategy and Algorithm

The important challenge to be considered in the Metasearch engine is how to integrate all the results of the multiple search engines and the ranking of those results. This includes the process of selecting the search engine results of important Web pages, removing unnecessary Web pages retrieved from the search engine, the ranking of Web pages. These challenges pose confront of designing aggregate ranking algorithm for Metasearch engine. The merging of multiple criteria is required to generate a score value for every page after the computation of the scores of multiple search engines through different criteria [23]. Convolutional search engines usually performed merging by assigning a numerical matching score to every search result obtained so that the result-merging techniques designed could normalize the scores retrieved from the search engines into values. The ranking is carried out in the search engine to rank the results based on the normalized scores. As mentioned in the architecture of metasearch engine, the usefulness of the search engines selected is estimated to assign weight to the normalized score that helps in ranking the results obtained from useful search engines to be higher. Aggregation of search results can also be performed using voting-based techniques.

Another approach in result merging is downloading all the documents obtained from multiple search engines and then finds their matching scores using a similarity function adopted by the metasearch engine. Thus, the approach computes the ranking score evenly by the score-based ranking. Presently, the search engines visualize the returned result using the title and the snippet, which can determine whether the query is relevant or not [21]. Hence, most of the result-merging techniques developed recently are based on titles and snippets. In such case, it

requires a matching score computed relying on the proximity of the query terms in the title and snippet. The results are assumed to be relevant if the same result is obtained from multiple search engines. The ranking score measured helps in ranking the search results in descending order.

4.5 Visualization of Merged Results

The final step is to visualize the ranked Web pages to user-friendly interface to easily read and analyze the retrieved information. The SAVVY is a metasearch engine developed to calculate other search engines through autonomous collaborative decisions and provides the vibrant semantic link design and harmonized multiple view visualization. For illustration, it is considered that a user can have access to a number of resources. The user selects and taps to any one of the links, which are properly created as his operation. The chosen link takes the user to the required resources and benefits the user through providing a number of related links and the user can view all the related resources. Thus, the link provides the contents of the normal Web browsing process and it provides divergence that is harmonized equally. The user clicks the available static link once he adds a keyword into the search engine against the backdrop of modern browsing scenarios. Moreover, this stationary link may not benefit the user, though it eases the access to view the resources that are linked to the user welfare. This problem is solved by using the traits of the energetic semantic link design and harmonized multiple view visualization. The semantic Web is a Web-based technology that enlarges the XML by providing the way to display the ontologies by describing the objects and relations existing among them. The advantage of SAVVY is that there is no need for the user to click the links of the list and browse the pages frequently.

To check Search Engine Optimization, which is the process of increasing the views of a particular Web site such that the Web site appears top in the result list provided by a search engine, text network visualization of search results can be extremely useful. Text network visualization is useful in identifying the areas between the keyword clusters that attempt to co-occur in the snippets. These empty areas indicate the missing parts in the results so that one can include those parts in the text to appear at the top of the search results of the search engine. The search engines provide snippets of text in their results that are relevant to the search query. Hence, it will be helpful if the words contained in those snippets are known, so that the content that is relevant both for the search engine and the audience can be attained. Therefore, the fundamental step in developing a metasearch engine is to better understand the interests of the people identifying the terms they are actually searching for, i.e., the context. Thus, it is possible to know other keywords to be included in the text to appear in the results of the search engine.

5 Stability Validation Criteria of Metasearch Engines Based on Various Queries

This section analyzes the effects of different kinds of metasearch engines on various queries. Here, four kinds of queries are taken for analyzing the retrieval effectiveness of the metasearch engines, as described below.

5.1 Contextual Queries

The contextual retrieval of Web pages is important for the search engines because the Web pages should be different for the same query based on the location. Accordingly, Fathalla et al. [24] had developed the NEC Research Institute (NECI) metasearch engine by downloading and examining each document so that results showing query terms in the context can be displayed. Thus, the efficiency and the precision of Web can be improved. Besides, the users can recognize easily whether the relevant document is retrieved progressively without downloading every page. This technique is simple and more effective, especially when handling with large, poorly organized, and diverse database of the Web [25]. Generally, the metasearch engines can reduce the trouble of returning focused queries if the user interface automatically selects the search engines that are context relevant. When a user who works on an economics paper is determined by the interface, it can return a context description based on this information, and thereby, select a dedicated and context-relevant search engine like CNN financial. According to the discussion made on contextual queries, following three issues are addressed:

- (i) Extraction and representation of the context: The way of determining and describing the query context.
- (ii) Characterization and Selection of the source: Selecting useful sources such that they support decisions regarding source relevancy to provide an effective access. For the better search, the process of characterization of the sources should be simple, fast, and robust, without depending on the cooperation of the sources.
- (iii) Selectivity: Determining when the specialized sources accessible become insufficient.

Eight kinds of context that a search engine used for contextual search are given as follows,

- *Individual*: Considering a person's history and context for an effective search.
- *Demographic Profile*: Including age, gender and occupation, as they can predict the interests of the person.
- *Interest Profile*: Expresses the topics one who is interested in.

- *Location*: Showing accessible shops, hotels, and so on, in the proximity of the town and the country.
- *Device*: Type of device and interface used.
- *Date*: Noticing weekdays, which are in proximity to events like Christmas or New Year.
- *Time*: Time of the day.
- *Weather*: Focusing on current weather to offer a search for local places, such as tourist spots that are popular on summer.
- *Mood*: Search based on the moods like excitement, positivity, and negativity, as they could impact the content.

5.2 Personalized Queries

The analysis of the personalized queries is important for the users to understand the benefits and the features of various metasearch engines [16]. More accurate results can be returned for personalized query satisfying the user queries. For an effective personalization, the search engine has to concentrate on the nature of task executed by a user, in addition to the needs, preferences, and interests of the user. These rely on the observed patterns and the resulting probabilities. Few strategies of personalization-based metasearch engines are as follows:

- (i) Based on the results retrieved: The average of the results obtained from the database of the search engines is computed. Then, select the useful search engine according to the averages of search results returned.
- (ii) Based on the experience of the user: The user selects the search engine based on his/her personal experience. Personalization depending on group-based behavior can personalize the results by analyzing the actions of the users who made similar queries before. The search engine customizes the results based on online behavior, search activity, and other details recorded in an anonymous cookie in the browser of the user. Utilizing these cookies, the search engines collect and store the search history in the databases. Metasearch engines personalize the search query based on certain features, some of which are given below,
 - *Device (Operating System and Browser)*: The search results may be customized based on the device used, as the content people search varies depending on the type of device they use. A mobile user may search for more actionable, driving directions, and on-the-go information, while a desktop user may browse completely different information for the same query.
 - *Location*: The locations of the searchers have an impact on the search results as people search for the same query at different locations.
 - *Previous Searches and Frequent Site Visits*: Metasearch engines like Bing and Google may customize the search results based on previous search behavior.

The techniques used, allow the metasearch engines to consider previous search queries, and the contents taken from previous searches.

- *Bookmarks*: A user may bookmark a site or visit often for specific kinds of searches, such as refilling prescriptions, booking tickets, and so on, as the user has to visit the page for those searches in the future. The search engines utilize all the personal information submitted to show the site when needed.
- *Logged In and Out*: When a user login to an e-mail account or other similar profiles with the search engines, the results are likely to bias heavily to the signals like location and operating system.

5.3 *Semantic Queries*

The query word that has different meanings is a common issue, which is to be considered in the semantic searches. Hence, the metasearch engines designed have to utilize the semantic information so that the results are retrieved focussing on the relevant pages of the user. Following are the features considered by a metasearch engine for semantic queries,

- *Language*: Understanding the synonymous languages to enhance the search process.
- *Linked data*: Determine the actual relationships between information to find similar answers.
 - *Context*: The search engine tool tends to understand the context and meaning of the words in the query to provide accurate search results.
 - *Previous Searches*: These are analyzed to find semantic connections. The search engines must understand the related topics, to determine the relationship between the items.

5.4 *Numerical Queries*

The effect of numerical queries and its relevancy in retrieving the Web pages by the different metasearch engine is discussed in this part. Even though numbers play a significant role in modern life, most of the search engines assume numbers as strings, neglecting their numeric values. This is because of an unreasonable expectation that the user offers exact numeric values during the searches. In reality, they search for values whose specifications approximately match the values given in the query. As a result, techniques are developed to obtain the specification documents for the extraction of the pairs of attributes and values in a document and are stored in a database. Hence, the queries can be processed with the utilization of nearest neighbor approaches [26]. Numerical queries are carried out based on,

- Economic value of the data;
- Chronological reviews;
- Probability of the value.

6 Current Problems and Issues to Be Addressed

This chapter discusses the problems and issues with respect to various kinds of metasearch engines explained. Presently, certain research groups provide search results from the semantics-based search engines but most of which are in their developing stages. One of the large global databases is current Web, which lacks the subsistence of a semantic structure. Another major issue deals with the ranking of the results. Based on the predetermined condition, search engines assign ranks to the generated documents in the descending order of relevance in accordance with the user's preferences. A long list of titles of the documents is the result of the ranking process obtained by a search engine. The major limitation of such method is that the user has to browse through this long list to find the results that the user is actually looking for. Inefficiency in making distinctions between the various concepts available in the query generated in the resultant list of the documents by the search engine is also a drawback, since the list must be ranked sequentially [1].

A common difficulty in metasearch engine is that it relies on the underlying search engines to return a significant set of results. Metasearch engines can only offer a limited number of results to be retrieved to the user. To improve the accuracy of the search results, dealing with the limitations regarding the results due to the issues caused by search engines, metasearch engines modify the query by providing a search range that uses certain options, like sorting by date and time period, specifying the number of items to be retrieved, language constraints, number of items to be displayed. Another issue is the problem regarding the selection of relevant search engines based on the user query [11]. This is a challenging task, which can be solved to an extent by selecting a large number of search engines for the query [25]. Adding more search terms for better results is another problem, as it becomes a burden to the user. Moreover, it is difficult even for the experts to choose the appropriate query terms for the information to be returned by the search engine. Therefore, it is difficult for a machine to recognize the information provided by the user. When the information was circulated in the Web, there exist two types of research problems in search engine; i.e.,

- How a search engine can map a query to the documents having information but are not retrieved in intelligent and useful information?
- The query results provided by the search engines are distributed across various documents that are connected with a hyperlink. How a search engine can identify such a distributed search results efficiently?

7 Conclusion and Future Research Paths

In this chapter, a detailed description of various metasearch engines developed for information extraction, solving the common issues of search engines is presented. Different categories of search engines and the architecture of a typical metasearch engine explaining the process of information retrieval are explained in detail. Further, the ranking mechanisms and the important conditions to be satisfied during the development of the metasearch engines are discussed. The chapter has also illustrated several criteria used for the validation of the effectiveness of the metasearch engines, addressing their common issues.

The metasearch engine can support several interesting and specialized applications. The application of metasearch engines to education and e-commerce problems has not been highly studied. In this context, we suggest two main future research directions:

- In a large organization with a number of branches, such as a university having many campuses, a metasearch engine that connects the search engines of all branches turn out to be an organization-wide search engine.
- When a metasearch engine is constructed over different e-commerce search engines that sell the same type of product, it is reasonable to form a comparison-shopping system. Certainly, it requires a different result-merging technique for comparison-shopping applications.

References

1. Hamdi, M. S. (2011). SOMSE: A semantic map based meta-search engine for the purpose of web information customization. *Applied Soft Computing*, 11(1), 1310–1321.
2. Lawrence, S., & Giles, C. L. (1998). Context and page analysis for improved web search. *IEEE Internet Computing*, 38–45.
3. Terziyan, V., Shevchenko, O., & Golovianko, M. (2014). An introduction to knowledge computing. *Eastern-European Journal of Enterprise Technologies*, 1(2), 27–40.
4. Agrawal, R., & Ramakrishnan, S. (2003). Searching with numbers. *IEEE Transactions on Knowledge and Data Engineering*, 15(4), 855–870.
5. Antoniou, D., Plegas, Y., Tsakalidis, A., Tzimas, G., & Viennas, E. (2012). Dynamic refinement of search engines results utilizing the user intervention. *Journal of Systems and Software*, 85(7), 1577–1587.
6. Kumar, D., & Kumar, A. (2013). Design issues for search engines and web crawlers: a review. *IOSR Journal of Computer Engineering (IOSR-JCE)*, 15(6), 34–37.
7. Sharma, S. (2008). Information retrieval in domain specific search engine with machine learning approaches. *International Journal of Social, Behavioral, Educational, Economic, Business and Industrial Engineering*, 2(6).
8. Meng, W.: Metasearch engines. In *Encyclopedia of database systems* (pp. 1730–1734).
9. Nwosu, O., & Anyira Echezonam, I. (2011). Inquiry into the use of Google and Yahoo search engines in retrieving web resources by internet users in Nigeria. *Indian Journal of Information Sources & Services (IJISS)*, 1(2), 61–67.

10. Vijaya, P., Raju, G., & Kumar Ray, S. (2014). S-MSE: Asemantic meta search engine using semantic similarity and reputation measure. *Journal of Theoretical & Applied Information Technology*, 60(2).
11. Ray, S. K., Vijaya, P., & Raju, G. (2013). An ontology based meta-search engine for effective web page retrieval. *International Review of Computers and Software (IRECOS)*, 8(2), 533–541.
12. Poulter, A. (1997). The design of world wide web search engines: A critical review. *Program*, 31(2).
13. Weideman, M., & Kritzinger, W. (2003). Search engine information retrieval: Empirical research on the usage of meta-tags to enhance website visibility and ranking of e-commerce websites. In *Proceedings of the 7th World Conference on Systemics, Cybernetics and Informatics* (Vol. 6, pp. 231–236).
14. Vijaya, P., Raju, G., & Ray, S. K. (2016). Artificial neural network-based merging score for Meta search engine. *Journal of Central South University*, 23(10), 2604–2615.
15. Naval, P., & Priyanka, S. (2012). A survey on personalized meta search engine. *International Journal*, 2(3).
16. Jadidoleslamy, H. (2012). Search result merging and ranking strategies in meta-search engines: A survey. *IJCSI International Journal of Computer Science Issues*, 9(3), 239–251.
17. Maloth, B. (2010). *Evaluation of integration algorithms for meta-search engine*. Technical report. [http://www.bvicam.ac.in/news/INDIACom%202010%20Proceedings/papers/Group3/INDIACom10_261_Paper%20\(3\).pdf](http://www.bvicam.ac.in/news/INDIACom%202010%20Proceedings/papers/Group3/INDIACom10_261_Paper%20(3).pdf).
18. Jadidoleslamy, H. (2011). Introduction to metasearch engines and result merging strategies: A survey. *International Journal of Advances in Engineering & Technology*, 1(5), 30–40.
19. Ngu, A. H. H. (2005). Business & economics, In *6th International Conference on Web Information Systems Engineering, WISE 2005*, New York, NY, USA.
20. Ngu, A. H. H., Kitsuregawa, M., Neuhold, E., Chung, J.-Y., & Sheng, Q. Z. (2005). Computers. In *6th International Conference on Web Information Systems Engineering, WISE 2005*, New York, NY, USA.
21. Li, Z., Wang, Y., & Oria, V. (2001). A new architecture for web meta-search engines. In *AMCIS 2001 Proceedings*, 31(84), 415–422.
22. Das, S., & Raghuvanshi, K. S. (2014). Search engine selection approach in meta search using past queries. *Oriental Journal of Computer Science & Technology*, 7(1), 177–183.
23. Rohini, M. (2015). Reformation of query based approach in search engine. *International Research Journal of Engineering and Technology (IRJET)*, 2(5).
24. Fathalla, S. M., Hassan, Y. F., & El-Sayed, M. (2012). A hybrid method for user query reformation and classification. In *Proceedings of Computer Theory and Applications (ICCTA), 22nd International Conference on IEEE2012* (pp. 132–138).
25. Pablos, O. (2012). Patricia: Advancing information management through semantic web concepts and ontologies. *IGI Global*.
26. Leake, D. B., & Scherle, R. (2001). Towards context-based search engine selection. In *Proceedings of the 6th International Conference on Intelligent User Interfaces*, ACM (pp. 109–112).