Refinement in Meta Search Engine for Effective Results Based On Relevance Feedback

[1]Kayalvizhi.C., [2]Swaraj Paul.C.,
[1]Student, [2]Assistant Professor,
Vels University,Chennai

Abstract - In recent days there is continuous expansion of data in World Wide Web which makes a single search engine to index the entire web for resources impractical. Metasearch engine is a key to solve this problem. Metasearch means placing the search query to multiple search engines and providing a merged result as output to the user. So thereby it complements user's experience in accessing the required information. This paper reveals the crucial aspect of duplicate items and missing documents in the result of merging activities of metasearch engine in particular and proposes new solutions. It also aims at collecting feedback to improve the ranking and to make changes in position vectors accordingly so that the results are displayed in a better way to the users.

Keywords: Missing documents, broken links, Duplicate documents, Data fusion, Rank aggregation, OWA operator, Searching, Information searches, Information retrieval.

I. INTRODUCTION

Information on the World Wide Web is searched and captured using a system software called search engines. The search results are generally presented in a line of results often referred to as[1] search engine results pages (SERPs). The information may be a mix of images, videos, web pages and any other types of files. First a search engine must find the file or document which the user searches. A search engine operates [1] in the following order:

1. Web crawling
2. Indexing
3. Searching

A special software robot, called **spiders**[2] is employed by search engines to find the required information from the existing millions of web pages. Spiders are used to build lists of the words found on Web sites. This above process is called Web Crawling.

When the Google spider looks at an HTML [2]page, it take note of two things:

- The words within the page
- Where the words were found

Relative position of words i.e whether the words are present at titles, subtitles or meta tags[2] is taken for consideration. The Google spider was designed in a way that it builds index with most significant words on a page. Each spider has their own approaches to build index. This difference makes [2]each spider to work faster than other and provide a better search results. For example, some spiders will keep track of the words in the title, sub-headings and links, along with the 100 most frequently [2]used words on the page and each word in the first 20 lines of text. Meta tags allow the owner of a page to specify key words and concepts under which the page will be indexed.

Next step of indexing is [2]to store all the information in such a manner that they are easily accessible. User writes a query submit it to the search engine, at the backend searching is done over the index.

The main aim of a metasearch engine is to submit a particular query to distinct search engines and to fuse the individual result lists into an [3] overall ranked list of documents that is presented to the user. By combining multiple results from different search engines, Metasearch engine is able to enhance[3] the user's experience for retrieving information, as less effort is required in order to access more materials. A simple architecture of metasearch engine is depicted in figure 1.1

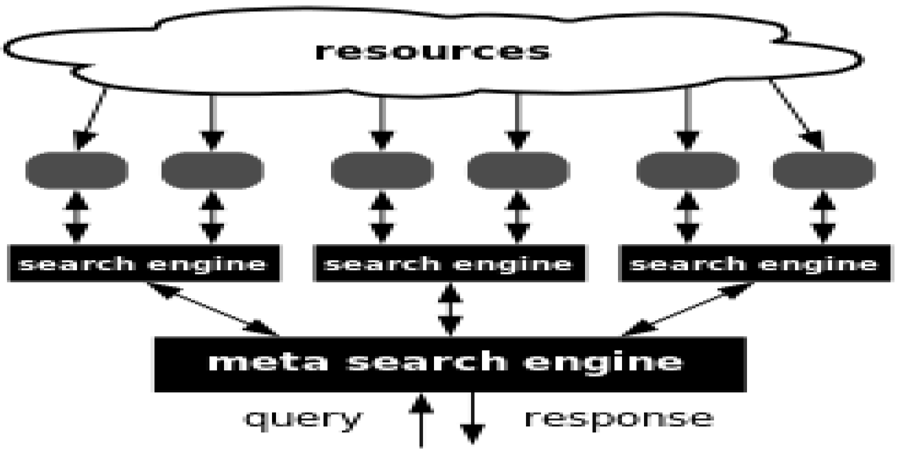

Fig 1.1 Architecture of metasearch engine[3]

II. CHALLENGES OF METASEARCH ENGINE AND SOLUTIONS

The role of missing documents in result aggregation

When you use a search engine you are not actually searching the live web. Instead you are working with the search engine’s [4] database of webpage information. The ongoing process of how search engine works is described in detail in first part of this paper and this field is in continuously developing. A missing document is a relevant document that has been retrieved by some search engines, but not by all [5] Hence they appear in some ranked lists.

Consider to a metasearch engine a user submits a query. The metasearch engine passes the query to its m ($m \geq 2$) underlying search engines called SE_1, SE_2, . . . , SE_k, . . . , SE_m and extracts [4]the top n results of each one. The positional value (PV) of a document d_i in the result list l_k returned by a search engine SE_k is defined by Diaz et al. [5]as:

$$PV_{ik} = n - r_{ik} + 1 \quad (1)$$

Where r_{ik} is the rank of d_i in search engine SE_k . Hence the higher the rank of a document in a result list, the larger the positional value of the document in that list.

A document might be missing from a ranked list if the search engine:

- If it is not crawled
- If it is not indexed
- If it is not retrieved

Calculation for missing [4] documents based on these cases is different. Now consider case i and ii, where there is missing of documents as search engine does not crawl or it does not index the document. So let m be the total number of search engines. Let q ($1 \leq q \leq m – 1$) indicate the number of search engines in which d_i has been crawled or indexed. Let p ($1 \leq p \leq q$) denote a search engine among q search engines. [4] Let PV_{ip} be the positional value of document d_i in the p_{th} search engine. In this case the PV of document d_i in the k_{th} search engine is computed as:

$$Pv_{ik} = \frac{\sum_{p=1}^{q} PV_{ip}}{q} \qquad (2)$$

Now consider case iii where the document is not retrieved though it is crawled and indexed , in this situation positional value [4]is assigned to be zero. Therefore in this case the PV of document[4] d_i in the k_{th} search engine is zero.

III. THE CHALLENGE OF DUPLICATE DOCUMENTS

A metasearch engine submits user query to multiple search engines, an now each search engine returns its response in the[4] form of search results. Though each search engines try not to present duplicate documents as their response to query, in merging process there are chances for metasearch engine to generate duplicate documents. Those metasearch engines which detect these duplicates are highly efficient.

Documents that have identical content are[6][7] known us “exact duplicates” and those which have small portion identical are called partial “duplicates”. In this paper we propose downloading-based technique to solve duplicate document problems. When a metasearch engine receives a query it passes to search engines behind it, and extract all relevant documents. Instead of fetching the entire content of each document, the metasearch[4] engine considers downloading the first part of each one. Next the documents are compared to one another using the downloaded portions and identical ones are categorized in the same groups as duplicates. To ensure that duplicates have been truly detected, [4]for each group, another small portion of each document is downloaded and compared to others. This approach, in addition[4]to detecting duplicate documents, has another advantage. When trying to download portions of documents, dead URLs can be identified and consequently removed from the result list.

Collecting Feedback:

Feedback of all the pages which the user visit from the search result is collected at the end when user closes the page. This feedback is used along with positional values and ranking is recomputed. Feedback is collected both explicitly by certain measures which are predefined , that make use of Rocchio algorithm, or implicitly by users behavior such as the documents which they select for viewing, time spent on each document, or scrolling action etc.

$$FS_{pg}=\frac{\sum_{i=1}^{n} w_i}{n} \qquad (3)$$

Where FS is feedback score of a page, which is given as average of individual feedback weight by each user.

The algorithm described above can be easily implemented by a programming language that is beyond the scope of this paper.

IV. CONCLUSION

In the present paper we have suggested approaches to handle the problem of result merging in a metasearch engine environment. We have discussed the concept of missing documents and their three different cases in detail .Duplicate document handling is also explained in brief manner. Feedback score is also used efficiently.

V. REFERENCES

[1] “Web search engine - Wikipedia, the free encyclopedia.” .

[2] C. Franklin, “How Internet Search Engines Work.” .

[3] “Metasearch engine - Wikipedia, the free encyclopedia.” .

[4] H. Sadeghi, “Empirical challenges and solutions in constructing a high-performance metasearch engine,” Online Inf. Rev., vol. 36, pp. 713–723, 2012.

[5] E. D. Diaz, A. De, and V. V Raghavan, “On Selective Result Merging in a Metasearch Environment,” pp. 52–59, 2004.

[6] G. S. Manku, A. Jain, and A. Das Sarma, “Detecting Near-duplicates for Web Crawling,” Proc. 16th Int. Conf. World Wide Web, pp. 141–150, 2007.

[7] B. S. Alsulami, M. F. Abulkhair, and F. E. Eassa, “Near Duplicate Document Detection Survey,” vol. 2, no. 2, pp. 147–151.