I.J. Information Technology and Computer Science, **2012, 3, 8-15**
Published Online April 2012 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijitcs.2012.03.02

Implementing SEReleC with EGG

Vishwas J Raval
M Tech CSE Scholar, Electronics & Computer Engineering Department, Indian Institute of Technology Roorkee, Uttarakhand, India
Email: vishwas.raval@gmail.com

Dr. Padam Kumar
Professor & Head, Electronics & Computer Engineering Department, Indian Institute of Technology Roorkee, Uttarakhand, India
Email: padamfec@iitr.ac.in

Abstract — The World Wide Web has immense resources for all kind of people for their specific needs. Searching on the Web using search engines such as Google, Bing, Ask have become an extremely common way of locating information. Searches are factorized by using either term or keyword sequentially or through short sentences. The challenge for the user is to come up with a set of search terms/keywords/sentence which is neither too large (making the search too specific and resulting in many false negatives) nor too small (making the search too general and resulting in many false positives) to get the desired result. No matter, how the user specifies the search query, the results retrieved, organized and presented by the search engines are in terms of millions of linked pages of which many of them might not be useful to the user fully. In fact, the end user never knows that which pages are exactly matching the query and which are not, till one check the pages individually. This task is quite tedious and a kind of drudgery. This is because of lack of refinement and any meaningful classification of search result. Providing the accurate and precise result to the end users has become Holy Grail for the search engines like Google, Bing, Ask etc. There are number of implementations arrived on web in order to provide better result to the users in the form of DuckDuckGo, Yippy, Dogpile etc. This research proposes development of a Meta search engine, called SEReleC that will provide an interface for refining and classifying the search engines' results so as to narrow down the search results in a sequentially linked manner resulting in drastic reduction of number of pages.

Index Terms— *web crawlers, Search Engine, HyperFilter, HyperUnique, HyperClass*

1. Introduction

Web search engines are the tools to search the contents stored across World Wide Web. The results generated may be pages, images, ppts or any other types of files. The results of search engines are displayed in the form of a list in which the numbers of pages might be in thousands or millions. The usual working of a search engines consists of following:

1) They search the Internet or select pieces of the Internet based on important words - crawling
2) They keep an index of the words they find, and where they find them - indexing
3) They allow users to look for words or combinations of words found in that index - searching

Early search engines used to hold an index of a few hundred thousand pages and documents, and received maybe one or two thousand inquiries each day. Today, a top search engine indexes hundreds of millions of pages, and respond to tens of millions of queries per day. This is done using proprietary algorithms, which work based on the assumption that if a page is useful, other pages covering the similar topic are likely to provide a link to it [1]. The famous search engines are Google, Yahoo, Bing, Ask.

Before moving ahead, we take a dip into the generations of search engines first. Around 1995-97, AltaVista, Excite, WebCrawler, etc. which are first generation used mostly on-page data (text and formatting) and was very close to classic Information Retrieval. They support mostly informational queries. In the beginning, search results were very basic and largely depended on what was on the Web page. Important factors included keyword density, title, and where in the document keywords appeared. First generation added relevancy for META tags, keywords in the domain name, and a few bonus points for having keywords in the URL. <Meta> tags allow the owner of a page to specify key words and concepts under which the page will be indexed. This can be helpful, especially in cases in which the words on the page might have double or triple meanings – the <meta> tags can guide the search engine in choosing which of the several possible meanings for these words is correct. There is, however, a danger in over-reliance on <meta> tags, because a careless or unscrupulous page owner might add <meta> tags that fit very popular topics but have nothing to do with the actual contents of the page. To protect against this, spiders will correlate <meta> tags with page content,

rejecting the meta tags that don't match the words on the page. Due to these limitations of <meta> tag, search engines started using web crawlers or known as spiders too. Following figure 1 and text illustrates the basic architecture of a web crawler.

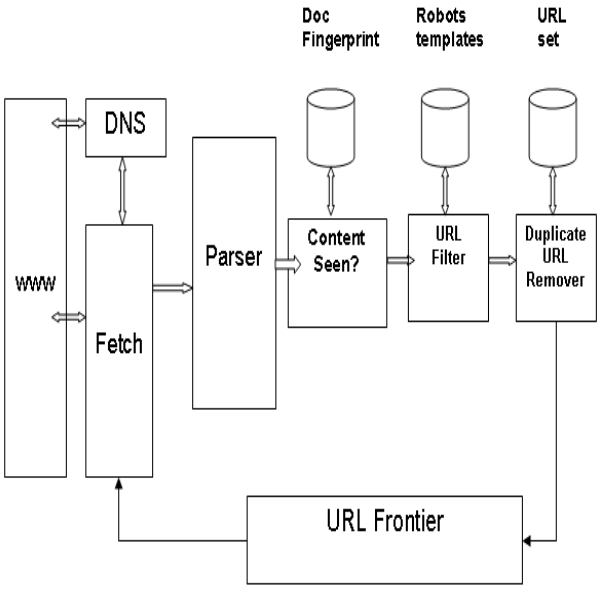

Figure 1. Architecture of Web Crawler

1) URL Frontier: It contains URLs yet to be fetched in the current crawl. At first, a seed set is stored in URL Frontier, and a crawler begins by taking a URL from the seed set.
2) DNS: Domain name service resolution which looks up IP address for domain names.
3) Fetch: It generally use the http protocol to fetch the URL.
4) Parse: The page is parsed. Texts (images, videos, and etc.) and Links are extracted.
5) Content Seen? This checks if a web page with the same content has already been seen at another URL. Need to develop a way to measure the fingerprint of a web page.
6) URL Filter:
    a. Whether the extracted URL should be excluded from the frontier.
    b. URL should be normalized (relative encoding).
        i. en.wikipedia.org/wiki/Main_Page
        ii. Disclaimers
7) Duplicate URL Remover: The URL is checked for duplicate elimination so that spider does not fall into a recursive loop.

The Second generation search engines use off-page, web-specific data such as link analysis, anchor-text, and Click-through data. This generation supports both informational and navigational queries and started in 1998-1999. Google was the first engine to use link analysis as a primary ranking factor and DirectHit concentrated on click-through data. By now, all major engines use all these types of data. Link analysis and anchor text seems crucial for navigational queries.

The Third generation which is now emerging is attempting to blend data from multiple sources in order to try to answer “the need behind the query”. For instance, when user searches for New York, the engine might present direct links to a hotel reservation page for New York, a map server, a weather server, etc. Thus third generation engines go beyond the limitation of a fixed corpus, via semantic analysis, context determination, natural language processing techniques, etc. The aim is to support informational, navigational, and transactional queries. This is a rapidly changing landscape. In spite of having entered into the third generation of search engines, no search engine has been able to provide the result which is most accurate and precise with reference to the search query though they have their own strong and efficient page ranking, indexing and search algorithms.

Meta search engines base their services on several individual search engines. They borrow services provided by their member search engines and return the integrated results. They neither own an index database or a classification directory, which is the biggest difference with individual search engines [15]. There are many meta-search engines like Dogpile, Yippy, DuckDuckGo etc. running on www which takes user input, pass it to other search engines, process the result and return it to the user in better way. A typical architecture of a meta-search engine is given in figure 2 [13].

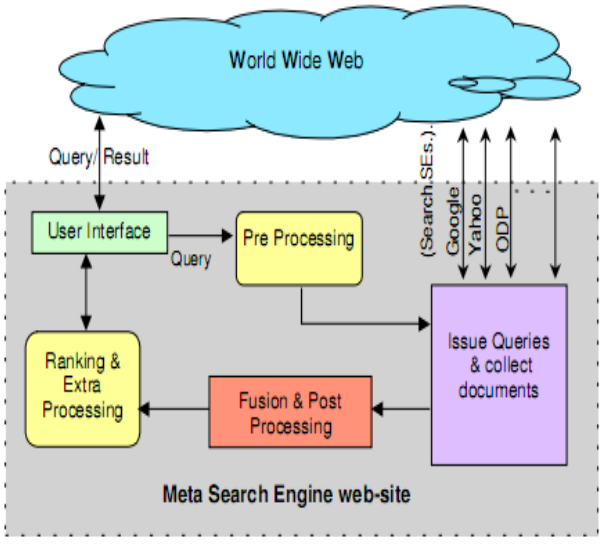

Figure 2. Architecture of a typical Meta-search engine

Though the search and meta-search engines provide users an ocean of pages as a response to their query, there are few very primitive limitations and problems [2] that have remained unaddressed till date, in spite of many attempts, due to which common users have to dig

through the returned result set to land onto their desired pages by refining it manually. Following section discusses those problems.

The paper contains mainly 7 sections. In Section 2 we show the results of experiments we carried out on several search and meta-search engines to highlight the limitations of them. In Section 3 we discuss the related work done in this area. Section 4 contains detailed description about the concept we proposed with architecture and interface. Section 5 shows the results derived from SEReleC with reference to section 2. Section 6 shows possible merits and demerits of the work and its implementation. In section 7, we conclude our work with remark on the current status of the work.

2. Primitive Problems of Search engines

In order to discuss the problems, we carried out several experiments using three basic and most famous search engines Google, Yahoo, Bing and three famous and widely used meta-search engines Yippy, DuckDuckGo and Dogpile. Though most of these search engines have advanced options for search, by an in depth study, we identified that all of these search engines have either or all of the below mentioned three basic problems in the results returned. We have experimented mainly for a normal search and an exact search. Normal search is the usual search which is performed all the users most of the time. Exact search is the search which is performed by putting keywords in double-quotation mark which find exact sequence of the words in the search string. Legends for search keywords given as input for a normal and exact search in Google, Bing, Yahoo, Dogpile, DuckDuckGo and Yippy are as under:

Following text and figures discuss the problems and results of one of our carried out experiments.

2.1 A large number of unnecessary and irrelevant links:

We define usefulness of links if the links are references to the pages that matched the search keywords exactly. Search Engines, mainly Google, returns millions of links in response to the query of which only few links are useful to the user which user is interested in. In our case, when the search keywords were Vishwas Raval, search Engines returned links of the pages which contained even Vishwasan or Ravalia, Ravalgaon etc. words. This is since these results are based on Page Relevance. However, when search query was “Vishwas Raval”, search engines returned those links that matched exact word in the same sequence. Noticeable thing is that in case of exact match only we get the 100% accurate results (Refer figure 3 and figure 4). Here, the results are not zero but with reference to Normal Search, the results returned by Bing, Google and Yahoo are 31, 230 and 31 respectively.

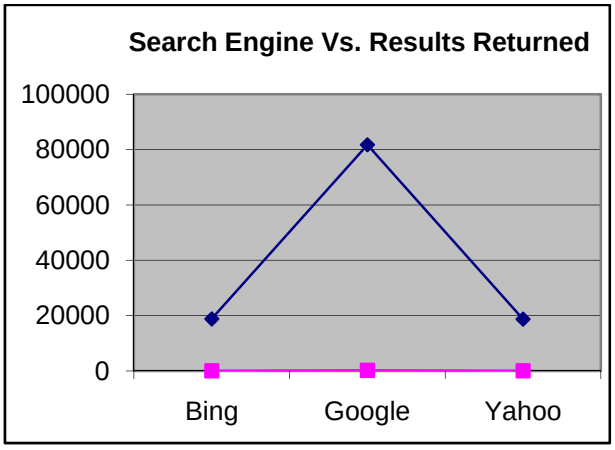

Figure 3. Number of Results returned by search engines

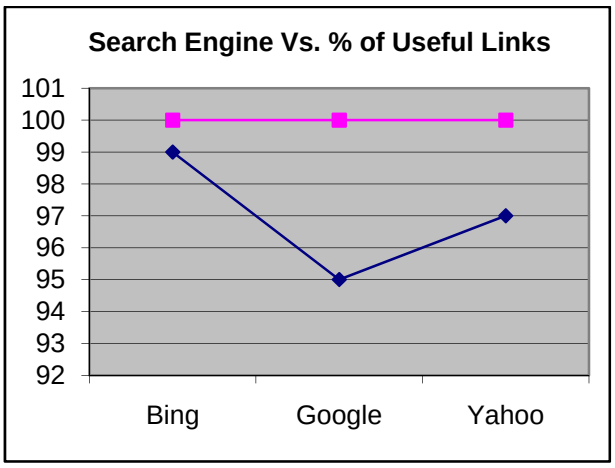

Figure 4. Useful links in first 100 links returned by search engines

Another important thing is that all the search engines including Google don’t perform combinatorial search. Combination of keywords plays a big role in accuracy. A query, for instance, “Vishwas Raval” should also return the results containing “Vishwas”, “Raval”, “Vishwas Raval” and “Raval Vishwas” too for a search since essentially these all could be reference to the same person. Omitting combinatorial search could miss some important relevant links which user might be interested in. A naïve internet user usually does not know the combination of keywords to give to search engines and hence many a times misses the important result which is not returned by search engine due to Page Relevance. So we require a method that provides combinatorial search with 100% accuracy which we developed as EGG [19].

In case of Meta search engines, when Vishwas Raval was the search string, the percentages of useful links were not up to the mark for Dogpile with reference to the results returned by it. See figure 5 and figure 6.

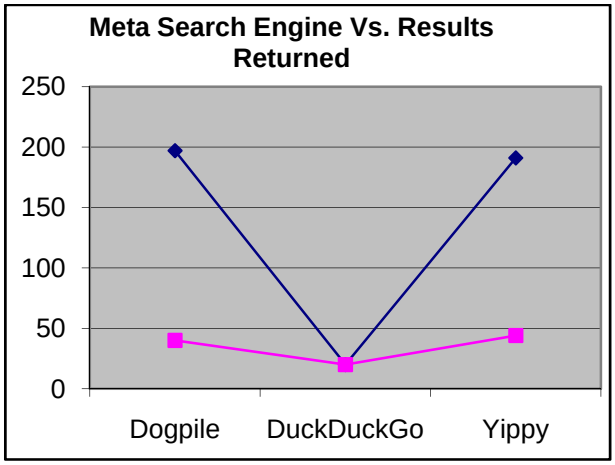

Figure 5. Number of Results returned by Meta search engines

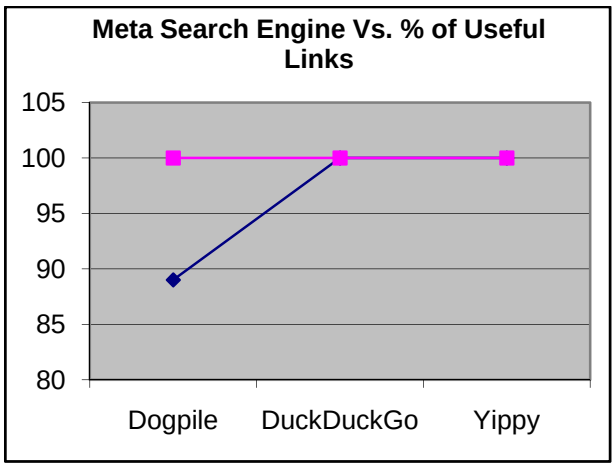

Figure 6. Useful links in first 100 links returned by Meta search engines

2.2 Redundant links:

Figure 7 and 8 shows the percentage of redundant links found with reference to the useful links found in figure 4 and figure 6.

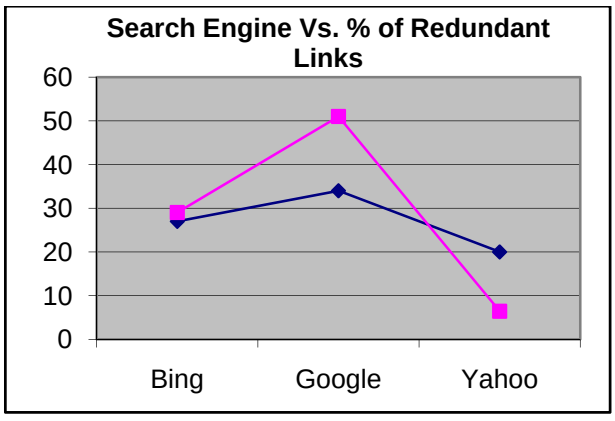

Figure 7. Redundant links in first 100 links returned by search engines

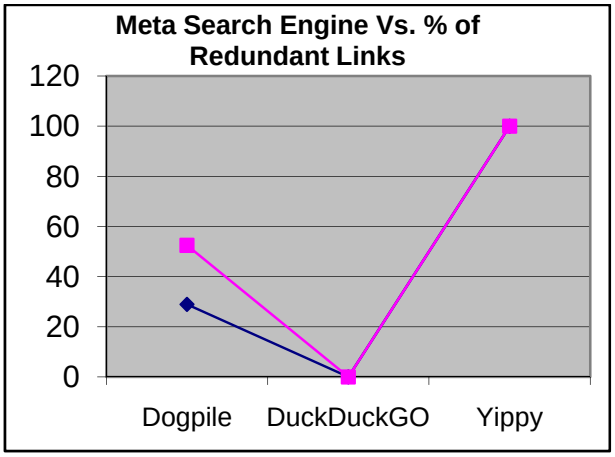

Figure 8. Redundant links in first 100 links returned by Meta search engines

Among the useful links from the result returned, many of the links were found redundant. For e.g. if Vishwas Raval is to be searched and it is found on various pages of some link www.abc.com then, just one link, stating existence of word Vishwas Raval on www.abc.com, is enough. Rests of the same links are not required as user would never go to same link again. This is how we define redundant links.

2.3 Unclassified results:

Though many of the above mentioned search engines classify the results but they are not based on search query. Google classifies based on chronology, images, shopping, blogs etc., whereas Yahoo and Yippy classify based on several famous words related to the query. Not any of the search engines classify the results based on the keywords in the search query hence we proposed a links classification algorithm based on combinatorial keyword search.

3. Related Work

The concept which we propose is not a new one. Many attempts have been made to resolve the issues but the works that have been carried out so far is lacking solution to one or other problem discussed above. One of the best examples of such a work is GuidedGoogle [5] which is implemented using Google Search API to guide google search engine for accurate search. Another example that is more closely related to Google would be the Google API Search Tool by Softnik Technologies [3]. It is a simple but powerful Windows software tool for searching Google. It is completely free and is not meant to be a commercial product. All that the users need to do is register with Google for a license key and they will be entitled to pose 1000 queries a day. It is also an efficient research tool because it allows the users to record the search results and create reports of their research easily and automatically. Similar work is implemented in CatS. CatS operates by forwarding the user query to a major Web search engine, and displaying the returned results together with a tree of topics which can be browsed to sort and refine the results [9]. Other related works [6] [7]

[8] [10] [11] [12] [14] [17] [18] have also been carried out but all of them were not addressing all the problems discussed in above section.

4. The SEReleC

Looking towards these unaddressed problems of search engines and Meta search engines, we propose an experimental development of an interface SEReleC[20] (**S**earch **E**ngine **Re**su**l**t **Re**finement and **C**lassification) which is a group of post-retrieval processes that works as a front-end and the search engine's Web Service interface (SEWS) [4] works in back end. We call it a meta-search engine as it works one layer above on existing search engines and dig through the search engines' results. Following are the details of how does SEReleC work:

4.1 HyperFilter:

When a user provides a search query to The SEReleC interface, it generates all possible combinations of search keywords and passes the query to search engine. HyperFilter makes sure that the results which it receives must exactly match with the search keywords and their possible combinations [5]. This is how it filters out the irrelevant and unnecessary links. It then passes the filtered results to HyperUnique.

4.2 HyperUnique:

Upon getting the filtered result from HyperFilter process, this process removes redundant links, if any and passes the result to the HyperClass process [16].

4.3 HyperClass:

This classifies the links of result set into classes created based on the all possible combinations of words of the query string. This process returns the classified result to the end user [5].

The three above mentioned modules are conceptual similar to OSI Layers. There is no explicit division of the modules. Figure 9 and figure 10 shows the proposed architecture and interface of The SEReleC respectively.

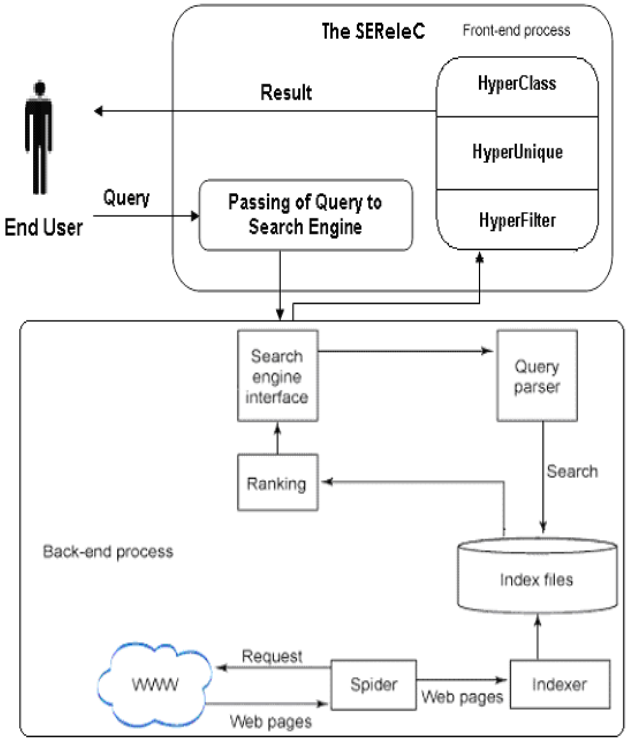

Figure 9. Proposed SEReleC Architecture

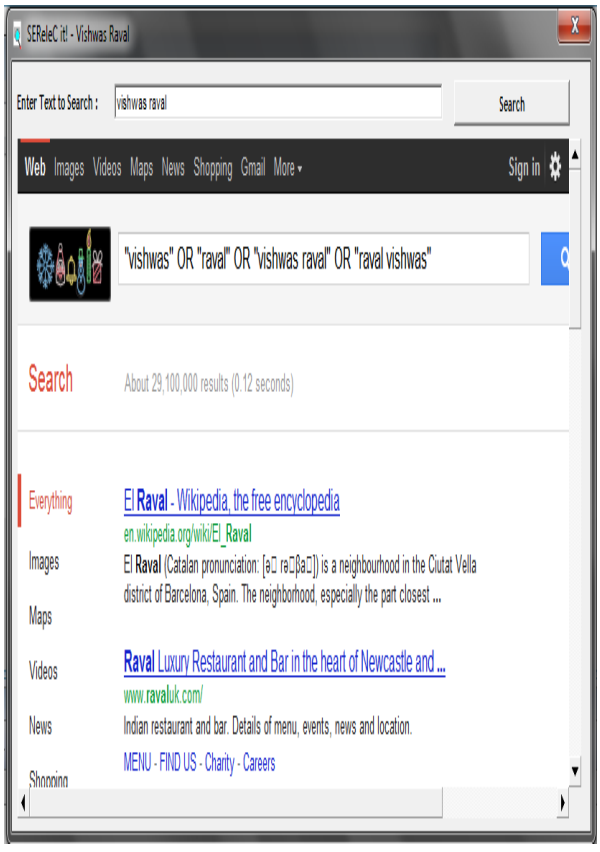

Figure 10. The SEReleC Interface

5. Results

With reference to the legends and results discussed in section-II, we present the results of SEReleC in comparison to the mentioned search and meta-search engines in figures 11 to 14.

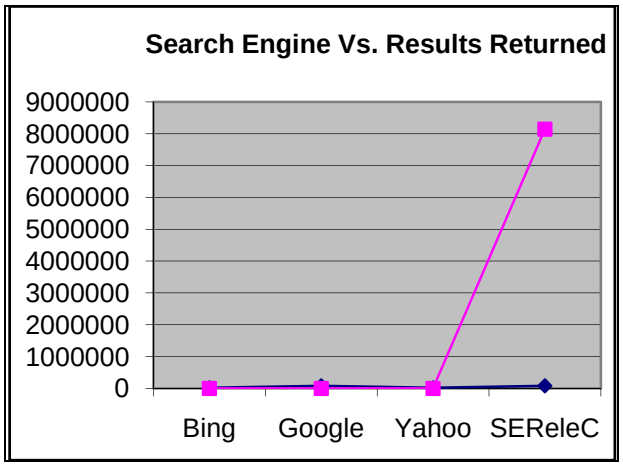

Figure 11. Number of Results returned by search engines

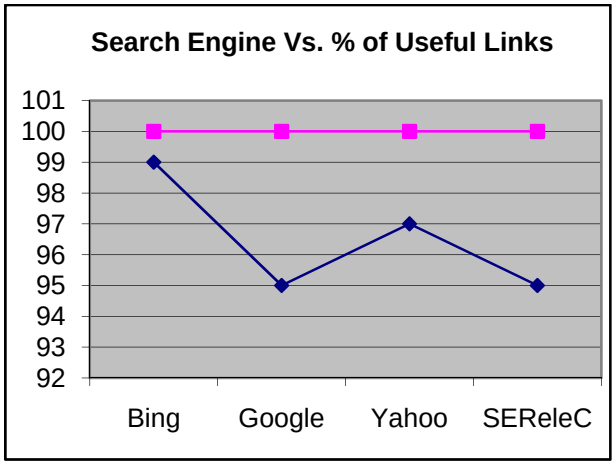

Figure 12. Useful links in first 100 links returned by search engines

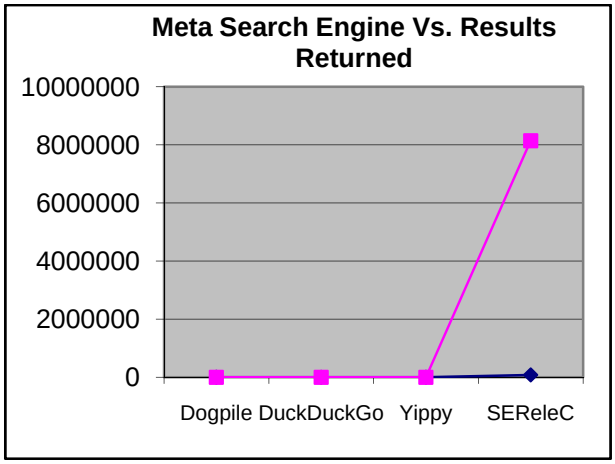

Figure 13. Number of Results returned by Meta search engines

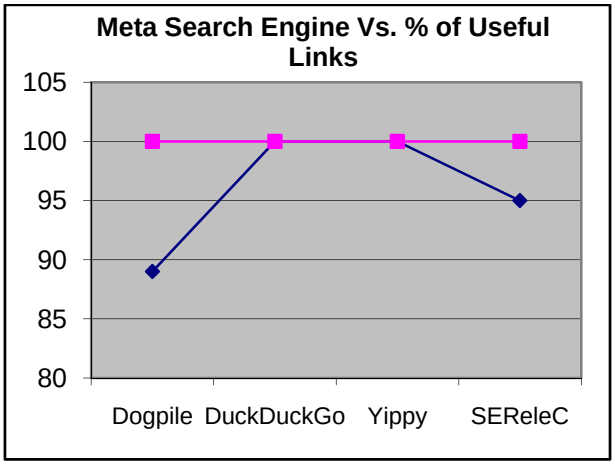

Figure 14. Useful links in first 100 links returned by Meta search engines

6. Merits And DeMerits

6.1 Merits

a. For each page user visits, the SEReleC will find exact match which would eliminate the irrelevant and unnecessary pages.

b. Redundant links will be eliminated that would in turn reduce the number of pages to visit.

c. Due to the classification, pages will be divided into categories as per the phrases and content matched, so user can directly go to the pages of interest directly which would save a lot of user efforts and time as well.

d. End users would always get what they want to have with lesser efforts.

6.2 Demerits

a. As it would work one layer above the search engines, so comparatively it will be a bit slower than search engines but that difference will be in seconds as compared to user efforts in minutes and hours.

b. Implementation point of view more efficient method of computing would be required.

7. Conclusion

SEReleC provides meta-search capability developed using power of Google based on concept of Guided Google and enhanced as EGG. It guides and allows the user to view the search results with different perspectives. This is achieved through simple manipulation and automation of the existing Google functions. This meta-search engine supports search based on “Combinatorial Keywords” and “Normal Search”. This work is part of my dissertation work under the guidance of Dr. Padam Kumar and the result given by the Google using EGG will be used for other modules of my dissertation work.

So far, HyperFilter and HyperUnique Modules have successfully been implemented and tested. They are implemented as independent research work named EGG (Enhanced Guided Google) [19]. The HyperClass is in progress to be finished.

Acknowledgment

We are thankful to The Omnipotent GOD for making us able to do something. We express our gratitude to our department of Electronics & Computer Engineering and the management of Indian Institute of Technology Roorkee for providing us research opportunities and motivating environment. Finally, our acknowledgement cannot end without thanking to the authors whose research work helped us in this research.

References

[1] Sergey Brin and Lawrence Page; "The Anatomy of a Large-Scale Hypertextual Web Search Engine"; Proceedings of the 7th World Wide Web Conference (WWW7), Brisbane, Australia; April 1998. http://www-db.stanford.edu/~backrub/google.html (Conference Proceedings)

[2] D Hawking and P Thistlewaite; "Methods for Information Server Selection"; ACM Transactions on Information Systems Vol. 17(1); January 1999. (Journal Publication)

[3] Softnik Technologies; "Google API Search Tool"; http://www.searchenginelab.com/common/products/gapis/docs/; 2003. (Internet Draft)

[4] Alex D; "Meta Search Engine Web services with .NET & Java"; EPFL, Lausanne; 2003 (Thesis)

[5] Choon H and Rajkumar B; "Guided Google: A Meta Search Engine and its Implementation using the Google Distributed Web Services"; International Journal of Computers and Applications Vol. 26(3) pp.181-187, ACTA Press; March 2004. (Journal Publication)

[6] Dou S, Zheng C, Qiang Y, Hua-Jun Z, Benyu Z, Yuchang L, Wei- Ying M; "Web-page classification through summarization"; Proceedings of the 27th annual international ACM SIGIR 04, conference on. Research and Development in Information Retrieval, New York, ACM Press, pp.242- 249. 2004. (Conference Proceedings)

[7] Amrish S and Keiichi N; "Hierarchical Classification of Web Search Results Using Personalized Ontologies", Proceedings of HCI International; 2005; doi=10.1.1.87.5902. (Conference Proceedings)

[8] Vogel D, Bickel S, Haider P, Schimpfk R, Siemen P, Bridges S and Scheffer T; "Classifying search engine queries using the web as background knowledge"; ACM SIGKDD Vol. 7(2) pp.117-122; 2005. (Journal Publication)

[9] Milos R and Mirjana I; "CatS: A Classification Powered Meta-Search Engine" Proceedings of Advances in Web Intelligence and Data Mining; pp.191-200; 2006. (Conference Proceedings)

[10] Debajyoti M, Pradipta B, Young-Chon K; "A Syntactic Classification based Web Page Ranking Algorithm"; Proceedings of 6th International Workshop on MSPT pp.83-92; 2006. (Conference Proceedings)

[11] Isak T, Sarah Z, Amanda S; "Using Web Search Logs to Identify Query Classification Terms"; Proceedings of IEEE International Conference on Information Technology; pp.469-474; 2007. (Conference Proceedings)

[12] Hao W, Liping F and Ling G; "Automatic Web Page Classification using various Features"; LNCS Springer Verlag Vol 5353 pp.368 -376; 2008. (Lecture Notes)

[13] Manoj M and Elizabeth Jacob; "Information Retrieval on Internet using meta-search engines: A review"; Journal of Scientific & Industrial Research Vol. 67 pp.739-746; October 2008. (Journal Publication)

[14] Keyhanipour A, Piroozmand M, Bidoki A.and Badie K; "User-based meta-search with the co-citation graph" ; Porceedings of International Conference on Applications of Digital Information and Web Technologies, pp.563-568; 2008. (Conference Proceedings)

[15] Lin G, Tang J and Wang C; "Studies and Evaluation on Meta Search Engines"; Proceedings of IEEE International Conference on Study & Evaluation of Meta Search Engines pp.191-193; 2010. (Conference Proceedings)

[16] Vishwas R, Amit T, Amit G and Yogesh K; "Re-Search & Re-Classification Algorithm – An Adaptive Algorithm for Web Technologies"; International Journal of Computer Theory & Engineering Vol. 2(6); pp.907-911; December 2010. (Journal Publication)

[17] Lovelyn R and Chandran K; "Web knowledge and Wordnet based Automatic Web Query Classification"; International Journal of Computer Applications Vol. 17(7) pp. 23-38; March 2011. (Journal Publication)

[18] Alamelu M and Santhosh K; "A Novel Approach for Web Page Classification using Optimum features"; International Journal of Computer Science and Network Security Vol.11(5) pp.252-257; May 2011. (Journal Publication)

[19] Vishwas R and Padam K; "EGG (Enhanced Guided Google) – A Meta Search Engine based on Combinatorial Keyword Search"; Proceedings of 2nd IEEE International Conference on Current Trends in Technology; December 2011. (Conference Proceedings)

[20] Vishwas R and Padam K; "SEReleC (Search Engine Result Refinement & Classification) – A Meta Search Engine based on based on Combinatorial Search and Search Keyword based Link Classification"; Proceedings of IEEE International Conference on Advances in Engineering, Sciences

and Management; March 2012. (Conference Proceeding)

Vishwas Raval: He is Assistant Professor at Charotar University of Science & Technology, Changa Gujarat (INDIA). He is MTech in Computer Science & Engineering from Indian Institute of Technology Roorkee, India. He has 7 international publications at conferences and journals of repute. He received many awards for his outstanding research work at conferences. His research areas are Web Technologies and IR.

Padam Kumar: Dr. Kumar did his B.Tech. (Electrical Engg.) and M.Tech. (Radar Systems) from IIT Delhi in 1970 and 1972 respectively. He diversified himself into Computer Science area and did his Ph.D. in the use of Functional Programming Languages in Multiprocessing Systems from University of Roorkee in 1990. Currently, he is Professor and Head of Electronics & Computer Engineering at Indian Institute of Technology Roorkee. His research areas are Parallel Processing, Multiprocessor Networks, Load Balancing and Scheduling, Grid Computing, Cloud Computing, Information Retrieval and Real Time Systems.

