www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 3 Issue 7 July, 2014 Page No. 7163-7172

An Advanced Approach for Ranking in Query Recommendation

Rinki Khanna[1], Asha Mishra[2]
[1]M.Tech Scholar, B.S Anangpuria Institute of Technology & Management, Faridabad
rinkikhanna89@yahoo.co.in
[2]Assistant Professor, B.S Anangpuria Institute of Technology & Management, Faridabad
asha1.mishra@gmail.com

Abstract- Search engines are programs that search documents for specified keywords and return a list of the documents where the keywords were found. They return long list of ranked pages, finding the relevant information related to a particular topic is becoming increasingly critical and therefore, Search Result Optimization techniques come in to play. In this work an algorithm has been applied to recommend related queries to a query submitted by user. Query logs are important information repositories to keep track of user activities through the search results. Query logs contain attributes like query name, clicked URL, rank, time. Then the similarity based on Keyword and Clicked URL's is calculated. Clusters have been obtained by combining the similarities of both keyword and clicked URL's to perform query clustering. Most favored queries are discovered within every query cluster. The proposed result optimization system presents a query recommendation scheme towards better information retrieval to enhance search engine effectiveness to a large scale.

Keywords- World Wide Web, Information Retrieval, Search Engine, Query Log, Query Clustering, Ranking Algorithm.

1 INTRODUCTION

1.1 WEB

The World Wide Web abbreviated as WWW or W3 commonly known as the Web is a system of interlinked hypertext documents accessed via the Internet. With a web browser, one can view web pages that may contain text, images, videos, and other multimedia, and navigate between them via hyperlinks.

Web developed three essential technologies:

- A system of globally unique identifiers for resources on the Web later known as Uniform Resource Locator (URL) and Uniform Resource Identifier (URI).
- The publishing language Hyper Text Markup Language (HTML)
- The Hypertext Transfer Protocol (HTTP).

1.2 FUNCTION

The Internet is a global system of interconnected computer networks. It is one of the services that run on the Internet. It is a collection of text documents and other resources, linked by hyperlinks and URLs, usually accessed by web browsers from web servers. The Web can be thought of as an application running on the Internet.

Viewing a web page on the World Wide Web normally begins either by typing the URL of the page into a web browser or by following a hyperlink to that page or resource. The web browser then initiates a series of communication messages, behind the scenes, in order to fetch and display it.

1.3 WEB MINING

It is the application of data mining techniques to discover patterns from the Web. According to analysis targets, web mining can be divided into three different types.

1.3.1 WEB CONTENT MINING

Mining, extraction and integration of useful data, information and knowledge from Web page contents.

1.3.2 WEB STRUCTURE MINING

It is the process of using graph theory to analyze the node and connection structure of a web site. According to the type of web structural data, it can be divided into two kinds:

- Extracting patterns from hyperlinks in the web: a hyperlink is a structural component that connects the web page to a different location.
- Mining the document structure: analysis of the tree-like structure of page structures to describe HTML or XML tag usage.

1.3.3 WEB USAGE MINING

Web Usage Mining (WUM) is a part of Web Mining, which, in turn, is a part of Data Mining [1]. As Data Mining involves the concept of extraction of meaningful and valuable information from large volume of data, it involves mining the usage characteristics of the users of Web Applications. This extracted information can then be used in a variety of ways such as improvement of the application, checking of fraudulent elements etc.

It is the process of extracting useful information from server logs and finding out what users are looking for on the Internet [2]. Some users might be looking at only textual data, whereas some others might be interested in multimedia data[3].

1.4 WEB USAGE MINING PROCESS

The main processes in Web Usage Mining are:

1.4.1 PREPROCESSING: Data preprocessing describes any type of processing performed on raw data to prepare it for another processing procedure. Commonly used as a preliminary data mining practice, data preprocessing transforms the data into a format that will be more easily and effectively processed for the purpose of the user.

1.4.2 PATTERN DISCOVERY: Web Usage mining can be used to uncover patterns in server logs but is often carried out only on samples of data. The mining process will be ineffective if the samples are not a good representation of the larger body of data. The following are the pattern discovery methods.

- Statistical Analysis
- Association Rules
- Clustering
- Classification
- URL Ranking

1.4.3 PATTERN ANALYSIS :This is the final step in the Web Usage Mining process. After the preprocessing and pattern discovery, the obtained usage patterns are analyzed to filter uninteresting information and extract the useful information. The methods like SQL (Structured Query Language) processing and OLAP (Online Analytical Processing) can be used.

1.5 WEB USAGE MINING APPLICATIONS

1.5.1 LETIZIA

Letizia is an application that assists a user browsing the Internet. As the user operates a conventional Web browser such as Mozilla, the application tracks usage patterns and attempts to predict items of interest by performing concurrent and autonomous exploration of links from the user's current position. The application uses a best-first search augmented by heuristics inferring user interest from browsing behavior.

1.5.2 WEB SIFT

The Web SIFT (Web Site Information Filter) system is another application which performs Web Usage Mining from server logs recorded in the extended NSCA format which is quite similar to the combined log format [8]. The preprocessing algorithms include identifying users, server sessions, and identifying cached page references through the use of the referrer field. It identifies interesting information and frequent item sets from mining usage data.

1.6 SEARCH RANKING

The Internet provides a rich source of information. With the Internet consisting of over 11.5 billion websites, it can be rather difficult to find subject specific information. To assist with this problem, search services were designed and provided Internet users with a fast and free way of finding specific information [1]. When a user executes a search on a search engine, the search query is used to search the database in order to retrieve and display the relevant

websites. Search engines use a mathematical algorithm to determine the relevancy of a website, and then rank the websites accordingly.

1.6.1 IMPORTANCE OF SEO

The Internet has a large number of users spending vast amount of money. Most of these users turn to search engines to find theproduct or service they are looking for. However between 60% and 73% of search engine users do not look beyond the first page of search results. Unfortunately, with many websites competing for these top results, getting listed on the first result page is not an easy Endeavour. SEO considers how search engines work, what people search for, the actual search terms or keywords typed into search engines and which search engines are preferred by their targeted audience.

1.7 MOTIVATION

Query recommendation is an essential ingredient for a user-oriented search engine. A common fact in search engine is that a user often needs multiple iterations of query refinement to find the desired results from a search engine [4]. This is partially because search queries are often extremely concise and therefore it conveys users search intent to the search engine.

Information overload: Too much information to sift/browse through in order to find desired information. Most information on Web is actually irrelevant to a particular user.

Query recommendation is thus a promising direction for improving the usability of search engines. The explicit task of query recommendation is to help users formulate queries that better represent their search intent during search interactions. Today, many Web applications are applying Web usage mining techniques to predict users' navigational behavior by automatically discovering the access patterns from one or more log files, but none have used them for search engine's result optimization.

1.8 OBJECTIVE

The purpose of web log mining is to improve the performance of the search engine by utilizing the mined knowledge. So objective of the proposed system is to optimize the search engine's result using query recommendation and rank optimization by improving their page ranks and thus increasing the relevancy of the pages according to user's feedback. The approach also recommends the user with a set of similar and most popular user queries so as to make search more efficient than the previous one.

1.9 APPROACH USED

The approach which is used here is to optimize the results returned by a search engine by improving the relevancy of the pages according to the user feedback [1]. To achieve this, the method first pre-mines the log using a novel similarity function to perform query clustering and then discovers the sequential order of clicked URLs in each cluster. The outputs of both mining processes are utilized to return relevant pages to the user as well as recommending him with a popular query. Ranks of the URL's of queries recommended will be updated using proposed formula.

1.10 WORKFLOW

Main Aspect of proposed application is to cluster the queries by finding similarities based on keywords and clicked URL's. In order to achieve the same firstly, Query log will be created, which contains attributes like query name, clicked URL, rank, time. Secondly, Similarities are calculated on the basis of query keywords as well as their clicked URL's [7]. Further, clusters have been obtained by combining the similarities of both keywords and clicked URL's. Once query clusters are formed, next step is to find a set of favored queries from each cluster.

A query is said to be favored which occupies a major portion of the search requests in a cluster. It is used to recommend the user with the most famous query along with many similar queries for a better search.

The final approach is to be carried out by re-ranking the search result list by modifying the already assigned rank score of the web pages. The rank updater improves the relevancy of a web page based on its access history [7]. This method not only discovered the related queries but also rank them according to a similarity measure. Finally the method has been evaluated using real data sets from the search engine query log.

On the other hand, users typically submit very short queries to the search engine, and short queries are more likely to be ambiguous. From a study of the log of a popular search engine, it concludes that most queries are short and imprecise. Users searching for the same information may phrase their queries differently. Often, users try different queries until they are satisfied with the results. In order to formulate effective queries, users may need to be familiar with specific terminology in a knowledge domain. This is not always the case: users may have little knowledge about the information they are searching, and worst, they could not even be certain about what to search for [4]. The idea is to use these expert queries to help non-expert users. In order to overcome these problems, some search engines have implemented methods to suggest alternative queries to users. Their aim is to help the users to specify alternative related queries in their search process. Typically, the list of suggested queries is computed by processing the query log of the search engine, which stores the history of previously submitted queries and the URL's selected in their answers. But it is one main fact that there may be more than one page for the related query and they also shows almost the same thing, then what to do? Then to overcome this problem the page rank updater is used which updates the rank of the page with every click of the related URL of the user [9]. Bt it does not only depends on the page rank updater, it also perform the task by calculating the time, weight, threshold value, that means by combining all these most relevant query can be found.

Search Engine is designed for searching the information on World Wide Web. Results are generally presented in a list of result often called SERP's or Search Engine Result Page. The Internet is a global data communications system. It is a hardware and software infrastructure that provides connectivity between computers. In contrast, the Web is one of the services communicated via the Internet [10]. It is a collection of interconnected documents and other resources, linked by hyperlinks and URLs (Uniform Resource Locator) or (Uniform Resource Identifier) URI which also specifies where the identified Resource is available and the protocol for retrieving it.

WORKING OF SEARCH ENGINE

Every engine relies on a crawler Module to provide the grist for its operation. Crawlers are small programs that browse the Web on the search engine's behalf, similarly to how a human user would follow links to reach different pages. The programs are given a starting set of URLs, whose pages they retrieve from the Web. The crawlers extract URLs appearing in the retrieved pages, and give this information to the crawler control module. This module determines what links to visit next, and feeds the links to visit back to the crawlers. The crawlers also pass the retrieved pages into a page repository. Crawlers continue visiting the Web, until local resources, such as storage, are exhausted [4] .Once the search engine has been through at least one complete crawling cycle; the crawl control module may be informed by several indexes that were created during the earlier crawl. Figure 1 shows the general search engine architecture.

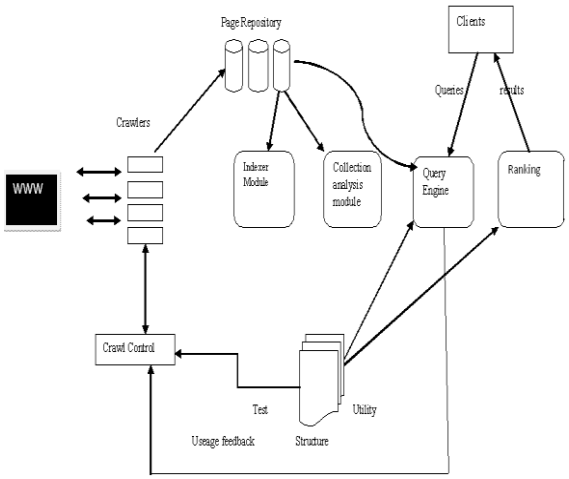

Figure 1: general search engine architecture

- The indexer module extracts all the words from each page, and records the URL where each word occurred. The result is a generally very large that can provide all the URLs that point to pages where a given word occurs. The table is of course limited to the pages that were covered in the crawling process. The collection analysis module is responsible for creating a variety of other indexes [10]. The utility index is created by the collection analysis module. The collection analysis module may use the text and structure indexes when creating utility indexes. During a crawling and indexing run, search engines must store the pages they retrieve

from the Web. The page repository represents this possibly temporary collection. Sometimes search engines maintain a cache of the pages they have visited beyond the time required to build the index. This cache allows them to serve out result pages very quickly to providing basic search facilities.

- The query engine module is responsible for receiving search requests from users. The engine relies heavily on the indexes, and sometimes on the page repository. Because of the Web's size, and the fact that users typically only enter one or two keywords, result sets are usually very large [10]. The ranking module therefore has the task of sorting the results such that results near the top are the most likely ones to be what the user is looking for.

2 PROPOSED WORK

2.1 PROBLEM STATEMENT

The major problem with Web Mining in general and Web Usage Mining in particular is the nature of the data they deal with. With the upsurge of Internet in this millennium, the Web Data has become huge in nature and a lot of transactions and usages are taking place by the seconds. Apart from the volume of the data, the data is not completely structured. It is in a semi-structured format so that it needs a lot of preprocessing and parsing before the actual extraction of the required information.

The problem of improving search engine results and obtaining the desired information from this huge amount of web contents has been processed by different ways such as clustering the search engine results in specific topics so the user can find the required results in selected category of search results. Although, the user doesn't use the proper search words or search query while searching so this leads to a problem of getting un-required results and the user have to be familiar with specific terminology in a knowledge domain. This is not always the case of many users; they have only a little background about the information they are searching and unfortunately they didn't get the required results.

In order to overcome this problem, it's not enough to use clustering search results method because the problem is not in obtaining the huge results but it's in the keywords used in searching are not strongly related.

The previous propose work is to cluster similar queries to recommend URLs to frequently asked queries of a search engine. They use four notions of query distance: based on keywords or phrases of the query; based on string matching of keywords; based on common clicked URLs; and based on the distance of the clicked documents in some pre-defined hierarchy.

2.2 PROPOSED ARCHITECTURE

The notion of query recommendation has been a subject of interest since many years. A number of researchers have discussed the problem of finding relevant search results from the search engines. Relevant query recommendation research is mainly based on previous query log of the search engine, which contains the history of submitted query and the user selected URLs. Figure 2 shows the working of proposed architecture.

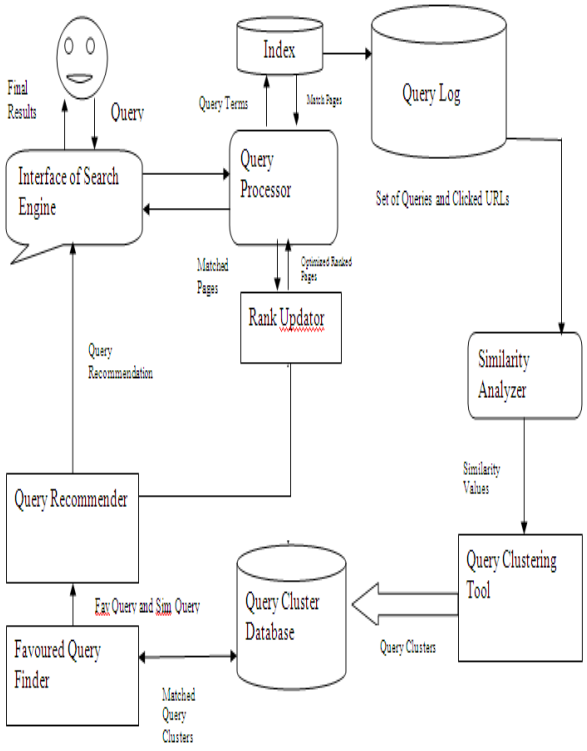

Figure 2: proposed architecture

Only a part of popular queries have sufficient log information for mining their common clicked URLs while distance matrices between most queries from real query logs are very sparse. As a result, many queries with semantic

similarity might appear orthogonal in such matrices. However, the fact that similar queries are submitted by different users in most of case, will also lead to serious problem. This is because the support of a rule increases only if its queries appear in the same query session, and thus they must be submitted by the same user. Query expansion is also adopted by search engines to recommend related queries. Its idea is to reformulate the query such that it gets closer to the term weight vector space of the documents the user is looking for. This approach aims at construction of queries rather than recommend previous registered queries in real log. Search engines are using some kind of optimization on their search results but they are not much beneficial due to the problems of finding the required information within search results. Hence, a mechanism needs to be introduced which gives prime importance to the information needs of users. Query log that keeps record of user queries on the basis of occurrence of query in the query cluster which is formed by clustering similar queries on the basis of keywords and clicked URLs is proposed and optimizes the rank values of returned web pages according to the favored query finder related to the search and returning the desired relevant pages in the top of the search result list.

User landing pages are those where users finally end up after post-query navigation to generate query suggestions. For each landing page of a user submitted query, they identify queries from query logs that have these landing pages as one of their top ten results. These queries are then used as suggestions. An algorithm is proposed based on hitting time on the Query-URL bipartite graph derived from search logs. Starting from a given initial query, a sub graph is extracted from the Query-URL bipartite using depth first search. A random walk is then conducted on this sub graph and hitting time is computed for all the query nodes. Queries with the smallest hitting time are then used as suggestions.

The potential clusters of queries are retrieved and then the most popular queries in each cluster are found. Each cluster entries are mined to extract sequential patterns of pages accessed by the users. The outputs of mining processes are utilized to return relevant results with popular historical queries.

The proposed system works as follow. The prime feature of the system is to per-form query clustering by finding the query similarity between the two queries, based on user query keywords and clicked URLs. After that, clusters are generated with the help of query clustering tool. This tool is used to cluster user queries using query logs built by search engines which in result produce query clusters. Once query clusters are formed, next step is to find a set of favored queries (which are based on the related keywords and also the related URL) from each cluster. Favored query are those that occupy a major portion of the whole search request in a cluster. Once favored queries from their query clusters are identified, next step is to optimize the user search by recommending him with most favored query related to his search and returning the desired relevant pages in the top of the search result list and to update the rank of the page or the related URL's.

2.2.1. QUERY LOGS

Query log has been a popular data source for query recommendation. Query logs are repositories that record all the interactions of users with a search engine for gaining insight into how a search engine is used and what the users' interests are. Since they form a complete record of what users searched for in a given time frame. Depending on the specifics of how the data is collected, typically logs of search engines include the following entries:

- User IDs,
- Query q issued by the user,
- URL selected by the user
- Rank of the URL clicked for the query
- Time at which the query has been submitted for search.

Table 1: Query Log

USER ID	QUERY	URL	RANK	TIME
Admin	data mining	www.dming.com	30	12:00
Admin	data ware	www.dming.com	29	12:00

	housing			
Admin	data mining	www.google.com	30	12:00
Admin	data ware housing	www.dwarehousing.com	14	12:00
Admin	search engine	www.dming.com	13	12:00

2.2.2. QUERY SIMILARITY ANALYZER

If two documents share common keywords, then they are thought to be similar to some extent. The approach of this module is based on two criteria: Similarity based on Query Keywords and Similarity based on Clicked URL's.

(a) SIMILARITY BASED ON QUERY KEYWORDS

If two user queries contain the same or similar terms, they denote the same or similar information needs. The following formula is used to measure the content similarity between two queries.

$$\mathrm{Sim}(p,q) = \frac{|\mathrm{KW}(p,q)|}{|\mathrm{kw}(p)\ \mathrm{U}\ \mathrm{kw}(q)|} \quad (1)$$

Where kw (p) and kw (q) are the sets of keywords in the queries p and q respectively, KW (p, q) is the set of common keywords in two queries.

For Example: p = "data mining" and q = "data warehousing"

Sim(p,q) = 2/(2+2) = 2/4 = 0.5

(b) SIMILARITY BASED ON CLICKED URLS

Two queries are considered same if they lead to the selection of same documents. If two queries p and q share a common document d, then similarity value is ratio of the total number of distinct clicks on d with respect to both queries and total number of distinct clicks on all the documents accessed for both the queries .If more than one document is shared, then numerator is obtained by summing up the documents clicks of all common documents.

The following formula dictates the similarity function based on documents clicks.

$$\mathrm{Sim}_{\mathrm{clickURL}}(p,q) = \frac{\sum \mathrm{LC}(p,di) + \mathrm{LC}(q,di)}{\sum \mathrm{LC}(p,xi) + \mathrm{LC}(q,xi)} \quad (2)$$

Where LC (p, d) and LC (q, d) are the number of clicks on document d corresponding to queries p and q respectively. CD (p) and CD (q) are the sets of clicked documents corresponding to queries p and q respectively.

For Example : Assuming same p and q queries.

Sim(p,q) = (30 + 14)/(60 + 14) = 44/74 = 0.59

(c) COMBINED SIMILARITY MEASURE

The two measures have their own advantages.By using first measure queries of similar keywords can be grouped together. By using second measure similarity based on clicked URL's is calculated. Both measures can be combined. In the below equation @ and β are constants with 0<=1 and @+β=1. It is better to combine them in a single measure. A simple way to do it is to combine both measures linearly as follows:

$$\mathrm{Sim}_{\mathrm{combines}}(p,q) = \alpha.\,\mathrm{Sim}_{\mathrm{Keyword}}(p,q) + \beta.\,\mathrm{Sim}_{\mathrm{clickURL}}(p,q) \quad (3)$$

Where α and β are constants with 0<=α(and β)<=1 and α+β=1

The values of constants can be decided by the expert analysts depending on the importance being given to two similarity measures. In the current implementation, these parameters are taken to be 0.5 each.

Sim(p,q) = (0.5*0.5) + (0.5*0.59) = 0.25 + 0.295 =0.545

2.2.3 CLUSTERING ALGORITHM

Another question involved is the clustering algorithm proper. There are many clustering algorithms available to us. The main characteristics that guide our choice are the following ones:

The algorithm should not require manual setting of the resulting form of the clusters, e.g. the number of clusters. It is unreasonable to determine these parameters manually in advance. The algorithm should filter out those queries with low frequencies. Since query logs usually are very large, the algorithm should be capable of handling a large data set within reasonable time and space constraints.

Due to the fact that the log data changes daily, the algorithm should be incremental.

Algorithm :Query_Clustering(Q,α,β,τ)

Given : A set of n queries and corresponding clicked url'sstored in an array

A set of n queries and corresponding clicked url's stored in an array Q[q1,URL1…..URL m] 1<=i<=n
α=β=0.5
Similarity Threshold τ
Output : A set C={C1,C2….Ck} of k query clusters
//Start Algorithm
K=1; // k is the number of clusters
For (each query p in Q)
Set Cluster Id(p) = Null; //Initially No query is clustered
For (each p € Q)
{
Cluster Id(p) = Ck;
Ck ={ p };
For each q € Q such that p ≠ q
{
If( )
Set ClusterId(q) = Ck;
Ck = Ck U {k};
Else
Continue;
} // End For
K=K+1;
} //End Outer For
Return Query Cluster Set C;

(a) Clustering tool

In support of the clustering process, this tool is used to cluster user queries using query clustering tool built by search engines and for this it assigns query cluster database log entries, which in result produces matched query clusters and favored queries as shown in Figure 3.

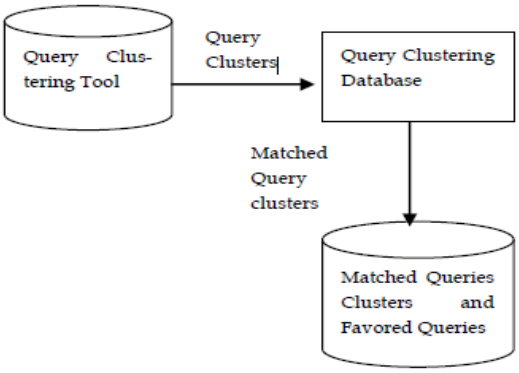

Figure 3: clustering tool

An important component in this work is the concept of clustering queries in user logs. The query clustering is a preprocessing phase and it can be conducted at periodical and regular intervals. Even though the need for query clustering is somewhat new, there have been general studies on document clustering, which are similar to query clustering. However, it is not reasonable to easily apply any document clustering algorithms to queries due to their own characteristics. It is usually observed that queries submitted to the search engines typically are very short, so the clustering algorithm should be suitable for short texts. Additionally query logs are usually very large, the method should be able of handling a large data set in reasonable time and space constraints. Furthermore, due to the fact that the log data changes daily, the method should also be incremental.

2.2.4 FAVORED QUERY FINDER

When query clusters are formed, another phase is to find a set of favored queries from each cluster. Query is said to be favored query that occupies the foremost portion of the search requests in a cluster. The process of finding favored queries is shown in Figure 4 which find the favored queries in one cluster. The method is applied in every clusters and output is stored in the Query Cluster Database.

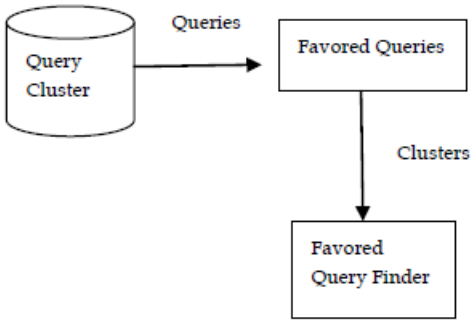

Figure 4: Favored Query finder

2.2.5 RANK UPDATER

This module takes input from the query processor in the form of matched documents of the user query and applies update on the rank score of these pages. This module works online at the query time.

Two popular algorithms were introduced in 1998 to rank web pages by popularity and provide better search results. They are:

•HITS (Hypertext Induced Topic Search)

•Page Rank

HITS was proposed by Jon Kleinberg who was a young scientist at IBM in Silicon Valley and now a professor at Cornell University.

Page Rank was proposed by Sergey Brin and Larry Page, students at Stanford University and the founders of Google. The Web's hyperlink structure forms a massive directed graph.

Hyperlinks into a page are called in link and point into nodes and out links point out from nodes.

Page Rank is a numeric value that represents the importance of a page present on the web.

When one page links to another page, it is effectively casting a vote for the other page. More votes implies more importance. Importance of the page that is casting the vote determines the importance of the vote. A web page is important if it is pointed to by other important web pages.

Google calculates a page's importance from the votes cast for it. Importance of each vote is taken into account when a page's Page Rank is calculated. Page Rank is Google's way of deciding a page's importance. It matters because it is one of the factors that determine a page's ranking in the search results.

FORMULA USED FOR RANK UPDATER

The popularity from the number of inlinks and outlinks is recorded as *Win*(*v*,*u*) and *Wout*(*v*,*u*) , respectively. *Win*(*v*,*u*) given in eq. (3) is the weight of *link*(*v*, *u*) calculated based on the number of inlinks of page *u* and the number of inlinks of all reference pages of page *v*.

$$W^{in}_{(v,u)} = \frac{I_u}{\sum_{p \in R(v)} I_p}$$

(4)

Where *Iu*and *Ip*represent the number of inlinks of page *u* and page *p*, respectively. *R* (*v*) denotes the reference page list of page *v*. *Wout*(*v*,*u*) given in eq. (4) is the weight of *link*(*v*, *u*) calculated based on the number of outlinks of page *u* and the number of outlinks of all reference pages of page *v*.

$$W^{out}(_{v,u}) = \frac{O_u}{\sum_{p \in R(v)} O_p}$$

(5)

Where *Ou*and *Op* represent the number of outlinks of page *u* and page *p*, respectively. *R* (*v*) denotes the reference page list of page *v*.

Considering the importance of pages, the original PageRank formula is modified as

$$PR(u) = (1-d) + d \sum_{v \in B(u)} PR(v) W^{in}_{(v,u)} W^{out}(_{v,u})$$

(6)

4.2.5.2 Proposed Formula

PR(u) =(1-d)+d ∑ PR(v) * W(in)*W(out)*D(v,u)

(7)

Introduced D in existing formula, D refers here with the number of duplicates.

D(v,u) = D (u)/D (v)

Here D(u) and D (p) are the no. of duplicates.

Introducing number of duplicates in existing formula, it can modify rank more efficiently. As in search engine optimization if number of links to navigate to other pages of same website then that website is considered better than other websites, using this approach in this thesis it is proposed that by including number of duplicates rank update can be more effective.

III CONCLUSION AND FUTURE WORK

CONCLUSION

In this paper, Architecture of result optimization system has been proposed based on query log for implementing effective web search. The most significant feature is that the result optimization method is based on users' feedback, which deter-mines the relevance between Web pages and user query words. The returned pages with better page ranks are directly mapped to the user feedbacks and dictate higher relevance than pages that exist in the result list but are never accessed by the user. Hence, the time user spends for looking for the required information from search result list can be reduced and the more important Web pages can be presented. As the system based on click through data in query log and semantic search has been proposed for implementing effective web search, the most important feature is that the proposed approach is based on users' behavior, which determines the relevance between Web pages and user query words.

The results obtained from practical evaluation are quite effective in respect to reduced search space and enhanced the use of interactive web search engines. As the future work, we can apply a more relevant formulas and algorithms to update the query more efficiently.

A novel approach based on query log analysis is proposed for implementing effective web search with improved page ranking. The most important feature is that the result optimization method is based on users' feedback, which determines the relevance between Web pages and user query words. Since result improvement is based on the analysis of query logs, the recommendations and the returned pages are mapped to the user feedbacks and dictate higher relevance than the pages, which exist in the result list but are never accessed by the user. By this way, the time user spends for seeking out the required information from search result list can be reduced and the more relevant Web pages can be presented.

The results obtained from practical evaluation are quite promising in respect to improving the effectiveness of interactive web search engines.

FUTURE WORK

Further investigation on mining log data deserves more of our attention. Further study may result in more advanced mining mechanism which can provide more comprehensive information about relevancy of the query terms and allow identifying user's information need more effectively.

REFRENCES

[1][Nee10] A. K. Sharma, NeelamDuhan, Neha Aggarwal, Rajang Gupta. Web Search Result Optimization by Mining the Search Engine logs. Proceedings of International Conference on Methods and Models in Computer Science (ICM2CS-2010), JNU, Delhi, India, Dec. 13-14, 2010.

[2] [Sri96] Spirant R., and Agawam R. "Mining Sequential Patterns: Generalizations and performance improvements", Proc. of 5th International Conference Extending Database Technology (EDBT), France, March 1996.

[3] [Bor98] A. Birchers, J. Her locker, J. Konstantin, and J. Riel, "Ganging up on information overload," Computer, Vol. 31, No. 4, pp. 106-108, 1998.

[4] [Ame2000] B. Amen to, L. Tureen, and W. Hill, "Does Authority Mean Quality? Predicting Expert Quality Ratings of Web Documents", In Proceedings of 23th International ACM SIGIR, pp. 296-303, 2000

[5] [Bee2000] Beeferman and Berger A., 2000. Agglomerative clustering of a search engine query log. In Proceedings of the 6th ACMSIGKDD International Conference on Knowledge Discovery and Data Mining, (August). Acme Press, New York, NY, 407–416.

[6] [Zha2000] D. Zhang and Y. Dong, “An Efficient Algorithm to Rank Web Resources,” In Proceedings of 9th International World Wide Web Conference, pp. 449-455, 2000.

[7] [Wen01] J. Went, J. Mie, and H. Zhang. Clustering user queries of a search engine. In Proc. at 10th InternationalWorld Wide Web Conference. W3C, 2001.

[8][Bem01] Bernard J. Jansen and Undo Pooch. A review of web searching studies and a framework for future research. J. Am. Soc. Inf. Sci. Technol., 52(3):235–246, 2001.

[9] [Ara01] A. Aras, J. Cho, H. Garcia-Molina, A. Peace, and S. Raghavan, “Searching the Web,” ACM Transactions on Internet Technology, Vol. 1, No. 1, pp. 97-101, 2001

[10] [Her01] M.R. Her zinger, “Hyperlink Analysis for the Web,” IEEE Internet Computing, Vol. 5, No.1, pp. 45-50, 2001.