Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 3, Issue. 4, April 2014, pg.989 – 995

SURVEY ARTICLE

A Survey on Advanced Page Ranking in Query Recommendation

Rinki Khanna[1], Asha Mishra[2]

[1]M.Tech Scholar, B.S Anangpuria Institute of Technology & Management, Faridabad

[2]Assistant Professor, B.S Anangpuria Institute of Technology & Management, Faridabad

[1] rinkikhanna89@yahoo.co.in; [2] 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 Retrieve; Search Engine; Query Log; Query Clustering; Ranking Algorithm

I. Introduction

With the development in information technology, the web [1] has turned out to be a vast information repository covering almost every area, in which a human user could be involved. In spite of recent advances in web search engine technologies, there are still many situations in which user is presented with undesired and non- relevant pages in the top most results of the ranked list. Search engine often have difficulties in forming a concise and precise representation of the response pages corresponding to a user query. Providing a set of web pages based on user query words is not a big problem in search engine. The difficulty arises at the user end as he has to sift through the long result list, to find his desired content. This problem is referred to as Information Overkill problem [2].

The architecture [3] of the search engine is shown in Figure1.

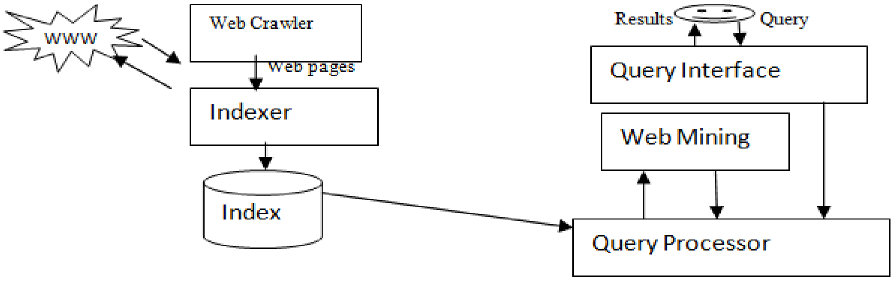

Fig 1: Architecture of Search Engine.

There are 3 components in search engine known as Crawler, Indexer and Ranking mechanism. The crawler is also called as robot that navigates the web and downloads the web pages. The downloaded pages are transferred to an indexing module and erect the index based on the keywords in individual pages. When a query is being floated by a user, it means the query transferred in terms of keywords on the interface of a search engine, the query mainframe section examine the query keywords with the index and precedes the URL's of the pages to the client. But presenting the pages to the client a ranking mechanism is completed by the search engines to present the most relevant pages at the top and less significant pages at the bottom.

A. *Query Logs*

The log keeps user's queries and their clicks as well as their browsing activities. The typical logs[4] of search engine include the following entries:

1) User IDs
2) Query q issued by the user
3) URL u clicked by the user
4) Rank r of the URL u clicked for the query q
5) Time t at which the query has been submitted

The information contained in query logs can be used in many ways[5,6], example to provide context during search, to classify queries. Query log is shown in Table 1.

User Id	Query	Clicked URL	r	Time
Admin	Data Mining	www.dming.com	6	12:10
Admin	Data ware housing	WWW.dming.com	5	8:30
Admin	Data Mining	www.google.com	5	11:10

Table 1 : Query Logs

II. Proposed Optimization System

A. *Proposed Architecture*

The proposed architecture of the optimization system is shown in Figure 2 which consists of the following components:

- Query Similarity Analyzer
- Query Clustering Tool
- Favored Query Finder
- Query Recommender
- Rank Updater

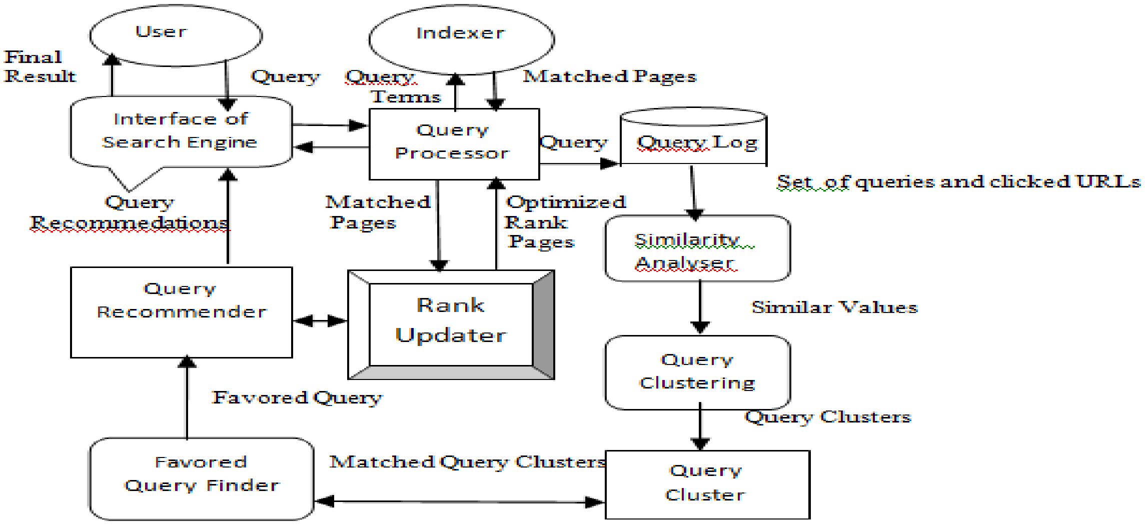

Fig 2: Proposed Architecture

When user submits a query on the search engine interface, the query processor component matches the query terms with the index repository of the search engine and gives a list of matched documents in reply. User browsing behavior including the submitted queries and the clicked URLs get stored in the logs and are analyzed continuously by the similarity analyzer module, the result of which is forwarded to the query clustering tool to generate groups of queries based on their similarities. Favored query finder extracts most popular query from each cluster and stores them for future reference. The rank updater component works online and takes as input the matched documents retrieved by query processor.

1) Query Similarity Analyzer

In Proposed architecture the next step is finding the query similarity. It has wide range of applications in information retrieval in query recommendation. Traditional approaches [7] make the use of keywords extracted from documents. 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 queries contains the similar terms, they denote the same information needs. The given formula is used to measure the similarity between two queries.

$$Sim(p,q)=\frac{|KW(p,q)|}{|kw(p)\cup kw(q)|}$$

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

b) Similarity based on Clicked URL’s

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 given formula finds similarity based on clicked URL's

$$\text{Sim}_{\text{click URL}}^{(p,q)} = \frac{\sum_{di\in CD(p)\cap CD(q)} LC(p,di)+LC(q,di)}{\sum_{xi\in CD(p)\cup CD(q)} LC(p,xi)+LC(q,xi)}$$

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

c) Combined Similarity Measure

The two measures [8] 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.

$$\text{Sim}_{\text{combines}}(p,q) = \alpha \cdot Sim_{keyword}\,(p,q) + \beta \cdot Sim_{click\ URL}(p,q)$$

Example-Similarity Calculations

Suppose we want to find the similarity based on queries keywords between the first and third queries of table 1. Let q1=Data Mining and q3= Data Mining.

Sim(q1,q2)= 2/4= 0.5

Similarly to find the similarity between first and second queries of table 1. Let q1=Data Mining and q2= Data ware housing.

Sim(q1,q2)= 1/5=0.2

Suppose we want to find the similarity based on clicked Urls between the first and second queries of table 1. Let q1=Data Mining and q2= Data ware housing.

$\text{Sim}_{\text{Click url}}$(q1,q2)= 6+5/11+16= 0.4

$\text{Sim}_{\text{combined}}$=(0.5)(0.2)+(0.5)(0.4)=0.3

II) Query Clustering

Query clustering is a technique for discovering similar queries on search engine. This module uses the algorithm where each run of the algorithm computes k clusters. As query logs are dynamic in nature, query clustering [9] algorithm should be incremental in nature.The algorithm given below works [9] in following steps. First all queries are not assigned to any clusters. Each query is assigned against all other queries by using combined similarity measure. If the similarity value is greater then prespecifiedthreshold value (T), then queries are grouped in to same cluster. The process is repeated until all queries grouped in to any one of the clusters. The returned cluster is stored in the query cluster database with related query keywords and clicked URL's.

Algorithm For Clustering Queries

```
Given: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)
{
    $Sim(p,q)= \frac{|KW(p,q)|}{|kw(p) \cup kw(q)|}$
    $Sim_{click\ URL}^{(p,q)} = \frac{\sum_{di \in CD(p) \cap CD(q)} LC(p,di)+LC(q,di)}{\sum_{xi \in CD(p) \cup CD(q)} LC(p,xi)+LC(q,xi)}$
    $Sim_{combines}(p,q)= \alpha \cdot Sim_{keyword}(p,q) + \beta \cdot Sim_{click\ URL}(p,q)$
    If($sim_{combined}(p,q) \geq \tau$) then
    Set ClusterId(q) =  Ck;
    Ck = Ck U {q};
    Else
    Continue;
    } // End For
    K=K+1;
    } //End Outer For
    Return Query Cluster Set C;
```

III) Favored Query Finder

Once query clusters are formed, next step is to find a set of favored queries. This is the query submitted by most of the users. The process of finding favored queriesis given in algorithm which finds the favored queries in one cluster.

Algorithm: Favored Query _Finder()

```
I/P: A cluster of Queries.
O/P: True or False

1.          Queries which are exactly same club them and make a set of <query,IP addresses> pairs.
2.          For (each q € Cluster)
Calculate the weight of query as :
Wt=$\frac{No.of\ IP\ addresses\ which\ fired\ the\ query}{Total\ no.of\ IP\ addresses\ in\ that\ Cluster}$
If (wt>= threshold value) then
Return True;     //query is considered as favored query.
Else
Return False;    //  query is considered as disfavored.
```

IV) Query Recommender

It provides the user with a set of recommended queries with the famous query being highlighted. The recommended queries are those that are related to the query submitted by user and these queries are contained in the cluster of that query.

V) 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.Following are the algorithms to find improved rank and weight of a page.

a) Page Rank Algorithm

PageRank algorithm was proposed by S.Brin and L. Page [10] at Stanford University. Page Rank algorithm is used by popular search engine of today, Google. The basic concept behind google is,it combined the actual page rank value of a page with the matching text value of query and find the overall score of a page [11].It is the most frequently used algorithm for ranking the various pages. Functioning of the page rank algorithm deals with the link structure of the web page. It is based on the concept that if a page sourrounds important links towards it then the links of this page near the other page are also to be believed as imperative pages. A page gets hold of a high rank if the addition of the ranks of its back links is high.

Basic formula of a page rank is given in equation 1:

$$PR(u)= C\sum_{v \in B(u)} \frac{PR(v)}{N_v} \qquad (1)$$

In equation 1 u represents a web page, B(u) is the set of pages that point to u. PR(u) is the page rank of page u that we want to find out. PR(v) is the page rank of page v that points to page u. N_v is the number of outgoing links of page v, c denotes the factor of normalization.
The modified page rank formula is thus given in equation 2. In equation (2) d is the damping factor that is set to 0.85.

$$PR(u) = (1\text{-}d)+d \sum_{v \in B(u)} \frac{PR(v)}{N_v} \qquad (2)$$

Example of hyperlink structure of four pages A, B, C, D shown in figure 1. The page rank of four pages that is A, B, C, D can be calculated using equation 2.

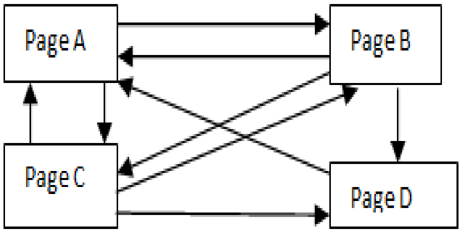

Figure 1: Hyperlink Structure for four pages.

b) Weighted Page Rank Algorithm

Weighted Page Rank was proposed by Wenpu Xing and Ali Ghorbani [12]. It is modified version of page rank algorithm. This algorithm does not divide page rank value evenly in between number of page links available in the page. It gives more value to the important pages.Each out link page gets a value proportional to its popularity (its no. of inlinks and outlinks).Importance of web pages can be determine by the help of outgoing links and incoming links to the web pages and are denoted as W^{in}(m,n) and W^{out}(m,n).Where W^{in}(m,n) is the weight of the link (m,n) calculated based upon number of inlinks of page n and the number of inlinks of all reference pages of page m. W^{in}(m,n) is shown in equation 3.

$$W^{in}(m,n)=\frac{I_n}{\sum p \in R(m) I_p} \qquad (3)$$

In equation (3) I_n and I_p are the number of incoming links of page n and page p. R(m) denotes the allusion page list of page m.
W^{out}(m,n) is shown in equation (4) is calculated based upon the number of outlinks of page n and the number of outlinks of all reference pages of page m. In equation (4) O_n and O_p are the number of outlinks of page n and page p.

$$W^{out}(m,n)=\frac{O_n}{\sum p \in R(m) O_p} \qquad (4)$$

The basic formula of weighted page rank is shown in equation (5) which is a modification of page rank formula.

$$WPR(n)= (1\text{-}d)+d \sum_{m \in B(n)} WPR(m) W^{in}(m,n) W^{out}(m,n) \qquad (5)$$

III. Conclusion and Future Work

In this paper the Proposed Architecture will work as follows. A query log is created which contains attributes like query name, Clicked url, rank , time. By using query log it will calculate similarities based on keywords and clicked url's by using formulas given in paper. Then it wll calculate combined similarity and cluster the queries on the basis of threshold value provided by user. The threshold value should be in the range of 0 to1. After this the system will generate clusters containing the similar queries.Once query clusters are formed , next step is to find a set of favored queries from each cluster. Favored query is used to recommend the user with the most famous query along with many similar queries for better search. For the favored queries the system will get all url's that belong to that query in a seprate list box and after navigating to that url rank will be updated using a proposed formula.

REFERENCES

1. A.Arasu, J.Cho, H.Gracia-olina, A.Paepcke, and S.Raghavan, "Searching the Web", ACM Transactions on Internet Technology , Vol.1 No. 1, pp.97-101, 2001.
2. A.Borchers, J.Herlocker, J.Konstanand, and J.Riedl,"Ganging up on information overload", computer, vol. 31, No. 4, pp. 106-108, 1998.
3. N.Duhan, A.K. Sharma and K.K Bhatia, "Page Ranking Algorithms: A survey", Proceedings of the IEEE International Conference on Advance Computing, 2009.
4. Edgar Meij, Marc Bron, Bouke Huurnink, Laura Hollink, and Maarten de Rijke. Learning semantic query suggestions. In 8th International Semantic Web Conference (ISWC 2009). Springer,October 2009.
5. K. Hofmann, M. de Rijke, B. Huurnink, E. Meij. A Semantic Perspective on Query Log Analysis, In Working notes for the CLEF 2009 Workshop, Cortu, Greece.
6. H. Ma, H. Yang, I. King, and M. R. Lyu. Learning latent semantic relations from clickthrough data for query suggestion. In CIKM '08: Proceeding of the 17th ACM conference on Information and knowledge management, pages 709–718, New York, NY, USA, 2008. ACM.
7. Salton,, G. and McGill, M.J. Introduction to modern information retrieval. McGraw hill-book Company, 1983.
8. A.K Sharma, Neelam Duhan, Neha Aggarwal, Ranjana Gupta. Web Search Result optimization by mining the Search Engine Query Logs. Proc. of International Conference on methods and models in Computer Science, Delhi, India, Dec.13-14, 2010.
9. Neelam Duhan, A.K Sharma."Rank Optimization and Query Recommendation in Search Engine using Web Log Mining Technique. Journal of computing. Vol 2, Issue 12, Dec. 2010.
10. S. Brin, and L. Page, "The Anatomy of a Large Scale Hypertextual Web Search Engine", Computer Network and ISDN Systems, Vol. 30, Issue 1-7, pp. 107-117, 1998.
11. Dilip Kumar Sharma, and A. K. Sharma, A Comparative Analysis of Web Page Ranking Algorithms, *International Journal on Computer Science and Engineering,08(02),* 2010, 2670-2676.
12. Wenpu Xing and Ali Ghorbani, "Weighted PageRank Algorithm", Proceedings of the Second Annual Conference on Communication Networks and Services Research (CNSR '04), IEEE, 2004.

.