Implementation of Search Engine Optimization Using 2 Phased Page Ranking Algorithm

K. SRI SAI KRISHNA [1], D. LALITHA BHASKARI [2]

[1] M. tech, Department of Computer Science & Systems Engineering,
College of Engineering, Andhra University.

[2] Professor, Department of Computer Science & Systems Engg, College of Engineering,
Andhra University, Visakhapatnam.

Abstract

In this paper we proposed a web-based searching technique. Present search engines generally handle search queries or keywords without considering user preferences or contexts in which users submit their queries. At times, user also fails to use proper keywords that represent their information need accurately. Ambiguous keywords, different needs of users at different times, and the limited ability of user to precisely express what they need have been widely recognized as one of the challenging obstacles in improving search results. Web based applications are at full stretch in today world. For a small piece of info, we can easily find it on web with the help of search engine to us on web. Arrangements of the travel-oriented phenomena, financial management, online purchases respectively we are dependent on web. To properly guide the users for their information quests on the web, search engines keep track of their queries and clicks while searching online. In this paper we are proposing a dynamic clustering algorithm will help us to group related queries together in such a way that the user can have faster access to their required links and the computational time is lesser.

Keywords: user search histories, web search, searching histories, 3 dimensional searches, web browsing history, Search engine, web search, web results, search queries.

1. Introduction

The World Wide Web has become a new communication medium with Web information access. This incorporates with informational, cultural, social and evidential values to be specific. With the existence of various Search Engines e.g. Google, Yahoo and many more, the users are tending to use them for retrieving their desired Web pages and their information. Although today's search engines can meet a general request, they cannot distinguish different users' specific needs. search engines like Google and Yahoo! These search engines also provide a very user friendly, yet simple user interfaces to pose search queries simply in terms of keywords, though the simple and user friendliness feature of a search engine fails at times to satisfy an individual's information goal. It has observed that the difficulty in finding only those which satisfy an individual's information goal increases with the keyword which has different meaning at different context. As the working of any search engine primarily based on the matching of the keywords to the desired documents to determine which Web pages will be returned given a search query. So, there are main two limitations of Keyword-based search queries. First, there are some keywords, which have different meanings in different context and hence the ambiguity of user needs to be resolved as to get proper and relevant information over the Internet. The search engines currently available and used by the users generally handle search queries without considering user preferences or contexts in which users submit their queries. Another limitation is to choose proper and relevant search terms which express the user's need the best in the given context. Ambiguous keywords used in Web queries, the diverse needs of users, and the limited ability of users to precisely express what they want to search in a few keywords have been widely recognized as a challenging obstacle in improving search quality. One of the popular approaches in recent data engineering field is encoding human search experiences and personalizing the search results using ranking optimization. This approach enhances the quality of information retrieval i.e. the quality of the search results of Web Search. The search results provided by the present search engines are primarily based on the matching of the keywords and hence another approach as result re-ranking can be seen for the refinement and quality improvement of the same well. Techniques implemented by these commercial search engines are usually confidential and not revealed to anyone, whereas many academic researchers have showed large amount of interest in the study of the process of query suggestion.

II. LITERATTURE SURVEY

This chapter deals with the Literature Review about the work done by various authors and discussed below. The paper "Combined Two Phase Page Ranking Algorithm for Sequencing the Web Pages", M. Usha, Dr.N. Nagadeepa, deals with two phased ranking algorithms where the ranking of a web page is done in two phases. In the first phase, score will be calculated based on the content relevancy and in the second phase rank will be given based on the user access time. By adding these two scores the total rank of the web page can be obtained. At last, the normalized value of each result page is sorted in descending order to get the most relevant page on the top most place. Similarity rank determines the relevance of a page with respect to query terms by counting the number of occurrences of the query terms within the web document. It gives weight based on the locality of the keyword. The result shows the most relevant pages on the top most place.

The paper “Enhancement of Web Search Engine Results Using Keyword Frequency Based Ranking”, Ms. Nilima V. Pardakhe, Prof. R. Keole deals with the problem of page ranking, in which an approach of relevant search which ranks the web pages based on the frequency or count of keywords (searched by user) is proposed. The web page containing maximum frequency or counts of keyword searched by the user is more relevant and displayed first in the list of web page links on the user screen. Every result is individually analyzed based on frequency of keywords and thus based on the user query, search results are obtained.

The paper “An improved PageRank algorithm based on web content”, Zhou Hao, Pu Qiumei, Zhang Hong, Sha Zhihao, deals with the page rank algorithm which gives the rank to the relavence pages, the pages which are analyzed by the <tittle> tag, <anchor> tag content and two of the three keywords matches. The relevance pages are ranked up.

The paper “CiteSeerX: AI in a Digital Library Search Engine”, Jian Wu, Kyle William, Hung-Hsuan Chen deals with the meta data was extracted. Header extraction and document title extraction is done. The graph was extracted by citiation graph.

The page “Modified Page Rank Algorithm: Efficient Version of Simple Page Rank with Time, Navigation and Synonym Factor” deals with the simple page ranking algorithm which gives the analyzed results by taking user usage time. it sets a rank to the pages by every 20 secs and increment the rank.

III. PROPOSED SYSTEM

In this work, a combined two-phase page ranking algorithm is used to maintain the page ordering by using the two-phase algorithm which has two phase which calculates similarity score computation and usage score computation. By these two phases the rank of the page was calculated and reordered the pages.

Advantages:

- It is well concentrated on user also.
- Improved computational speed

IV. RELATED WORK

The study of the log of a popular search engine reported that most search queries are about two terms per query. Therefore, the difficulty is that since Web users typically submit very short queries to search engines, the very small term overlap between queries cannot accurately estimate their relatedness. Given this problem, the technique to find semantically related queries (though probably dissimilar in their terms) is becoming an increasingly important research topic that attracts considerable attention. After the survey and research, it has been found that the need of having a search engine procedure or any searching technique which gives more refined and accurate search results in any of the user defined context. As the various search engines currently present in the market may or may not give the relevant or related search results. So, to fill the gap between the output of a search engine from related search results to more related and relevant search results, a technique is required. With the previous work and researches, the goal is to propose a technique or a procedure of learning the behavior of a user surfing the net over a period of time and to refine the search results using the same click-through data in the context of personalized search, one of the main components are learning user’s interest and their preferences. Many schemes for building and learning user profiles includes several schemes to figure user preferences from text documents. But the observation says that modeling user profiles or learning from text documents shows some amount of error which generally doesn’t consider the term correlations. Hence, a kind of a simple scheme is a taxonomic hierarchy, particularly generated as a tree structure, which also overcomes the drawbacks of learning from text documents, also called as the bag of words

A. Finding related keyword:

The techniques to find semantically related queries is becoming an increasingly important research topic that attracts considerable attention. Existing techniques differ from one another in terms of how to improve the naive query term-based suggestion which simple thinks that two Web queries are related if they share common terms. On the Web, recent studies are interested in using Web logs as an additional source to enrich short Web queries. There are two kinds of feature spaces commonly used in the literature, i.e., content-sensitive and content-ignorant features. Beeferman et al. [5] used single-linkage clustering to cluster related queries based on the common clicked URLs two queries share, a content-ignorant feature space. Wen et al. [13] further proposed three kinds of features to compute query to query relatedness: 1) based on terms of the query, 2) based on common clicked URLs, and 3) based on the distance of the clicked Web pages in a predefined hierarchy. The terms in a short Web query would not give reliable information, while the limitation of URL feature space is that two Web pages with different URLs may be semantically related in contents. The third features in [13] needs a concept taxonomy and requires Web pages to be classified into the taxonomy as well. Such taxonomy is not generally available. Baeza-Yates et al. [1, 2] find related queries based on the content of clicked Web pages using click frequency as a weighting scheme. Their experiments show that using the content information of a Web page (e.g., nouns) is a more accurate query enrichment way to measure query similarity than using the URL of a Web page.

B. Query-URL context

The content-based feature space, e.g., terms of a Web page, however, is not applicable, at least in principle, in settings including: non-text pages like multimedia (image) files, Usenet archives, sites with registration requirement, and dynamic pages returned in response to a submitted query and so forth. It is crucial to improve the quality of the URL (content-ignorant) feature space since it is generally available in all types of Web pages. The query-URL relationship can be represented by a bipartite graph. Finding biclique is a natural way of collecting the most related queries and URLs. A well-known problem related to biclique is the maximum clique, which is one of the most widely studied NP-complete problems in the literature [10]. Graph partitioning is an alternative for grouping which is done by cutting the set of vertices into disjoint sets. Beeferman et al. [1, 5] viewed the click-through data as a bipartite graph, and utilized an iterative, agglomerative clustering algorithm to the vertices of the graph for clustering queries and URLs, respectively. However, the selected queries may not be the best query suggestion, as the frequency is not always the best descriptor of relatedness because it does not discern the individual targeted queries. In addition, an alternative representation for query-URL data can be given by a contingency matrix whose rows correspond to queries and columns to URLs. This matrix is sparse, since the majority of queries retrieve only a small number of URLs. The elements of the matrix can be set as binary or weighted according to a measure (e.g., each entry is the probability of choosing same query and same URL).

C. Query clustering

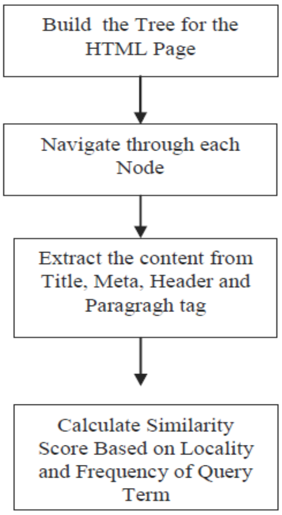
Query clustering also helps find related Web queries, which appears to be less explored than clustering. Web pages or documents [1, 2, 5, 13]. Went et al. [1, 13] proposed to cluster similar queries to recommend URLs to frequently asked queries of a search engine. They combined similarities based on query contents and user clicks, and regarded user clicks as an implicit relevance feedback but not the top ranked Web pages. The distilled search-related navigation information from proxy logs to cluster queries. The data they relied on differed from those used in the above other studies. In addition, there are three URL-based similarity measures analytically and empirically to provide better understanding of the propagation of similarity from query to query by inducing an implicit topical relatedness between queries.

V. METHODOLOGY

Search Engine is a tool enabling document search with respect to specified keywords in the web and returns a list of documents where the keywords were found. The search results are generally presented in a line of results often referred to as Search Engine Result Pages (SERPs). The html pages are collected and the source of these web pages is parsed as

Fig 5: Flow chart of Tag Analyzer Algorithm Document Object Model (DOM) tree. DOM is the interface which allows scripts and programs to dynamically access and handle all the elements such as content, structure and style of web pages. We travel through the DOM tree to identify title, Meta, Heading and paragraph tags. The title tag is an HTML component to identify the title of a document. Meta tag gives the basic information about the HTML document. The H1 tag will usually consists the title of a web document.

Algorithm 1: Tag Analyzer Algorithm


Build the Tree for the HTML Page
Navigate through each Node
Extract the content from Title, Meta, Header and Paragragh tag
Calculate Similarity Score Based on Locality and Frequency of Query Term

Step 1: Extracting content from Title and Meta Tag

```
Input the raw HTML page P to be processed
Build the tree
Navigate the nth hierarchy nodes, T is the total number of the n hierarchy
T ← number of nodes in x
Tt -<title> tag
mt - <meta> tag
for i ← 1 to M
if (Node[i]. Text = tt)
X ={x↓(i) | iε [1, n], qk}
Sc1←nq/n
End if
If (Node[i]. Text = mt)
Y={y↓(i) | i ε [1, m]}
Sc2←mq / m
end if
end for
```

STEP 2: Frequency in Heading Tags

Headlines and important segments are usually more highlighted in the body of the web page. The proposed algorithm considers Query Keyword qki that appears in header tags (H1, H2, H3… H6) is more essential than other tags. It first navigates through the entire page and fetches the content of all header tags. Then it compares the texts to search whether the Query Keyword qki appears within heading tags.

$$F(qk_i) = \sum_{j=1}^{6}(s_i f_i)$$

where fi is the frequency of appearance of qki in header i and Si is the score of header i. Similarly, to the scores are fixed to values (6, 5, 4, 3, 2,1) respectively. The score F is then normalized to the scale of [0, 1] by the following formula for header 3 the score S_3 is

where *F*min = min (*F* (*qk*1), *F* (qk2) … *F* (qkp)) and *F*max= max (*F* (qk1), *F* (qk2) … *F* (qkp)) for all values.

Step 3: PCExtracter

$$S_3 = \frac{F(qki) - F\min}{F\max - F\min}$$

The tags removed include <head>, <script>, <style>, <b>, <i> and so on. The algorithm looks at each line and creates the block using Line-block concept. Then it computes features for all blocks to decide whether they are content or not. TTR and ATTR formulas are used. If

keyword density is greater than the threshold then the algorithm adds it in the output block. After calculating the content features, the system decides whether the block is
content or not. This will be done based on the feature's values. The proposed system uses threshold methods to categorize the main content and non –content. Finally, the results are analysed. The threshold method uses standard derivation method. Threshold methods use three thresholds for TTR, ATTR and TKD. If TTR>TTR's threshold and ATTR<ATTR's threshold and TKD>=TKD's threshold then the block is main content [7]. Otherwise, the block is noise block. After extracting the more accurately main contents, page score on the basis of paragraph content can be computed.

Algorithm 2: Event Explore Algorithm

Event Explore technique is used to record the user access time of a web page. When a web page is loaded into the user's browser, timer will be triggered. Every second the timer will invoke this function. This function checks whether the user is in idle state or in active state. This verification is done by binding mouse events and keyboard events. If the user is idle continuously for 5 minutes (i.e.), if there is no event occurs on a page, then the timer will be reset. Otherwise the user access time will be computed using the timer value.

```
Start Timer
Initialize Idle State=false;
Initialize idleTimer=null;
Idletimeout=3000; // 5 mins;
If (page.event = true)
Idletimer.reset;
Else
Idletimer=idletimer+1
If idletimer>=idletimeout
Timer.Reset = true
Else
Page.AcsTime = Timer.value
```

Algorithm 3: Two Phase Page Ranking (TPPR)

I. Phase 1: Compute Similarity Score

Content Weight is based on how many terms in different web page fields match with the Query keywords. The content weight is calculated differently for different types of pages to give more weightage to page characteristics. Every page containing the query term is added to the list of the pages [lop]. Each page from the obtained list of pages has been examined for finding the location of the keyword. The keyword occurs in Meta tag gets more weight than the keyword occurs in title tag. The keyword occurs in title tag gets more weight than the keyword occurs in heading tag. The keyword occurs in heading tag gets more weight than the keyword occurs in paragraph tag. Finally, all the weights are summed up. The final score [SimRank] is the score given to the page based on the content.

```
Initialize SimRank=0
For $k = 0$ to $n$ do
If page Pgk contains Kwd
Insert Pgk in lop
End if
End for
For j = 0 to lop do
Find locality of Kwd for each page Pg jϵlop
Calculate SimRank = 0.2 * S3 + 0.3*S1 + 0.4*S2 + 0.1*S4
End for
```

Phase 2: Compute Usage Score

```
Initialize AcsTime=0 for each page Pgk
α – Minimum Threshold Value
ϒ- Maximum Threshold Value
For k=0 to lop do
AcsTime = AcsTime+ AcsTime (Pgk)
If AcsTime > ϒ
AcsTime=MaxTh
Else if AcsTime < α
AcsTime=0
End for
```

In phase 2, score is calculated based on the user access time of a web page. If the access time is greater than the threshold value, then it will be assigned to maximum value otherwise it will be assigned to 0.

Final page rank value can be calculated by using,

$$PRV = 0.6*SimRank + 0.4*AcsTime$$

In phase 1, Content based rank is computed and in phase 2, rank is computed based on the user access time. At last, both ranks are added together to get the total rank of a web page. Here 60% of content score and 40% of user access time are considered to get the final score.

CONCLUSION

By this we have to conclude that search engine using two phased algorithm such that the similarity score and the usage score are continuously calculated and rank the pages by the similarity and user usage score. These scores are combined to get most efficient and effective results. Therefore, we can get the data most relevantly and user supportively. The use of Internet in the recent years is growing rapidly which makes the need of a technique which can give accurate and relevant results to the user. Although there are several search engines currently present, it has been observed that they fail to capture user's preference and behavior and hence the search results may or may not be related with the context of the user. In this paper, hence we proposed a possible technique which can give users an experience of personalized web search and ultimately users can get what they want in a crisp manner in shorter time and fewer clicks as well. In future, the concept of query keyword suggestion can be added and with the feature of query formulation and query expansion, which helps the user at those times when users are not sure about the search query terms

REFERENCES

[1] C.S.Dadiyala, Pragati Patil and Girish Agrwal. A review of Query Log and Query Clustering. In Proceedings of the International Journal of IJERT, vol.1-issue 10, December 2012.

[2] Z. Dou, R. Song, and J.-R. Wen. A large-scale evaluation and analysis of personalized search strategies. In Proceedings of the 16th International Conference on World Wide Web (WWW'07), pages 581–590, Banff, Alberta, Canada, 2007.

[3] R. A. Baeza-Yates, C. A. Hurtado, and M. Mendoza. Improving search engines by query clustering. JASIST, 58(12):1793–1804, 2007.

[4] P. Ferragina and A. Gulli. A personalized search engine based on web-snippet hierarchical clustering. In Proceedings of the 14th International Conference on World Wide Web - Special interest tracks and posters (WWW'06), pages 801–810, Chiba, Japan, 2005.

[5] T. H. Haveliwala. Topic-sensitive pagerank: A context-sensitive ranking algorithm for web search. IEEE Trans. Knowl. Data Eng., 15(4):784–796, 2003.

[6] L. Li, Z. Yang, B. Wang, and M. Kitsuregawa. Dynamic adaptation strategies for long-term and short-term user profile to personalize search. In Proceedings of a Joint Conference of the 9th Asia-Pacific Web Conference and the 8th International Conference on Web-Age Information Management, pages 228–240, Huang Shan, China, 2007.

[7] Y. Li, Z. Bandar, and D. McLean. An approach for measuring semantic similarity between words using multiple information sources. IEEE Trans. Knowl. Data Eng., 15(4):871–882, 2003.

[8] F. Liu, C. T. Yu, and W. Meng. Personalized web search for improving retrieval effectiveness. IEEE Trans. Knowl. Data Eng., 16(1):28–40, 2004.

[9] Y. Lv, L. Sun, J. Zhang, J.-Y. Nie, W. Chen, and W. Zhang. An iterative implicit feedback approach to personalized search. In Proceedings of the 44th Annual Meeting of the Association for Computational Linguistics (ACL'06), pages 585–592, Sydney, Australia, 2006.

[10] S. Otsuka and M. Kitsuregawa. Clustering of search engine keywords using access logs. In Proceedings of 17th International Conference on Database and Expert Systems Applications (DEXA'06), pages 842–852, Krak´ow, Poland, 2006.