International Journal of Modern Engineering Research (IJMER)
www.ijmer.com Vol. 3, Issue. 4, Jul. - Aug. 2013 pp-2324-2326 ISSN: 2249-6645

Dynamic Organization of User Historical Queries

M. A. Arif[1], Syed Gulam Gouse[2]

[1]M. Tech, Nimra College of Engineering & Technology, Vijayawada, A.P., India.
[2]Professor, Dept. of CSE, Nimra College of Engineering & Technology, Vijayawada, A.P., India

ABSTRACT: *With the increasing number of published electronic materials, the World Wide Web (WWW) has become a vast resource for the individuals to acquire knowledge, solve problems, and complete tasks that use the Web information. To better support the users in their long term information quests on the Web, search engines keep track of their queries and clicks while searching online. In this paper, our goal is to automatically organize a user's search history into query groups, each containing one or more related queries and their corresponding clicks. Each query group corresponds to the atomic information. First and foremost, query grouping allows the search engine to better understand the user's session and potentially tailor that user's search experience according to her needs. Once query groups have been identified, the search engines can have a good representation of the search context behind the current query using queries and clicks in the corresponding query group. This will help to improve the quality of the key components of search engines such as result ranking, query suggestions, query alterations, sessionization, and collaborative search.*

KEYWORDS: *Log, Search engine, Query, WWW.*

I. INTRODUCTION

As the web is growing very rapidly, the user interacts very often and carries out many complex-task oriented operations over the net. The burst in the size and the richness of web is directly proportional to a variety and the complexity of task performed by user. Hence, the behavior of the user is unpredictable and untraceable as in a user may perform many different search terms over small period of time or may perform many similar searches at different times. Query log is generated by any user are hence no longer related to issuing simple navigational queries. Various studies on query logs[1][2] reveal that only about twenty percent of queries are navigational, while remaining are just transactional or navigational.

The main reason is now the user follows much elaborate task-oriented goals and operations such as planning a tour, planning a purchase and related decisions, managing their finances. The main way of accessing the information over the internet is through the keywords and queries using a search engine. A search engine has become a very important component of internet and they are broadly used for accessing the information over the net. However, a user decomposes the complex task oriented operation into number of smaller and simplified the queries, such as purchasing decision can be broken down into number of co-dependent steps over a period of time. Each step requires one or more queries, and each query results in one or more clicks on the relevant pages. During their complex search online, one of the important step towards providing the services and features that can help users is the capability to identify and group related queries together. This can be traced by using a new feature provided by any search engine which gives the user about their post navigational and task-oriented clicks and queries generally termed as "search histories".

In fact, identifying groups of related queries has applications beyond helping the users to make sense and keep track of queries and the clicks in their search history. Hence query grouping allows the search engine to better understand the user search behavior according to his/her need and his session. Once the query grouping is identified, the search engine can represent the result of the current queries and clicks by the user in the context. Query suggestions, result ranking, query alterations, sessionization, and collaborative search are the key components of the search engines, which may be improved via proper query grouping. Query grouping can also help different users by promoting the task level collaborative search. For example, a group of queries provided by some expert users, we can select the one which is highly relevant to the current user's activity and can suggest it to him. In this paper, we study the main concept of organizing users historical queries.

II. RELATED WORK

In [3][4], the authors investigate the search-task identification problem. More specifically, in [3], the authors considered a search session to consist of a number of tasks (missions), and each task further consists of a number of sub-tasks (goals). In [4], the authors employed similar features to construct a query flow graph, where two queries linked by an edge were likely to be part of the same search mission. In [5][6], the authors used the overlap of terms of two queries to detect changes in the topics of the searches. In [7], the authors studied the different refinement classes based on the keywords in queries, and attempted to predict these classes using a Bayesian classifier. In [8], the authors identified query sequences, called chains, by employing a classifier that combines a timeout threshold with textual similarity features of the queries, as well as the results returned by those queries. The problem of online query grouping is also related to query clustering in [9], [10]. In [9], the authors found query clusters to be used as possible questions for a FAQ feature in an Encarta reference Web site by relying on both text and click features. Graphs based on query and click logs [11] have also been used in previous work for different applications such as query expansion [12], query suggestions [4], ranking [13]. In several cases, variations of random walks have been applied on the graph in order to identify the most important nodes. In [13], a Markov random walk was applied on the click graph to improve ranking. In [14], a random walk was applied on the click-through graph to determine useful keywords.

III. QUERY RELEVANCE USING SEARCH LOGS

In this paper, we develop the machinery to define the query relevance based on Web search logs. Our measure of relevance is aimed at capturing two important properties of relevant queries, namely- (1) queries that frequently appear together as reformulations and (2) queries that have induced the users to click on similar sets of pages. We start our discussion by introducing three search behavior graphs that capture aforementioned properties. Following that, we show how we can use these graphs to compute the query relevance and how we can incorporate the clicks following a user's query in order to enhance our relevance metric. One way to identify the relevant queries is to consider query reformulations that are typically found within the query logs of a search engine. If two queries that are issued consecutively by many users occur frequently enough, they are likely to be used for reformulations of each other. To measure the relevance between any two queries issued by a user, the time-based metric, sometime, makes use of the interval between the timestamps of the queries within the user's search history. In contrast, our approach is defined by the statistical frequency with which any two queries appear next to each other in the entire query log, over all of the users of the system.

A different way to capture relevant queries from the search logs is to consider the queries that are likely to induce users to click frequently on the same set of URLs. For example, although the queries “ipod” and “apple store” do not share any text or appear temporally close in a user's search history, they are relevant because they are likely to have the resulted in clicks about the ipod product. In order to capture such property of relevant queries, we construct a graph called the query click graph (QCG). The query reformulation graph (QRG) and the query click graph (QCG) capture two important properties of relevant queries respectively. In order to make more effective use of the both properties, we combine the query reformulation information within QRG and the query click information within QCG into a single graph, QFG = (VQ, EQF), that we refer to as the query fusion graph. At a high level, EQF contains a set of edges that exist in either EQR or EQC. The weight of the edge (qi, qj) in QFG, wf (qi, qj), is taken to be a linear sum of the edge's weights, wr (qi, qj) in EQR and wc(qi, qj) in EQC,as follows:

$wf\ (qi,\ qj) = \_ \times wr(qi,\ qj) + (1 - \alpha) \times wc\ (qi,\ qj)$

The following Algorithm is used for calculating the query relevance by simulating random walks over the query fusion graph.

Relevance (q)

Input:

1) the query fusion graph, QFG
2) the jump vector, g
3) the damping factor, d
4) the total number of random walks, numRWs
5) the size of neighborhood, maxHops
6) the given query, q

Output:

the fusion relevance vector for q, relF q

```
( 1) Initialize relF q = 0
( 2) numWalks = 0; numVisits = 0
( 3) while numWalks < numRWs
( 4) numHops = 0; v = q
( 5) while v 6= NULL ^ numHops < maxHops
( 6) numHops++
( 7) relF q (v)++; numVisits++
( 8) v = SelectNextNodeToVisit (v)
( 9) numWalks++
( 10) For each v, normalize relF q (v) = relF , q (v)/numVisits
```

We use the jump vector “gq” to pick the random walk starting point. At each node v, for a given damping factor d, the random walk either continues by following one of the outgoing edges of v with a probability of d, or stops and then re-starts at one of the starting points in gq with a probability of (1−d). Then, each outgoing edge, (v, qi), is selected with the probability wf (v, qi), and the random walk always re-starts if v has no outgoing edge. The selection of the next node to visit based on the outgoing edges of the current node v in QFG and the damping factor d is performed by the SelectNextNodeToVisit process in the Step (7) of the algorithm. In addition to the query reformulations, user activities also include clicks on the URLs following each query submission.

IV. QUERY GROUPING USING THE QFG

In this section, we outline our proposed similarity function “simrel” to be used in the online query grouping process. For each query, we maintain a query image, which represents the relevance of the other queries to this query. For each query group, we maintain the context vector, which aggregates the images of its member queries to form an overall representation. We then propose a similarity function simrel for any two query groups based on these concepts of context vectors and query images. Note that our proposed of query reformulation graph, query images, and the context vectors are crucial ingredients, which lend significant novelty to the Markov chain process for determining relevance between queries and query groups.

For each query group, we maintain the context vector which is used to compute the similarity between the query group and the user's latest singleton query group. The context vector for a query group s, denoted "cxts", contains the relevance scores of each query in VQ to the query group s, and is obtained by aggregating the fusion relevance vectors of the queries and clicks in s. If s is a singleton query group containing only {qs1 , clks1}, it is defined as the fusion relevance vector rel(qs1,clks1). For a query groups = h{qs1 , clks1}, . . . , {qsk , clksk}i with k > 1, there are a number of different ways to define cxts. For instance, we can define it as the fusion relevance vector of the most recently added query and the clicks, rel(qsk ,clksk). Other possibilities include the average or the weighted sum of all the fusion relevance vectors of the queries and the clicks in the query group.

V. CONCLUSION

This paper shows how to organize user's historical queries in the search engine, means if the user search in the search engine, then that user query and URL will be stored in the history log. To reduce the burden on the users, in this paper we have created a feature called "Query Group". In this group we store the user queries. This Query Group having only related queries and the query groups will be created with different related queries. For example several users have been searching for the banks, then that time a new Query Group will be created about bank and another user have been searching for Hospitals information then the new Query Group will be created automatic and Dynamic fashion. This feature will be useful to the users for selecting or searching the related queries.

REFERENCES

[1]. J. Teevan, E. Adar, R. Jones, and M. A. S. Potts, "Information reretrieval: repeat queries in yahoo's logs," in SIGIR. New York, NY, USA: ACM, 2007, pp. 151–158.
[2]. Broder, "A taxonomy of web search," SIGIR Forum, vol. 36, no. 2, pp. 3–10, 2002.
[3]. R. Jones and K. L. Klinkner, "Beyond the session timeout: Automatic hierarchical segmentation of search topics in query logs," in CIKM, 2008.
[4]. P. Boldi, F. Bonchi, C. Castillo, D. Donato, A. Gionis, and S. Vigna, "The query-flow graph: Model and applications," in CIKM, 2008.
[5]. D. He, A. Goker, and D. J. Harper, "Combining evidence for automatic Web session identification," Information Processing and Management, vol. 38, no. 5, pp. 727–742, 2002.
[6]. H. C. Ozmutlu and F. C¸ avdur, "Application of automatic topic identification on Excite Web search engine data logs," Information Processing and Management, vol. 41, no. 5, pp. 1243–1262, 2005.
[7]. T. Lau and E. Horvitz, "Patterns of search: Analyzing and modelling Web query refinement," in UM, 1999.
[8]. F. Radlinski and T. Joachims, "Query chains: Learning to rank from implicit feedback," in KDD, 2005.
[9]. J.-R. Wen, J.-Y. Nie, and H.-J. Zhang, "Query clustering using user logs," ACM Transactions in Information Systems, vol. 20, no. 1, pp. 59–81, 2002.
[10]. J. Yi and F. Maghoul, "Query clustering using click-through graph," in WWW, 2009.
[11]. R. Baeza-Yates, "Graphs from search engine queries," Theory and Practice of Computer Science (SOFSEM), vol. 4362, pp. 1–8, 2007.
[12]. K. Collins-Thompson and J. Callan, "Query expansion using random walk models," in CIKM, 2005.
[13]. N. Craswell and M. Szummer, "Random walks on the click graph," in SIGIR, 2007.
[14]. Fuxman, P. Tsaparas, K. Achan, and R. Agrawal, "Using the wisdom of the crowds for keyword generation," in WWW, 2008.