Automatic User Histories Using Top-K Results

Podogu Anil Kumar [1], V.Srinivas [2]

[1]**Student, AL-AMEER College of Engineering & Information Technology,Anandapuram,Visakapatnam.**

[2]**Assistant Professor, AL-AMEER College of Engineering & Information Technology,Anandapuram,Visakapatnam**

Abstract: Most users want their search engine to incorporate three key features in query results. Relevant results (results they are actually interested in), Uncluttered (easy to read interface), Helpful options to broaden or tighten a search for accuracy. Automatically identifying query groups is helpful for a number of different search engine components and applications, such as query suggestions, result ranking, query alterations, sessionization, and collaborative search. In our approach, we go beyond approaches that rely on textual similarity or time thresholds, and we propose a more robust approach that leverages search query logs. We show through comprehensive experimental evaluation the effectiveness and the robustness of our proposed search log-based method, especially when combined with approaches using other signals such as text similarity. We propose to implement Random walk propagation methods that can construct user profiles based on the credentials obtained from their prior search history repositories. Combined with click points driven click graphs of user search behavior the IR system can support complex queries for future requests at reduced navigations. Random walk propagation over the query fusion graph methods support complex search quests in IR systems at reduced times. For developing an interactive IR system we also propose to use these search quests as auto complete features in similar query propagations. Biasing the ranking of search results can also be provided using ranking algorithms (top-k algorithms).Supporting these methods yields dynamic and improved performance in IR systems, by providing enriched user querying experience.

Index Terms: *query clustering, query reformulation; click graph, task identification, and Query Relevance and Search logs.*

I. INTRODUCTION

Now a day’s searching information from Web is the main aspect in Information Retrieving in the commercial way. Then most of users selecting various techniques were developed retrieving information. Users are no longer content with issuing simple navigational queries. Various studies on query logs reveal that only about 20% of queries are navigational. The rest are informational or transactional in nature. This is because users now pursue much broader informational and task-oriented goals such as arranging for future travel, managing their finances, or planning their purchase decisions. A recent Pew Internet and American Life report showed that Internet searches are a top Internet activity, second only to email in the

ISSN-2320-7884 (ONLINE)
ISSN-2321-0257 (PRINT)

report of the web search. The perceived need for information that leads to someone using an information retrieval system in the first place. The data transactional in nature in query logs is due to the user now to know much ample informational and the task oriented goals such as the arranging for future travel, managing their finance, or planning their purchase decisions. The search engine is still through the keyword queries to access information online. Over a period of time the complex tasks such as the travel arrangement has broken into the no. of codependent steps. Although some websites such as the redbus, ebay are helpful to provide from the single database we can attain the During the online complex quest to identify and to group the related queries together we have a standout step towards the enabling services and feature that are capable. Currently we are using the "Search History" in major search engines where users can allow tracking their online searches by recording their queries and clicks. The figure1 can illustrate the portion of user's history as it shown by the yahoo search engine. In addition to viewing their search history, users can manipulate it by manually editing and organizing related queries and clicks into groups, or by sharing them with their friends. As the history get long over time it will be untenable to the manual efforts are obstreperous, where the above features may helpful. Identifying groups of related queries has applications beyond helping the users to make sense and keep track of queries and clicks in their search history.

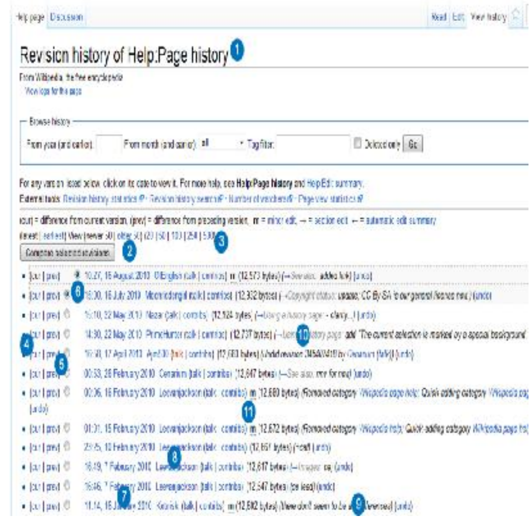

Figure 1: User Histories with page ranking.

The key components of search engines such as the result ranking, sessionization, query suggestions, collaborative search and query alterations. Consider a search engine knows that a current query "financial statement" belongs to the {"Bank of India", "financial statement"} group query. The rank of the page can be boosted, that provides information about how to get a Bank of India statement instead of the Wikipedia article on "financial statement". Query grouping can also assist other users by promoting task-level collaborative search. Explicit collaborative search can also be performed by allowing users in a trusted community to merge, find and share relevant query groups to perform larger, long-term tasks on the Web.

II. RELATED WORK

Our work differs from these prior works in the following aspects. First, the query-log based features in are extracted from co-occurrence statistics of query pairs. In our work, we additionally consider query pairs having common clicked URLs and we exploit both concurrence and click information through a combined query fusion graph [4] will not be able to break ties when an incoming query is considered relevant to two existing query groups. Additionally, our approach does not involve learning and thus does not require manual labeling and re-training as more search data come in; our Markov random walk approach essentially requires maintaining an updated query fusion graph. Finally, our goal is to provide users with useful query groups on the fly while respecting existing query groups. On the other hand, search task identification is ostly done at server side with goals such as ersonalization, query suggestions [5] etc.

We study the problem of organizing a user's historical queries into groups in a dynamic and automated fashion. Automatically identifying query groups is helpful for a number of different search engine components and applications, such as query suggestions, result ranking, query alterations, sessionization, and collaborative search. In our approach, we go beyond approaches that rely on textual similarity or time thresholds, and we propose a more robust approach that leverages search query logs. We experimentally study the performance of different techniques, and showcase their potential, especially when combined together.

III. EXSTING SYSTEM

The search engine displays results based page ranking algorithms. Users are no longer content with issuing simple navigational queries. The primary means of accessing information online is still through keyword queries to a search engine. Keyword based search engines cannot address this kind of complicated tasks. A complex task such as travel arrangement has to be broken down into a number of co-dependent steps over a period of time. For instance, a user may first search on possible destinations, timeline, events, etc. After deciding when and where to go, the user may then search for the most suitable arrangements for air tickets, rental cars, lodging, meals, etc. Each step requires one or more queries, and each query results in one or more clicks on relevant pages. Search Engine tries to construct user profile based on his ipaddress/login credentials from its user search history repositories. If the user already exists, the search engine checks from its user search history repositories up to a certain threshold whether the user already queried the same query previously. If the user did, then search engine further retrieves click points from user search history repositories and reformulates query results by generating click graphs. Click graphs contain useful information on user behavior when searching online. This step is called query fusion graph.

IV. PROPOSED APPROACH

Random walk propagation over the query fusion graph methods support complex search quests in IR systems. For making the IR Systems effective and dynamic we propose to use these search quests as auto complete features in similar query propagations. Biasing the ranking of search results can also be provided using any ranking algorithms (top-k algorithms). Supporting these methods yields dynamic performance in IR systems, by providing enriched user querying experience.

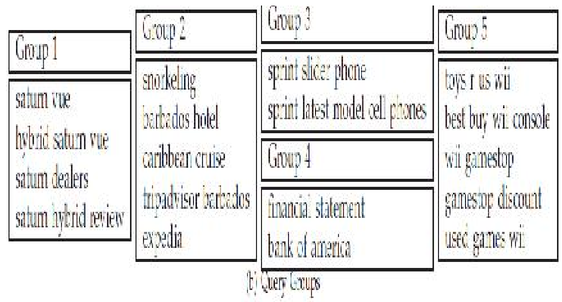

Figure 2: User groups retrieving from user hostories

User nformation was collected from various user techniques for retrieving relevent information.

V. PERFORMANCE ANALYSIS

In this section we will describe the efficient results of the every user present in the Google search web application

Module Description:

1. Query Group
2. Search history
3. Query Relevance and Search logs
4. **Dynamic Query Grouping**

Query Group:

We need a relevance measure that is robust enough to identify similar query groups beyond the approaches that simply rely on the textual content of queries or time interval between them. Our approach makes use of search logs in order to determine the relevance between query groups more effectively. In fact, the search history of a large number of users contains signals regarding query relevance, such as which queries tend to be issued closely together (query reformulations), and which queries tend to lead to clicks on similar URLs (query clicks). Such signals are user-generated and are likely to be more robust, especially when considered at scale. We suggest measuring the relevance between query groups by exploiting the query logs and the click logs simultaneously.

Search History:

We study the problem of organizing a user's search history into a set of *query groups* in an automated and dynamic fashion. Each query group is a collection of queries by the same user that are relevant to each other around a common informational need. These query groups are dynamically updated as the user issues new queries, and new query groups may be created over time.

Rank Search:

Our proposed approach mainly focused on the efficient user histories based on ranking procedure of the page rank.

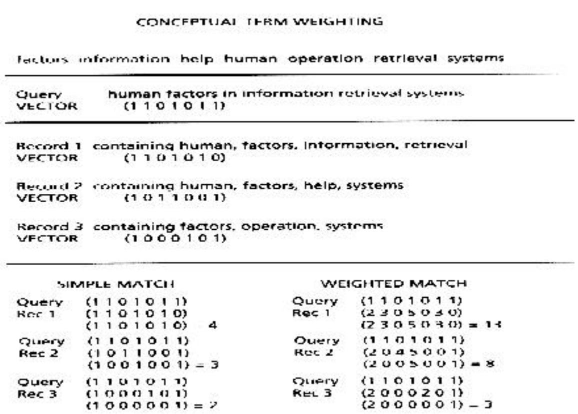

Figure 3: Raking procedure

Query Relevance and Search logs:

We now 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 the aforementioned properties. Following that, we show how we can use these graphs to compute query relevance and how we can incorporate the clicks following a user's query in order to enhance our relevance metric.

Dynamic Query Grouping:

One approach to the identification of query groups is to first treat every query in a user's history as a singleton query group, and then merge these singleton query groups in an iterative fashion (in a k-means or agglomerative way. However, this is impractical in our scenario for two reasons. First, existing query groups, potentially doing the user's own manual efforts in organizing her history. Second, it involves a high computational cost, since we would have to repeat a large number of query group similarity computations for every new query.

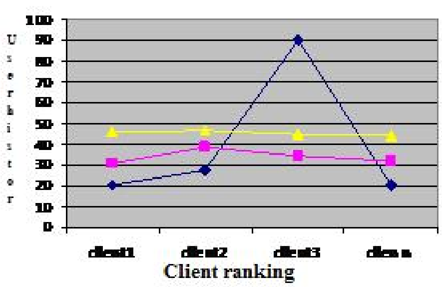

Figure 4: Performance of the user results

Above diagram shows the efficient results for user's data in number of users increasing in the order for retrieving information.

VI. CONCLUSION

Search Engine tries to construct user profile based on his ipaddress/login credentials from its user search history repositories. If the user already exists, the search engine checks from its user search history repositories up to a certain threshold whether the user already queried the same query previously. If the user did, then search engine further retrieves click points from user search history repositories and reformulates query results by generating click graphs. Biasing the ranking of search results can also be provided using any ranking algorithms (top-k algorithms). Supporting these methods yields dynamic performance in IR systems, by providing enriched user querying experience.

VII. REFERENCES

1)Heasoo Hwang, Hady W. Lauw, Lise Getoor and Alexandros Ntoulas"Organizing User Histories" IEEE 2012 Transactions on Knowledge and Data Engineering, Volume: 24 , Issue: 5,2012.

2) Mark J. Carman and Fabio Crestani, Morgan A. Harvey and Mark Baillie" Towards Query Log based Personalization using Topic Models" IEEE transactions volume:18,Issue:5,2010.

3) 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.

4) A. Broder, "A taxonomy of web search," *SIGIR Forum*, vol. 36, no. 2, pp. 3–10, 2002.

5)A. Spink, M. Park, B. J. Jansen, and J. Pedersen, "Multitasking during Web search sessions," *Information Processing and Manage- ment*, vol. 42, no. 1, pp. 264–275, 2006.