User Search History Using Query Clustering for Information Retrieval

Smruti Patel, Dharmesh Bhalodia
Computer Engineering, Silver Oak College of Engineering & Technology, Ahmedabad, Gujarat, India

Abstract - As the size and richness of information on the Web grows, so does the variety and the complexity of tasks that users try to accomplish online. The existing search engines organize such queries only in chronological order. However, when the queries are grouped together based on the relevancy that might be very useful to users as they can reuse queries with ease. In this paper, we study the problem of organizing a user’s historical queries into groups in a dynamic and automated fashion. This organization of user search histories can have various real-time utilities such as result ranking, query alternations, query suggestions, sessionization, and collaborative search. We experimentally study the performance of different techniques, and showcase their potential, especially when combined together.

Index Terms - search history, query grouping, query reformulation, click graph

I. INTRODUCTION

Clustering is a useful technique for the discovery of data distribution and patterns in the underlying data. The goal of clustering is to discover both the dense and the sparse regions in a data set. Clustering is the process of organizing data into meaningful groups, and these groups are called clusters. [3] Query Clustering itself means Query grouping. Query grouping allows the search engine to better understand a user’s session and potentially tailor that user’s search experience according to her need.

In this paper we study the problem of organizing a user’s search history into a set of query groups in an automatically and dynamic fashion. Each group is a collection of a query by the same user that is relevant to each other around a common information need. This query groups are dynamically update as the user issues new query, and query groups may be created over time.

For example, if a search engine knows that a current query “financial statement” belongs to a (“bank of America”, “financial statement”) query group, it can boost the rank of the page that provides information about how to get a Bank of America statement instead of the Wikipedia article on “financial statement”, or the pages related to financial statements from other banks.

II. PROBLEM DEFINITION

- Query Reformulation Graph
- Query Click Graph
- Query Fusion Graph
- Semantic similarity

Query Reformulation Graph
One way to identify 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 reformulations of each other. To measure the relevance between two queries issued by a user, the time-based metric, simtime, 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 two queries appear next to each other in the entire query log, over all of the users of the system.

Query Click Graph
A different way to capture relevant queries from the search logs is to consider 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 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.
Query Fusion Graph
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 both properties, we combine the query reformulation information within QRG and the query click information within QCG into a single graph, \( QFG \equiv (V_Q, EQ_F) \), that we refer to as the query fusion graph.

Semantic similarity
It is useful to organize search histories in some meaningful way other than chronological. We need to depend on word net API for result. Semantic similarity is a metric defined over a set of documents or terms, where the idea of distance between them is based on the likeness of their meaning or semantic content as opposed to similarity which can be estimated regarding their syntactical representation.

III. PROPOSED MODEL
Organizing the query groups within a user’s history is challenging for a number of reasons. First, related queries may not appear close to one another, as a search task may span days or even weeks. This is further complicated by the interleaving of queries and clicks from different search tasks due to users’ multitasking, opening multiple browser tabs, and frequently changing search topics.

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. [2] Our query grouping algorithm relies heavily on the use of search logs in two ways: first, to construct the query fusion graph used in computing query relevance, and, second, to expand the set of queries considered when computing query relevance.

- **Group creation**

  \( S = \{s_1, s_2, s_3, ..., s_n\} \)  \( (S = \) set existing query group)  
  \( S_1 = (\{q_1, clk_1\}, \{q_2, clk_2\}, ..., \{q_k, clk_k\}) \)  
  \( S_2 = (\{q_1, clk_1\}, \{q_2, clk_2\}, ..., \{q_k, clk_k\}) \)  

  Example: group creation
  \( S_1 = \text{Finance} \)  
  \( \{q_1, clk_1\} \rightarrow \text{Bank of America} \)  
  \( \{q_2, clk_2\} \rightarrow \text{HDFC Bank} \)  
  \( \{q_3, clk_3\} \rightarrow \text{Indian finance system} \)  

  \( S_1 = \text{Travel} \)  
  \( \{q_1, clk_1\} \rightarrow \text{South Africa Safari} \)  
  \( \{q_2, clk_2\} \rightarrow \text{Shimla manali} \)  
  \( \{q_3, clk_3\} \rightarrow \text{Mauritius} \)  

  \( S_c = \{\text{qc}, \text{clkc}\} \rightarrow \text{Tata Consultancy Services} \)

  Where \( S_c \) is current query group

In Dynamic query grouping we first check \( S_c \) contains in set of existing query groups \( S \) or not. If it contains than add that click and query to the existing group, otherwise we need to create new group called ‘IT Corporate’ in \( S \).

IV. RESULTS
The implementation of the proposed Algorithm of “User Search History Using Query Clustering for Effective Information Retrieval Using Query Reformulation Graph, Query Click Graph, Query Fusion Graph, and Semantic Similarity Methodology” is implemented in Asp.net.
Here, in the implementation implemented using four Methodologies:

1) Query Reformulation Graph
2) Query Click Graph
3) Query Fusion Graph
4) Semantic Similarity

Run This Methods I use Microsoft visual studio, Google API and Browser.

Figure: Search and grouping

Figure: Search Query and result
Figure: click on display Result

Figure: clicking on another result

Figure: past search data
Figure: Suggestion in textbox

Figure: People also searching for

Figure: Manage history
V. CONCLUSION AND FUTURE WORK

The query reformulation and click graphs contain useful information on user behaviour when searching online. We show how such information can be used effectively for the task of organizing user search histories into query groups. The future work intend to investigate the usefulness of the knowledge gained from these query groups in various applications such as providing ranking of search results.

REFERENCE