Enhancing Web Search by Mining Task Trails in Web Logs

Josseena Jose, Hafsath C.A

Abstract*— Web search logs record the users search queries and related actions in search engines. By mining these information it is possible to understand user search behaviors. A task can be defined as atomic user information need, whereas a task trail represents all user activities within that particular task, such as query reformulations, URL clicks. In most of the previous works, web search logs have been studied mainly at session, query or task level where users may submit several queries within one task and handle several tasks within one session. Instead of analyzing task within a session, cross session task can be analysed to determine the user search behaviour much more efficiently.

***Index Terms*— Search log mining, task trail, cross-session search task**

I. INTRODUCTION

Search engine is a web software program available over the Internet that searches documents and files for keywords and returns the list of results containing those keywords. Nowadays, search engines have become the most important and indispensable Web portal, where people perform a wide range of searches in order to satisfy various information needs. Web logs records these search queries and related actions of a user on internet.

Web logs contain a set of users, and each user has a sequence of consecutive behaviors e_1, e_2, … e_n, where each behaviour e_i can be a search behavior or a browse behavior. A search behavior is a single query submitted to the search engine. A browse behavior belongs to one of the following activities: 1) user starts to surf from the homepage of the browser; 2) user types a URL address in the browser; 3) user pastes the URL address from other place into browser; 4) user clicks a bookmark or favourite page in the browser; 5) user clicks the “back” or “forward” button in the browser; 6) user clicks an anchor link or a search result.

Web log query clustering is a technique for discovering similar queries on a search engine. The web log query clustering techniques can be mainly of Query level, Session level and Task-level. The Query level clustering analyses each query in the web log separately, ie treat one query plus its followers as an independent query trail. The session-level query clustering technique groups a set of queries issued by the user of a web search engine within a particular time period. Task level query clustering groups a set of non-consecutive queries issued by a user to carry out a particular task. After clustering the queries into sessions or tasks, the web log can be analysed and required knowledge can be extracted. The need of web log analysis is to determine the user search behaviour and it can be used in various applications such as webpage re-ranking, page utility estimation, website recommendation, predicting user search interest and suggesting related queries on the internet. Since many tasks have been shown to span multiple search sessions, the empirically-set timeout threshold may not be a valid criterion for identifying the semantic structure among queries.

In this paper, we have presented a method to accurately extract cross-session search tasks from users historic search activities. Extracting cross session search task eliminates the error that can be caused during extracting in-session search tasks by considering all the available query pairs.

The rest of the paper is organized as follows. Section 2 gives an overview of the most relevant related works. Section 3 defines the problem and section 4 describes the proposed system. Section 5 describes an application using extracted search tasks. The experiments and results are given in Section 6 followed by conclusion in Section 7.

II. RELATED WORKS

Researches are always been conducted to understand users search behaviour. This section presents some of the previous approaches for segmentation of web logs.

In [1] R. White and J. Huang found that following the query trails, users can find more useful information. The trail starts with a search engine query and comprises a set of pages visited until the trail terminates with a new query. The problem is that it fails in revealing task-based sessions due to the multi-tasking user’s behavior. Liu et al. [2] modeled the user browsing behavior as a continuous Markov model and propose a BrowseRank algorithm to calculate page importance scores with the model. It uses a time threshold as a delimiter to segment query sessions.

In [3] Beeferman and Berger et.al, proposed an agglomerative clustering approach for the segmentation of web logs. It constructs a query-page bipartite graph with one set corresponding to queries submitted by the user, and the other set corresponding to clicked pages. During the clustering process, the algorithm combines two most similar queries into one query node, then the two most similar pages into one page node iteratively.

Manuscript published on 30 November 2015.
***Correspondence Author(s)**

Josseena Jose, PG Scholar, Department of Computer Science and Engineering, Ilahia College of Engineering and Technology, MVPA, (Kerala). India.

Hafsath C.A, Asst. Professor, Department of Computer Science and Engineering, Ilahia College of Engineering and Technology, MVPA, (Kerala). India.

In [4] Huang presents an effective log-based approach to relevant term extraction and term suggestion. It uses a time threshold as a delimiter to segment query sessions. Then the relevance between each pair of query terms in a query log is computed. The drawback is that a session contains multiple atomic information needs which is not considered here. In [5] R. Jones and K. L. Klinkner propose and evaluates a method for the automated segmentation of users query streams into hierarchical units. In order to address interleaved goals, consider all possible pairs of queries, and consider whether the pair of queries comes from the same task. Including the interleaving in the model allows us to more accurately measure the length of time or number of queries a user needs to complete tasks.

In [6] Lucchese et.al proposed Query Clustering using Weighted Connected Component (QC-WCC) method. Upon the query similarity function, an undirected graph is built for queries within a session. The vertices of the graph represent queries and the edges represent similarity scores between queries. After removing the suspicious edges with scores below a threshold, any connected component of the remaining graph is identified as a task.The main drawback of this approach is that it has high time complexity. To address this issue, Lucchese et al. proposed head-tail component query clustering (QC-HTC) approach. It utilizes the heuristic that queries are submitted sequentially by users. The main drawback of this approach is that computing similarity between head and tail parts of the query sequence violates the task interleaving observation in search logs. In [7] Zhen Liao and Yang Song et.al proposed spread Query Task Clustering algorithm (QC-SP). For this approach, if q_1 is similar to q_2 and q_2 is similar to q_3, then there is no need to calculate the relevance between q_1 and q_3. But the problem is that if all the tasks are short and interleaved each other, QC-SP has same time complexity as QC-WCC. In [8] Zhen Liao and Yang Song et.al proposed a Bounded Spread Query Task Clustering approach, named QC-BSP. By setting a length bound bl, the time complexity of QC-BSP is further reduced. Here in this paper we present a method to extract cross session search task. Search tasks frequently span multiple sessions, and thus developing methods to extract these tasks from historic data is central to understanding search behaviors. Extracting cross session search task reduces the error rate.

III. PROBLEM DEFINITION

A search log can be regarded as a sequence of query and click events. It contains a set of users, and each user has a sequence of consecutive behaviors e_1, e_2, …e_n where each behaviour e_i can be a search behavior or a browse behavior. A search behavior is a single query submitted to the search engine. A browse behavior belongs to one of the following activities: 1) user starts to surf from the homepage of browser; 2) user types a URL address in the browser; 3) user pastes a URL address into the browser; 4) user clicks a bookmark or favourite page in the browser; 5) user clicks the “back” or “forward” button in the browser; 6) user clicks an anchor link or a search result. A query trail q is defined as a sequence of user behaviors e_1^q , e_2^q,… e_m^q of a particular user u, starting from a single query, followed by a sequence of browsing behaviors before the next query is submitted by the same user. The main problem with this approach is that the semantic associations between adjacent query trails are lost since the next query submitted by the same user for the same purpose is taken as a different query trail. A session trail s is defined as a sequence of user behaviors e_1^s , e_2^s,… e_k^s of a particular user u, where any two consecutive behaviors e_i , e_{i+1} occurred within time threshold θ. The disadvantage is that a session contains multiple atomic information needs which are semantically irrelevant to each other. A task trail t is defined as a sequence of user behaviors e_1^t ,e_2^t,… e_m^t of a particular user u, occurred within a session, where all user behaviors collectively define an atomic user information need. Since many tasks have been shown to span multiple search sessions, the timeout threshold may not be a valid criterion for identifying the semantic structure among queries. The user activities within one task may not be necessarily consecutive in web logs because multiple tasks can be interleaved with each other. Also the in-session clustering approach does not consider the query pairs that lie near the boundaries of the session.

In order to overcome all these problems we propose cross-session search task method to extract user search behavior. A cross-session search task c is defined as a sequence of user behaviors e_1^c , e_2^c,… e_m^c of a particular user u, where all user behaviors collectively define an atomic user information need.

IV. PROPOSED SYSTEM

A task can be defined as a set of semantically relevant query trails to satisfy a particular information need. Two queries can be grouped into same task if they satisfy any of the following conditions: (1) they are identical; (2) one is a part of the other (e.g., “amazon” and “amazon shopping”); (3) two partially agree to each other (e.g., “gate result” and “gate score”); (4) one is a type of the other (e.g., “machnie learning” and “machine learning”). These rules can be used in the dataset construction process and propose an efficient clustering framework to group similar queries into same tasks.

TASK EXTRACTION

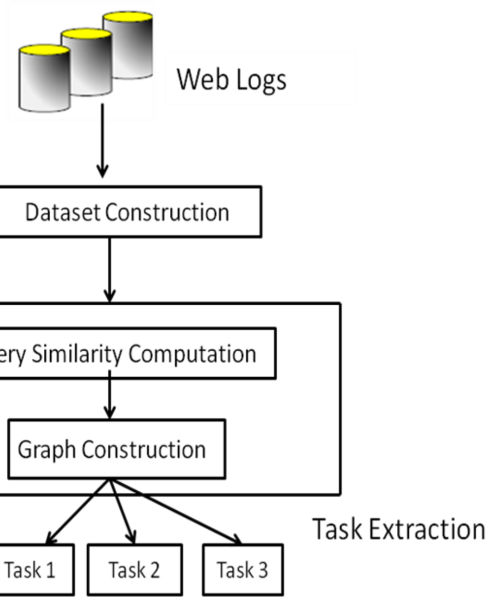

Fig. 1. Architecture of proposed system

First, we generate all the available combinations of the query pairs and then construct a labeled dataset for task classification by categorizing the query pairs into same and different task. Then compute the similarity between any two queries. A linear SVM classifier can be used to learn the weights of various features on labeled data. Last, queries similar to each other are clustered into the same task.

A. Query Similarity

A linear SVM can be used to compute the similarity between two queries. First, a data set should be constructed to learn a good query similarity function on labeled data for task classification. The labels include same task and different task. Here we use total 13 features to measure the similarity between queries. These features can be categorized into two groups: such as time based (temporal) and query word based features. The details of these features are mentioned in the following table.

TABLE 1. Features of Query Pair

Feature Description	Weight
Temporal Features	
timediff_1: Time difference in seconds	-0.1121
timediff_2: Category for 1/5/10/30 mins	-0.0623
Word Features	
lv_1: Levenshtein distance of two queries	0.0106
lv_2: lv_1 after removing stop-words	-0.1951
prec_1: average rate of common terms	-0.2870
prec_2 : prec_1 after removing stop-words	1.2058
prec_3: prec_1 If term A contains B A=B	0.5292
rate_s: rate of common characters from left	1.6318
rate_e: rate of common characters from right	0.4014
rate_l: rate of longest common substring	0.4941
b_1: 1 If one query contains another, else 0	0.6361
q _cosine: cosine similarity between two queries	5.30
q _ jac: Jaccard coeff between two queries	1.51

The column weight in the table represents the weight of each feature for the similarity function. Here some frequent searched but meaningless words are selected as the stop words.

B. Clustering Queries into Tasks

Here we present a method to accurately extract cross-session search tasks from users historic search activities. Search tasks frequently span multiple sessions, and thus develop methods to extract these tasks from historic data to understand search behaviors. Extracting cross session search task reduces the error rate. The traditional search task extraction method provides a flat clustering structure, but this method provides a hierarchical structure. Comparing to the flat clustering, the hierarchical structure provides more in-depth details to understand users search behaviors and their information needs. In the cross-session search task extraction problem, we treat a user's entire query log as a whole and explicitly model the dependency among queries and cluster queries into same task or different task.

Algorithm

```
Input: Query set Q= {$q_1$,$q_2$…$q_N$},cut-off threshold b;
Output: A set of tasks S;
Initialization: S = ∅; cid: content task id
Query to task table M=∅;
// Initialize queries that are same into one task
cid=0;
for i = 1 to N do
if M[$_{Qi}$] exists then
add Qi into S(M[$Q_i$]);
else
M[$Q_i$]=cid++;
if |S| = 1 return S;
for i = 1 to N do
// if two queries are not in the same task
if L[$Q_i$ ] ≠ L[$Q_{i+len}$] then
T ← sim (L[$Q_i$], L[$Q_{i+N}$]);
if T ≥ b then
merge S($Q_i$) and S($Q_{i+N}$);
modify L;
// break if there is only one task
if |S| = 1 break;
return S;
```

The above algorithm finds the similarity between two queries. If two queries belongs to same task, they are inserted it into the task table. If similarity measure of tasks is greater than the threshold value, it is added to queries of same tasks otherwise ignore the query. This algorithm accurately clusters the queries into same and different tasks.

V. APPLICATION: QUERY SUGGESTION

The related queries mined from cross-sessions can be used for query suggestion. Search tasks frequently span multiple sessions, and thus extracting cross-session search tasks from historic data helps to understand user search behaviors and also to accurately suggest related queries.

In web search, there are many techniques used extensively for effective retrieval of information from the server. When a user issued a query into the search engine, it may return large number of web page results and therefore, it becomes extremely important to rank these results in such a manner that it returns accurate and more relevant web results. These tasks of prioritizing the search results are performed by various ranking algorithms. But the most important thing is that the search interest of every user differs with every other user uniquely. In most of the previous approaches, the search engine that return results for a given user query is not uniquely based on their earlier searching behavior. This paper personalizes user search activities in order to find search interest of each user. Here uses simple authentication of username and password to maintain each user's searched profiles in the web log. The personalization of user search behavior is done by storing user activities and analyzing web log by clustering frequently accessed tasks by semantic task clustering algorithm and ranks web results based on time spent within a particular site. Thus it uniquely ranks web results for each user.

VI. EXPERIMENTAL EVALUATION

For the experimental study, web log data sets of various users are extracted. It contains the searching activities of users in search engines. The web log is segmented at cross-session task level, so that a user's entire query log as a whole is taken as input and explicitly models the dependency among queries and cluster queries into same task or different task. This method accurately identifies the user search behaviour. Implementations were done in Java. From the experimental results it can be seen that, the in-session clustering approach does not consider the query pairs that lie near the boundaries of the session. This fails in revealing same task accurately. But in the case of cross-session clustering approach, it considers all the query pairs and understands user search behavior accurately.

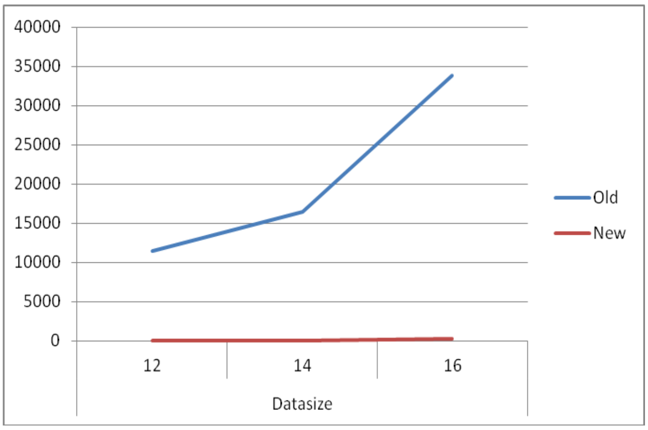

Fig. 2. Data size plotted against time

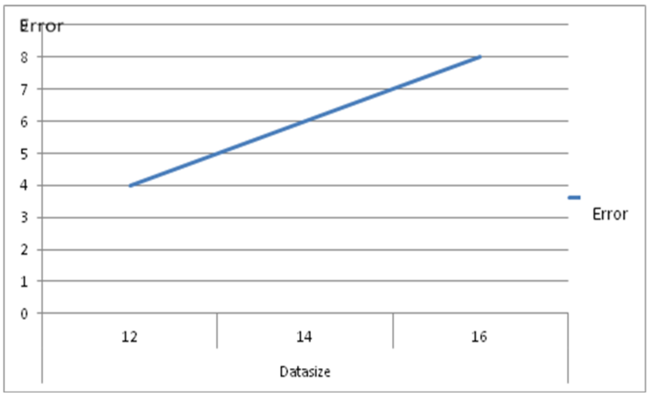

Fig. 3. Data size plotted against error rate

VII. CONCLUSION

In this paper, we present a method to accurately extract cross-session search tasks from users historic search activities. Search tasks frequently span multiple sessions, and thus developing methods to extract these tasks from historic data is central to understanding search behaviors. Extracting cross session search task reduces the error rate. Comparing to the flat clustering, the hierarchical structure provides more in-depth details to understand users' search behaviors and their information needs. The related queries mined from cross-sessions can be used for query suggestion. As a future work, combine task segmentation and query suggestion with prediction of user satisfaction, which opens up the possibility of truly understanding whether web search engines are satisfying their users.

ACKNOWLEDGEMENT

The authors wish thanks to the Management and Principal and Head of the Department (CSE) of Ilahia College of Engineering and Technology for their support and help in completing this work.

REFERENCES

1. R. White and J. Huang, "Assessing the scenic route: measuring the value of search trails in web logs," in Proc. 33rd Int. ACM SIGIR Conf. Res. Develop. Inform. Retrieval, 2010, pp. 587–594.
2. Y. Liu, B. Gao, T.-Y. Liu, Y. Zhang, Z. Ma, S. He, and H. Li, "Browserank: letting web users vote for page importance," in Proc. 31st Annu. Int. ACM SIGIR Conf. Res. Develop. Inform. Retrieval, 2008, pp. 451–458.
3. D. Beeferman and A. Berger, "Agglomerative clustering of a search engine query log," in Proc. 6th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2000, pp. 407–416.
4. J. C. K. Huang, L. F. Chien, and Y. J. Oyang, "Relevant term suggestion in interactive web search based on contextual information in query session logs," J. Amer. Soc. Inform. Sci. Technol., vol. 54, pp. 638–649, 2003.
5. R. Jones and K. L. Klinkner, "Beyond the session timeout: Automatic hierarchical segmentation of search topics in query logs," in Proc. 17th ACM Conf. Inform. Knowl. Manage., 2008, pp. 699–708.
6. C. Lucchese, S. Orlando, R. Perego, F. Silvestri, and G. Tolomei, "Identifying task-based sessions in search engine query logs," in Proc. 4thACMInt. Conf. Web Search Data Mining, 2011, pp. 277–286.
7. Z. Liao, Y. Song, L. -w. He, and Y. Huang, "Evaluating the effectiveness of search task trails," in Proc. 21st Int. Conf. World Wide Web, 2012, pp. 489–498.
8. Zhen Liao, Yang Song, Yalou Huang, Li-wei He, and Qi He" Task Trail: An effective segmentation of user search behavior" IEEE transactions on knowledge and data engineering, vol. 26, no. 12, December 2014