ISSN No. 0976-5697

Volume 7, No. 2, March-April 2016

International Journal of Advanced Research in Computer Science

REVIEW ARTICLE

Available Online at www.ijarcs.info

An Effective Segmentation of User Search Interests Based On Task Track

Arati A. Topai
Department of Computer Science,
Ashokrao Mane Group of Institutions
Vathar-416112, India.

A. B. Rajmane
Department of Computer Science,
Ashokrao Mane Group of Institutions
Vathar-416112, India.

Abstract:Web logs record the users search queries and related actions in search engines. It is possible to understand user search behaviors by mining these information. A task can be defined as atomic user information need, whereas a task track represents activities of all user within that particular task, such as query reformulations, URL clicks. In the previous works, web logs have been studied at session, query or task level where users have to submit several queries within one task and handle several tasks within a session. Although previous studies have addressed the problem i.e. identification of task, little is known about the advantage of using task over session or query for search applications. It is defined to conduct immense analyses and comparisons to evaluate the efficacy of task track in search applications: user satisfaction determination, user search interest's prediction and related query suggestions.

Keywords- log analysis, Search log mining, Task Track

I. INTRODUCTION

Nowadays, most users leverage search engines as an important tool to accomplish various information seeking tasks, e.g., to find particular Web pages, locating target resources, or accessing information of certain topics. Searching activities can be recorded by web search logs. These search logs can be used for user satisfaction analysis [1], page utility estimation [2], user search interest prediction [3], query suggestion [4], webpage re-ranking [5], website recommendation [6], etc. Most of the work analyzed this log on query and session level. The query level analysis is the finest grained, but treats one query and its followers as a separate query. In session-level analysis groups a set of queries within particular time issued by the user of web search engine. However, the tasks have not been explicitly identified. In previous work presented an approach that is significant for studying Web users search contexts. The approach automatically groups consecutive search activities of a user's on the same search topic into one session. It uses Dempster–Shafer theory to combine evidence obtained from two sources, each of which is based on the statistical data from Web search logs. Detecting query reformulations by a Web searcher during a search episode is an important area of research for designing helpful searching systems, recommender systems, personalization and targeting content to particular users. One more important thing for analysis of web search log is that task identification. It requires systematically analyze the utilities of task-level search log analysis and compare it with session-level and query-level search log analyses in real applications.

In task-level search log analysis, named as task track to understand user search behavior. A Task is defined as atomic user information needs, whereas a task track represents activities of all users within that particular task, such as query reformulations, URL clicks. It is focused on comparison of task, session and query trails in the search applications like user satisfaction determination, user search interest's prediction and related query suggestions.

II. RELATED WORK

Researches are conducted to understand behavior of users search. This section represents some of the previous work for segmentation of web logs.

S. Fox, K. Karnawat, M. Mydland, S. Dumais [1] outlined the relationship between implicit and explicit measures of user satisfaction which is focused on web search applications. They used Bayesian modeling techniques and found that a combination of the right implicit measures can provide good predictions of explicit judgments of user satisfaction. They also explored the use of usage patterns for characterizing sequences of user behavior patterns and predicting user satisfaction.

R. White and J. Huang [2] outlined the log-based methodology estimating the value to users of traversing multi-page search trails. The evaluation showed that full-trails and sub-trails provide users with significantly more topic coverage, topic diversity, and novelty than trail origins, and slightly more useful but slightly less relevant information than the origins.

R. White, P. Bennett, and S. Dumais [3] outlined the effectiveness of activity-based context in predicting users search interests. They demonstrated that context can be captured and modeled for a significant portion of search queries and explored the value of modeling the current query, its context, and their combination, and different sources of context. They showed that intent models developed from many sources perform best overall.

H. Cao, D. Jiang, J. Pei, Q. He, Z. Liao, E. Chen, and H. Li [4] proposed a novel approach to query suggestion using click-through and session data. This approach considers not only the current query but also the recent queries in the same session to provide more meaningful suggestions. Moreover, they grouped similar queries into concepts and provide suggestions based on the concepts.

B. Xiang, D. Jiang, J. Pei, X. Sun, E. Chen, and H. Li [5] outlined the problem of using context information in ranking documents in Web search and conducted an empirical study on real search logs and developed different ranking principles for different types of contexts. They further adopted a learning-to-rank approach and incorporated their principles to ranking models. The experimental results verified that context-aware ranking approach improves the ranking of a commercial search engine which ignores context information.

White, R., Bilenko, M. and Cucerzan, S. [6] outlined a novel approach for enhancing users' Web search interaction by providing links to websites frequently visited by past searchers with similar information needs. A user study conducted was evaluated the effectiveness of the proposed technique compared with a query refinement system and unaided Web search. Results show that search enhanced by destination suggestions outperforms other systems for exploratory tasks, with best performance obtained from mining past user behavior at query-level granularity.

D. G. He, A. Geoker, and D. J. Harper [7] outlined some information about web session identification based on two sources of evidence time interval and search pattern obtained from analyzing a large batch of Web search logs. The approach automatically groups a users consecutive search activities on the same search topic into one session. It uses Dempster–Shafer theory to combine evidence extracted from two sources, each of which is based on the statistical data from Web search logs.

C. Lucchese, S. Orlando, R. Perego, F. Silvestri, and G. Tolomei [8] outlined identifying task-based sessions in search engine query logs. They explain several variants of well-known clustering algorithms, as well as a novel efficient heuristic algorithm, specifically tuned for solving the Task-based Session Discovery Problem (TSDP). These algorithms also exploit the collaborative knowledge collected by Wiktionary and Wikipedia for detecting query pairs that are not similar from a lexical content point of view, but actually semantically related.

III. TASK EXTRACTION

A task can be defined as a set of syntactically relevant query trails to satisfy need of a particular information. Two queries belongs to same task if they satisfy any of the following conditions: (1) they are indistinguishable; (2) one is a factor of the other (e.g., "Flipkart" and "Flipkart shopping"); (3) both somewhat agree to each other (e.g., "apple company" and "apple inc"); (4) one is a form of the other (e.g., "machnie learning" and "machine learning"). In the dataset construction process these rules can be used and an efficient clustering framework is proposed to group queries similar into same tasks.

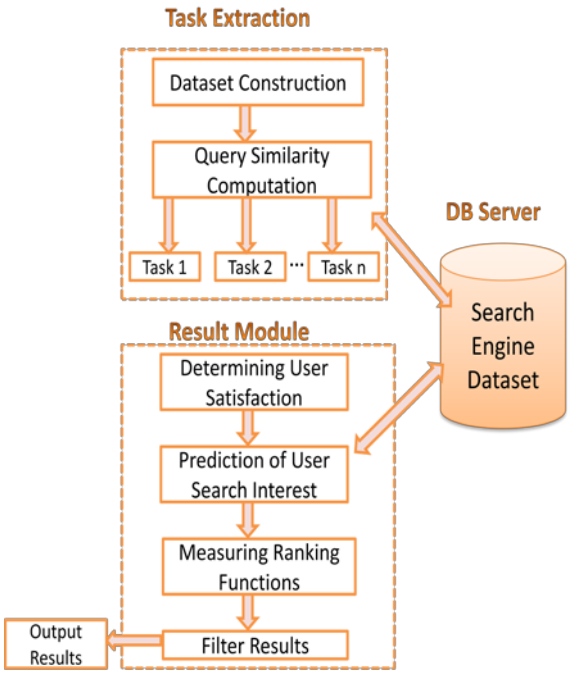

Fig. 1: System Architecture.

The basic clustering techniques ideas are described as follows. First, extract tasks out from each session; to segment logs into sessions follow the time threshold method by choosing a time threshold θ. Second, compute the similarity between any two queries. Last, cluster queries similar to each other are into the same task.

A. Query Similarity

To compute the similarity between two queries a linear SVM can be used. First, to learn a good query similarity function on labelled data for task classification a data set should be constructed. The labels include same task and different task. Here total 13 features are used 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

For the similarity function, the weight column in the table represents the weight of each feature. Here meaningless words are selected as the stop words from some frequent searched.

B. Clustering Queries into Tasks

Here is a method to extract accurately cross-session search tasks from users search activities. Search tasks frequently span multiple sessions and thus develop methods to extract these tasks from historic data to understand search behaviours. Cross session search task extraction 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 behaviours 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 1: *Spread Query Task Clustering (QC-SP)*

Input:Query set Q= {q1,q2…qN},cut-off threshold b;
Output: A set of tasks S;
Initialization: S = ∅; cid: content task id Query to task table M=∅;
1: // Initialize queries that are same into one task
2: cid=0;
3: for i = 1 to N do
4: if M[Qi] exists then
5: add Qi into S(M[Qi]);
6: else
7: M[Qi]=cid++;
8: if |S| = 1 return S;
9: for i = 1 to N do
10: // if two queries are not in the same task
11: if L[ ] ≠ L[+] then
12: T ← sim (L[Qi], L[Qi+N]);
13: if T ≥ b then
14: merge S(Qi) and S(Qi+N);
15: modify L;
16: // break if there is only one task
17: if |S| = 1 break;
18: return S;

The QC-SP algorithm finds the similarity between two queries. Based on the observation that consecutive query pairs than non-consecutive ones are likely belonging to same task, QC-SP prefers to first compute the consecutive query pairs similarities by timestamps. For example, given a series of queries {q1, q2, q3, q4}, first compute for pairs {q1 → q2, q2 → q3, q3 → q4}, it can reduce the computational cost from $O(k \cdot N^2)$ to $O(k \cdot N)$ if there is only one task in the session. Based on the statistics that about 50% sessions only have one task, QC-SP is efficient to identify them. For sessions with multiple tasks, QC-SP is also faster than standard implementation. For example, if the sequences {q1, q2, q3, q4} can be grouped into {q1} and {q2, q3, q4}, the standard approach enumerate all 6 query pairs but QC-SP only needs to compute 5 pairs while pair {q2q4} is skipped. That is because it skips computing the similarity of query pairs from the same task. In addition, QC-SP needs extra O(N) space for storing a query to task mapping table, which is affordable in current applications. This algorithm accurately clusters the queries into same and different tasks.

IV. SEARCH APPLICATIONS

Here presented the methods and metrics in search applications.

A. Determining User Satisfaction

To know whether the user is obtaining the findings from query executed by him on the search engine, considered features which take into account the entire pattern of user search behavior, including query, click and dwell-time as well as number of reformulations.

1. Clicks: Considering the last click of a session may be the most important piece of information in relating user clicks to document relevance. Clicks in a user search goal as well as the times between actions allow us to predict the user's success at that goal.
2. Dwell Time: Dwell time of a click is the amount of time between the click and the next action (query, click, or end). We calculate the dwell times for all clicks during goal and use the maximum, minimum, and average dwell times as features to predict success. It is widely believed that long dwell time of clicks is an important predictor of success.
3. Goal Success: We can build two Markov models to compute the probability of user success and failure. The Markov Model includes {clicks, queries, dwell time (>30 sec)} as states {Q, SS, SS_Long}, respectively. On the basis of labelled dataset, we split two Markov models. Given a new user task trail, we can compare the probability from successful and unsuccessful models and estimate the label of user satisfaction. By using the task success labels based on Hidden Markov Model, we can study the percentage of multitask sessions with both successful and unsuccessful task.

B. Prediction on User Search Interests

For improving ranking or personalization of search systems, user search interests can be captured. Since queries submitted by users reflect user information needs, used queries to represent user search interests. On the other hand, queries are often short and ambiguous. Therefore we can summarize user search interests at topic level. By taking previous co-session or co-task queries as context information to user's current query, we can construct different context models. To know which context model can predict user search interests better, we can compare topic similarities of co-session and co-task query pairs.

V. REFERENCES

[1] S. Fox, K. Karnawat, M. Mydland, S. Dumais, and T. White, "Evaluating implicit measures to improve web

search," ACM Trans. Inform. Syst., vol. 23, pp. 147–168, 2005.

[2] 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.

[3] R. White, P. Bennett, and S. Dumais, "Predicting short-term interests using activity-based search context," in Proc. 19th ACM Int. Conf. Inform. Knowl. Manage., 2010, pp. 1009–1018.

[4] H. Cao, D. Jiang, J. Pei, Q. He, Z. Liao, E. Chen, and H. Li, "Context-aware query suggestion by mining click-through and session data," in Proc. 14th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2008, pp. 875–883.

[5] B. Xiang, D. Jiang, J. Pei, X. Sun, E. Chen, and H. Li, "Context aware ranking in web search," in Proc. 33rd Int. ACM SIGIR Conf. Res. Develop. Inform. Retrieval, 2010, pp. 451–458.

[6] White, R., Bilenko, M. and Cucerzan, S., "Studying the use of popular destinations to enhance web search interaction," ser. SIGIR '07, 2007, pp. 159–166.

[7] D. G. He, A. G€oker, and D. J. Harper, "Combining evidence for automatic web session identification," Inform. Process. Manage., vol. 38, no. 5, pp. 727–742, 2002.

[8] C. Lucchese, S. Orlando, R. Perego, F. Silvestri, and G. Tolomei, "Identifying task-based sessions in search engine query logs," in Proc.4thACMInt.Conf.WebSearchDataMining, 2011, pp.277–286.