Estimation of User Satisfaction and Search Interest through Task Trail

Surabhi S.Golechha, Prof. R.R. Keole

***Abstract*—This paper analyze the user search behavior and the adequacy of task level search log analysis and compare it with session and query level search log analysis in real applications. We have adapted a new idea called Task Trail which represents all user activities. We proposed the clustering algorithms and perform expedient search operation compared to session and query trails. Web logs are used to determine the following: User Search Behavior, Prediction of User search interest and Determining User Satisfaction. Thus, the findings in this paper extract task trails from web search logs and reinforce applications in search and recommendation systems.**

***Index Terms*— Query trail, Session trail, Task trail, Log analysis, cluster algorithm**

I. Introduction

Web log [1] is a knapsack of valuable information that records users search queries and related actions on the internet. By mining the captured or recorded information, it is possible to accomplish the users underlying goals, interests and search behaviors. In order to collect information from web logs, the web logs should be segmented into sessions or tasks by clustering the queries. In this work, Task Trail is introduced to understand user search behaviors. Logs containing the search engine interactions of many users have been mined extensively to enhance search-result ranking. Search trails comprising a query and post-query page views can be mined from these logs. Although trail components origins (clicked search results) and destinations (trail end points) have been used previously to support search, the typical application of trails is to better rank Web pages [2]. Search trails are a series of web pages starting with a search query and terminating with an event such as session inactivity. Although the traversal of trails following a query is common, about how much value users derive from following the trail versus sticking with the origin (the clicked search result) or jumping to the destination page at the end of the trail [3]. In this paper we present a clustering algorithm evaluating the result of queries to users of traversing multi-page search trails.

The searching activities of users are captured by web search log. Web search log can be used in various applications like Prediction of user search interest, website recommendation, web page re-ranking methods and query Suggestion.

In this paper, we compare the performances of session, query, and task trails in applications including:
a) Determination of user satisfaction where dwell time (amount of time between click and action) and success score of markov models [4] are mined to measure user satisfaction
b) Prediction of user search interest where semantic information is used to measure topic similarity.

Dwell Time: It is the amount of time between the click and next action. This time, is a good indicator for user satisfaction. The more the dwell time the more is the success of the search. The behavior of set of users in the web log may be either search behavior or browse behavior. Search behavior is a single query submitted to the search engine. Browse behavior may be one of the following: 1) Starting to surf from the home page. 2) Typing a URL address 3) Pasting a URL address from another page to the address bar of the existing webpage. 4) User clicks a bookmark or a back or forward button in a browser. Log Segmentation can be done in any one of the following trails:

Query Trail: It represents a sequence of user behaviors of one of the user starting from a query followed by sequence of browsing behaviors that are triggered by this query.

Session Trail: It represents a sequence of user behaviors of one of the user where user behaviors are consecutive and any two consecutive occurred within the time threshold.

II. Proposed Objectives

The Study described in this paper consistently analyzes the utilities of task level search log analysis.

Also focus on the comparison of task, session and query trails in the applications of user search behavior, determining user satisfaction, predicting user search interests. To use task trail as a useful segmentation of user search behaviors as well as to evaluate the expedient of task trails.

To extract task trails from web search logs, we proposed two clustering algorithms and suggest potential applications of using task trails in search recommendation systems.

Manuscript received Nov, 2015.
***Surabhi S.Golechha**, Computer Science and Information Technology, H.V.P.M. College of Engineering & Technology, Amravati, India, 8983799909*
***Prof. R.R. Keole,** Computer Science and Information Technology, H.V.P.M. College of Engineering & Technology, Amravati, India, 9823852893*

ISSN: 2278 – 7798

This paper consider task trail as a peculiar way to segment search logs and an additional information source to classic session and query trails.

III. Literature Review

R. White and J. Haung [5] represented a log-based study, the findings have implications for the design of search systems, including trail recommendation systems that show trails on search result pages.

D.Beeferman and A. Berger et.al [6] proposed an agglomerative clustering algorithm[4] for the segmentation of web search log. It is a task-level segmentation method.

B.Xiang, D. Jiang, J.Pei, Q. He, Z. Liao [7] studied the problem of using context information in ranking documents in web search.

A. Hassan, R. Jones, and K. Klinkner [8] found the problem of predicting user search goal success by modeling user behavior. It shows empirically that user behavior alone can give an accurate picture of the success of the user's web search goals, without considering the relevance of the documents displayed.

F. Radlinski and N. Craswell [9] presented a detailed comparison between performance as measured by judgments-based information retrieval metrics and performance as measured by usage-based interleaving on five real pairs of web search ranking functions.

A. Singla, R. White, J. Huang quantified the benefit that users currently obtain from trail following and compare different methods for finding the best trail for a given query and each top-ranked result.

A. Hassan, Y. Song, L.-w. He [10] performed a large scale user study collected explicit judgments of user satisfaction with the entire search task. Results were analyzed using sequence models that incorporate user behavior to predict whether the user ended up being satisfied with a search or not.

C. Lucchesez, S. Orlandoy, R. Peregoz, F. Silvestriz, G. Tolomeizy [11] proposed a clustering-based solution, leveraging distance measures based on query content and semantics, while query timestamps were used for a first preprocessing breaking phase.

Y. Song, D. Zhou, L. w. He [12] presented a novel query suggestion framework which extracted user preference data from user sessions in search engine logs, then used the user patterns to build two suggestion models.

H. Wang, Y. Song, M.-W. Chang, X. He, R. White, and W. Chu [13] targeted the identification of long-term, or cross-session, search tasks (transcending session boundaries) by investigating inter-query dependencies learned from users searching behaviors.

H. Feild and J.Allan introduced a novel generalized model for generating recommendations over a search context. While only considered query text in this study, the model can handle integration over arbitrary user search behavior, such as page visits, dwell times, and query abandonment. In addition, it can be used for other types of recommendation, including personalized web search.

Zhen Liao introduced "task trail" to understand user search behaviors. It conducted extensive analyses and comparisons to evaluate the effectiveness of task trails in several search applications: determining user satisfaction, predicting user search interests, and suggesting related queries

IV. Proposed System & Methodology

Task Extraction**:** A task can be defined as a set of semantically appropriate query trails within a session. Two queries can be grouped into same task if they satisfy any of the rules described in clustering schema. Those rules can be used in the annotation process and proposes an efficient clustering framework to group queries into tasks. The basic ideas of our clustering framework are described as follows. First, since tasks are extracted out from each session, we follow the time threshold method to segment logs into sessions by choosing a time threshold *ot*. We quantitatively compute the similarity between any two queries. Last, queries similar to each other are clustered into the same task. We proposed two efficient algorithms to group similar queries into tasks. We present the details about the query similarity function and clustering algorithms with empirically evaluation in following sections.

To conquer the shortcomings of session and query trails we have introduced Task trails.

Task Trail: It represents a series of user behaviors of one user within one session where all user behaviors collectively define an atomic user information need.

Task segmentation contains two steps:

1. Logs are segmented into sessions based on threshold.
2. Segment session into tasks based on semantic relationships between queries.

Advantages

1.Task trail performs better than session and query trail in determination of user satisfaction.

2. Page utilities can be increased by Task trail.

3. Topic similarity can be well preserved by task trail as because it provides atomic user information needs And the user interest can be predicted.

Clustering Schema:

Query trail can be represented by starting search query of each query trail. We can combine two queries into the same task if a) Two queries are identical b) One is a part of the other(e.g., "ice-cream" and "ice-cream flavour"); c) Two partially agree to each other(e.g., "book rate" and "book price"); d) One is the type of the other(e.g., "oxford

ISSN: 2278 – 7798

university” and “oxfrod university”). The basic rules of clustering schema are

1) Logs are segmented into sessions by using some threshold.
2) Similarity between two queries is computed.
3) Similar queries are clustered into the same task.

We build an undirected graph for queries within a session. Vertices of graph indicate queries and an edge indicates similarity scores between queries. The edges having the score below than threshold are skeptical edges and they are to be discarded. After discarding the skeptical edges, any connected component of the remaining graph is identified as a task. This approach is called query clustering using weighted connected component of a graph (QC-WCC). However this method has disadvantage that it has high time complexity for building the graph [O $(K \cdot N^2)$] were N is the numbering of queries and K is the dimension of features. Computing the pair wise similarity for all consecutive query pairs is the better approximation.

Algorithm 1: Query Clustering (QC).

Input: Query set Q, cut-off threshold ot;
Output: A set of tasks Θ;
Initialization: $\Theta = \Phi$; Query to task table $L=\Phi$;
Length= l;

```
for l = 1 : |Q| − 1 do
  for i = 1 : |Q| − 1 do
    // if two queries are not in the similar task
    if L [Q_i] ≠ L [Q_{i+l}] then
      // compute similarity takes O (k)
      s ← sim (L [Q_i]; L [Q_{i+l}]);
      if s ≥ ot then
        merge Θ (Q_i) and Θ (Q_{i+l});
        modify L;
  // break if there is only one task
  if |Θ| = 1 break;
return Θ;
```

By the above algorithm we examined that consecutive query pairs are more likely belonging to same tasks compared to non-consecutive query. In this, we will evaluate the similarities for consecutive query pairs. Consider an example consisting of sequence of queries, this algorithm will evaluate for pairs $\{q_1 q_2, q_2 q_3, q_3 q_4\}$ This will take a time complexity of O (K.N) if the sequences $\{q_1, q_2, q_3, q_4\}$ are grouped into $\{q_1\}$ and $\{q_2, q_3, q_4\}$ the standard approach will compute all six query pairs but BS-QC needs to calculate for five query pairs only. Since the query pairs are similar, $\{q_2\ q_3\}$ is skipped.

Head tail component query clustering (QC-HTC)[11] uses the heuristics that queries are submitted sequentially by users. It violates the task interleaving observations found by us in search logs. So an algorithm called query clustering using bounded spread method (BS-QC) is used.

From the intuition that consecutive queries more likely belong to the same task than non-consecutive ones, a better approximation is to compute the pair-wise similarity for all consecutive query pairs. In this work, a clustering algorithm Query Clustering using Bounded Spread method (BS-QC) is proposed for task extraction, as shown in Algorithm 2.

Algorithm 2: Bounded Spread Query Clustering (BS-QC)

Input: Query set Q, cut-off threshold ot; bounded length bl;
Output: A set of tasks Θ;
Initialization: $\Theta = \Phi$; Query to task table $L=\Phi$;
M= Φ; *length*= l

```
// initialize same queries into one task
cid=0;
for i = 1: |Q| − 1 do
  if M [Q_i] exists then
    add Q_i into Θ (M [Qi]);
  else
    M [Q_i] = cid++
if |Θ| = 1 return Θ;
for len = 1 : bl do
  for i = 1 : |Q| − 1 do
    // if two queries are not in the same task
    if L [Q_i] ≠ L [Q_{i+l}] then
      // compute similarity takes O (k)
      s ← sim (L [Q_i];L [Q_{i+l}]);
      if s ≥ ct then
        merge Θ (Q_i) and Θ (Q_{i+l});
          modify L;
  // break if there is only one task
  if |Θ| = 1 break;
return Θ;
```

For the experimental study, web log data set of a particular user is extracted. The data set consists of user browsing logs and search logs from a widely used browser. It contains URL visits of the user and also contains session ID, User clicked/visited URLs, as well as queries related to user clicks, a referrer URL where current URL comes from and Time stamps of user events. The web log is segmented at task level, so that the user search behavior can be easily identified.

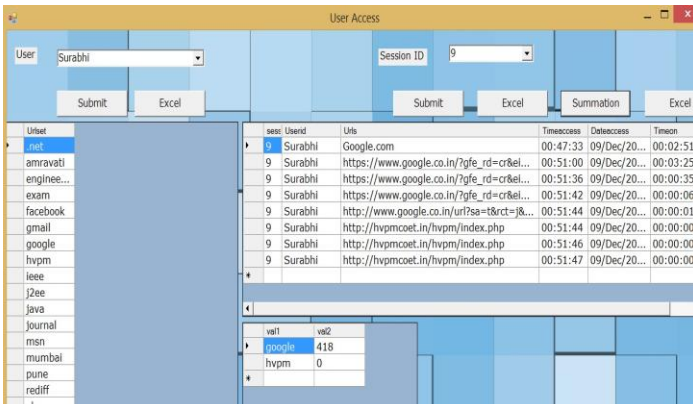

Figure 1: User Search Behavior Analysis

From the experimental results it can be seen that, the BSP clustering algorithm does not consider the semantic

ISSN: 2278 – 7798

similarity between the query words. But in the case of modified BS-QC algorithm considers the semantic similarity between the query words also. Figure 1 represents the task identification for BS-QC clustering algorithm corresponds to different query words.

V. Results and Analysis

Here, we present experimental observations and results on estimating the effectiveness of task trails in real applications.

A. User Satisfaction Determination

After the search process, to understand whether a user was satisfied or not in search process, several indirect feedback signals can be used as measures.

Clicks: The total number of clicks to perform a particular task can be taken as a signal of user satisfaction on that task. Clicking on search results often indicates the relevance between queries and clicked pages.

Dwell time: Dwell time can also be considered as a signal of user satisfaction. It is because users are more likely to stay on useful pages.

Markov Model Success Score: The Markov model can be used to create the user's search activities as a sequential process. The Markov model takes queries, clicks, dwell time (> 35seconds) as states Q, C, C$Long$, respectively. Two Markov models can be built to evaluate the likelihood of user satisfaction and dissatisfaction. When a new user's search activities are given, the score of Markov Models can be estimated to determine the label of user satisfaction.

Table 1
User Satisfaction Using Different Signals at Different Levels

	Session	Task	Query
Click Rate	0.467	0.472	0.459
35s-Click Rate	0.209	0.221	0.209
MM-Success Rate	0.506	0.510	0.501

In the experiments, we computed the average ClickRate, 35sClick Rate, and MMSuccessRate at query, task, and session levels. The results on both browse and search logs are reported in Table 7. From the table, we can observe that: the task-level user satisfaction rates are higher than those at session and query levels (all p-values <0:01, t-test). Before illustrating the reasons of the results, we show a simple example as follows. Suppose a user issued 4 queries {q1, q2, q3, q4} while in the first query he went to a daily used website (e.g., Flipkart) and come back to search something but gave up after three trials. Therefore, the first query {q1} belongs to a task TA and the rest queries {q2, q3, q4} belong to another task TB. At both session and query levels, the average success rate is 0:25 (one query successes and three fails). However, at task level, the average success rate is 0:5 (one task TA successes and the other TB fails). Based on previous studies [10], failed tasks tend to contain more failed queries since the user may try more times. If our task extraction method can group user's queries as atomic information needs, then the satisfaction score of a failed query is added to its own task. Then a failed query is likely belonging to a failed task which is longer than other tasks. Then we can infer that shorter tasks tend to have higher satisfaction scores than longer tasks. When we calculate the average user satisfaction rate at task level, the result can be higher than that of query or session level since several failed queries can be grouped by only one failed task. If the experiment results verify this conjecture, then our task extraction method is likely to be correct. (Note that although the average user satisfaction rate is higher at our extracted task level, it may not lead to the correct task extraction.) From another side, it also shows that task-level user satisfaction rates are more precise than session and query levels. As we can observe from Table 1, task level user satisfaction rates are higher than session and query levels among all implicit measures. The results indicate that extracting tasks from sessions is not so important, since we can more accurately capture users success or fail search experience in task level, which is to use logs for determining user satisfaction.

B. User Interest Prediction

User search interests can be represented by their queries. Summarizing queries into topics can help understanding user search interests at a higher level. Given two queries submitted by one user, they may come from: (1) different sessions (inter-sessions); (2) same session (intra-session); (3) different tasks in different sessions (inter-tasks among sessions); (4) different tasks in same session (inter-tasks within sessions); (5) same task in same session (intra-task). All these five sources can provide query pairs. Besides, capturing user search interests at topic level is useful to understand user behaviors. For example, average topic similarity between query pairs from different sessions can help tracing the user search interests during a relative long period. Topic similarity between query pairs from same session can reflect user search interests in a relative short time.

VI. Conclusion

Segmentation of web log can be done in query level, session level, or task level. Task trail is a sequence of user behaviors occurred within a session, where they define atomic user information need collectively. Task trail is an effective method to segment the web log and also to determine the user search behavior. Web logs are segmented into sessions by choosing a time threshold. Queries similar to each other are clustered into same task after computing the query similarity by using the clustering algorithms. From the extracted tasks, user search behavior can be determined. We have shown that a supervised Markov model of user search behavior including the sequence of all queries and clicks in a user search behavior as well as the times between actions allow us to predict the user satisfaction.

ISSN: 2278 – 7798

ACKNOWLEDGMENT

My thanks to the guide Prof.R.R.Keole and Principal Dr.A.B.Marathe, for providing me constructive and positive feedback during the preparation of this paper.

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] L. D. Catledge and J. E. Pitkow, ―Characterizing browsing strategies in the world-wide web,‖ Comput. Netw. ISDN Syst., vol. 27, no. 6, pp. 1065–1073, 1995.

[3] 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

[4] A. Hassan, Y. Song, and L.-w. He, "A task level user satisfaction metric and its application on improving relevance estimation," in Proc.20thACMInt.Conf.Inform.Knowl.Manage.,2011,pp.125–134.

[5] 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.

[6] D. Beeferman, A. Berger, "Agglomerative Clustering of a Search Engine Query Log", Proc. ACM SIGKDD,2000.

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

[8] A. Hassan, R. Jones, and K. Klinkner, ―Beyond DCG: User behavior as a predictor of a successful search,‖ in Proc. 3rd ACM Int. Conf. Web Search Data Mining, 2010, pp. 221–230.

[9] F. Radlinski and N. Craswell, ―Comparing the sensitivity of information retrieval metrics,‖ in Proc. 33rd Int. ACM SIGIR Conf. Res. Develop. Inform. Retrieval, 2010, pp. 667–674.

[10] A. Hassan, Y. Song, and L.-w. He, ―A task level user satisfaction metric and its application on improving relevance estimation,‖ in Proc. 20thACMInt. Conf. Inform. Knowl. Manage., 2011, pp. 125–134.

[11] 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.

[12] Y. Song, D. Zhou, and L.-w. He, "Query suggestion by construct- ing term-transition graphs," in Proc. 5th ACM Int. Conf. Web Search Data Mining, 2012, pp. 353–362.

[13] H. Wang, Y. Song, M.-W. Chang, X. He, R. White, and W. Chu, "Learning to extract cross-session search tasks," in Proc. 22nd Int. Conf. World Wide Web, 2013, pp. 1353–1364.

Surabhi S. Golechha, received the B.E. degree in Computer Science from H.V.P.M's College of Engineering & Technology, Amravati in 2013. She is currently pursuing Master's Degree in Computer Science and Information Technology from H.V.P.M's College of Engineering & Technology, Amravati.

Prof. R. R.Keole, received the B.E. and M.E. degree in Computer Science from Prof. Ram Meghe Institute of Technology, Badnera in 1992 and 2008, respectively. His field of specialization is Web & Data Mining. He is currently working as an Assistant Professor at H.V.P.M's College of Engineering & Technology, Amravati.

ISSN: 2278 – 7798