ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 3, March 2015

Personalized Search of User Search Behaviour with Ontology

N.Hemamalini, M.Gomati, V.Indumathi, S.Jegadeesan

UG Final Year Student, Dept. of I.T., Velammal Institute of Technology, Chennai, India.

UG Final Year Student, Dept. of I.T., Velammal Institute of Technology, Chennai, India

UG Final Year Student, Dept. of I.T., Velammal Institute of Technology, Chennai, India.

Assistant Professor, Dept. of I.T., Velammal Institute of Technology, Chennai, India.

ABSTRACT: In this paper, we introduce "Ontology and generic programming" to understand user search behaviors. Personalized search is an important research area that aims to resolve the ambiguity of query terms. To increase the relevance of search results, personalized search engines create user profiles to capture the users' personal preferences and as such identify the actual goal of the input query. Since users are usually reluctant to explicitly provide their preferences due to the extra manual effort involved, recent research has focused on the automatic learning of user preferences from users' search histories or browsed documents and the development of personalized systems based on the learned user preferences. Most personalization methods focused on the creation of one single profile for a user and applied the same profile to all of the user's queries. We believe that different queries from a user should be handled differently because a user's preferences may vary across queries. In this paper, we conduct extensive analyses and comparisons to evaluate the effectiveness of ontology in several search applications: determining user satisfaction, predicting user search interests, and suggesting related queries. Experiments on large scale datasets of a commercial search engine show that: (1) ontology performs better than session, query and task trails in determining user satisfaction; (2) Ontology increases web page utilities of end users comparing to session, query and task trails ; (3) generic programming is more sensitive than other trail methods in measuring different ranking functions; (4) Query suggestion based on ontology is a good complement of query suggestions based on session trail and click-through bipartite. The findings in this paper verify the need of extracting ontology from web search logs and enhance applications in search and recommendation systems.

KEYWORDS: Semantic based search, Segmentation of Task, Task Analysis, Ontology based personalized search, Plug rising unauthorized access

I. INTRODUCTION

Web search logs record the searching activities of users in search engines. previous studies have shown that search logs can be used in various applications including user satisfaction analysis [1], page utility estimation [2], user search interest prediction [3], query suggestion [4], web page re-ranking [5], web site recommendation [6], etc. most of previous work analysed web search logs at session or query level, where a session is defined as "a series of queries by a single user made within a small range of time" [6], [7].however, few of them have considered search logs at task (atomic user information need) level. Consider the example shown in table 1, which is a real user search session from bing (http://www.bing. com). This session contains 4 different search tasks: facebook, amazon kindle books, Gmail, and lyrics of a song. The "Gmail" task is interleaved with the "amazon kindle books" task. the reasons causing the interleave phenomenon are: (1) web search logs are ordered chronologically; (2) users often open several tabs or browsers and conduct multiple tasks at the same timetable 1 indicates that the granularity of session is too coarse to manifest details of user behaviors by missing multiple tasks within a session. The query level analysis is the finest grained, but fails to capture the interleave relationships between tasks and the generalization/specification/refinement relationships between adjacent queries within the same task. our analytics based on search logs of 159, 668, 543 users in 3 months find that following a widely used session definition (30 minutes timeout [8], [9], about 30% of the sessions contain multiple tasks and about 5% of the sessions contain interleaved tasks. thus, task-level log analysis strikes a good balance between details of user behaviors and relationships between queries. Previous work attempted to enhance session-level analysis by distilling semantic features such as query reformulation patterns [10], [11] into the session boundary detection. however, the tasks have not been explicitly identified. Jones et al. [12] first extracted tasks from

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 3, March 2015

sessions based on time and query word features. Lucchese et al. [13] further improved task identification by leveraging external encyclopaedias like Wikipedia. however, all these work only focuses on task extraction.

II. RELATED WORK

Web search logs record user activities on search engines, such as queries and clicks. Search trails record the footprints left by users in their search processes. in the literature of studying search trails, much of previous work have applied to them in the applications of user satisfaction analysis, ranking function evaluation, query suggestion, etc. here we classify related works into four categories: (1) user behaviour segmentation, (2)user satisfaction and interest analysis, (3) enhance ranking based on web logs, and (4) query suggestion. 2.1 user behavior segmentation session and task are two primary types of user search behavior segmentations. The term session was proposed in [7], [8]. Cat ledge et al. [8] analysed user browsing logs captured from client-side user events. They found that 25.5 minutes timeout is good or separating consecutive user activities into different sessions Silberstein et al. [7] defined "session" as "a series of queries by a single user made within a small range of time" in their study of AltaVista search logs, where they used 5 minutes as the timeout threshold. he et al. [10] proposed to detect session boundary based on time and query reformulation patterns and found that 10 to 12 minutes timeouts are good. Jansen et al. [11] clarified the session as "a series of interactions by the user towards addressing a single information need" and found that about 30 minutes timeout is better than others. As a result, later studies [4]–[6], [11] often used 30 minutes timeout for session segmentation.

Considering the multitasking behaviors within a session, Jones and Klinkler [12] proposed to classify query pairs into a same task via features based on time, word, web search results, etc. Their approach achieved about 90% accuracy in task boundary detection and same task identification. Boldi et al. applied the query flow graph in finding logical session and query recommendation. They formulated the problem of mining logical sessions as an Asymmetric travelling Salesman Problem. Lucchese et al. [13] proposed to identify task-based sessions by combining content (query word, edit distance) and semantic (Wikipedia) features. Donato et al. proposed to identify those complex tasks as research missions which need users to explore multiple pages. Kotov et al. proposed to model and analyse cross-session search tasks, and they applied classification approach to predict the re visiting likelihood of tasks. In this paper, we adapt methods described in [12], [13] to extract tasks from sessions. We learn the query similarity function via machine learning and cluster queries within a session into tasks.

III. PROPOSED ALGORITHM

A. *Design Considerations:*

- Create User Interface
- Search log mining
- Task evaluation
- Log analysis
- Task trail
- Overall Report Generation

.

B. *Description of the Proposed Algorithm:*

The study described in this paper differs from previous work in that we focus on the comparison of personalized search in the applications of determining user satisfaction, predicting user search interests, suggesting related queries and measuring ranking functions, rather than identifying session boundary [10], [11], extracting tasks from session [12], [13], or estimating web page relevance using ontology [5], [25], [27]. As a result, ontology can be considered as a novel way to segment search logs and an additional information source to classic session and query trails.

Label	Query A	Query B
Same Task	gmail.com Florida statutes	Login gmail Florida evidence code
Diff. Task	facebook.com definitions of tarsorraphy	Fall out 3 books at twireless
Unknown	Sunday night football Snow mobiling in	Nbc sports nfl Snow mobile parts

TABLE 1

Examples of labeled query pairs.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 3, March 2015

Labels include *same task*, *different task* or *unknown*. The final label of each query pair was obtained by voting. After labeling, we totally obtained 5,668 positive pairs, 9,370 negative pairs, and 1,334 unknown pairs. We ignored the unknown pairs since they are hard to understand even by human annotators. To better understand labeling results, we show some labeled query pairs in Table 2.

Feature Description	Weight
Temporal features	
Time diff1:time difference in seconds	-0.1121
Time diff2:category for 1/5/10/30mins	-0.0623
Word features	
Lv_1:Levenshtein distance of two queries	0.0106
Lv_2:lv1after 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_1 : rate of longest common substring	0.4941
b1:1ifonequerycontainsanother,else0	0.6361

TABLE 3

Features of Query pair.

We present details of features in Table 3, where 215 frequent searched but meaningless words are selected as stop words. The column *weight* in the table lists the weight of each feature for similarity function. We obtained the weights by training a linear SVM model [33] on labeled data.We chose linear-SVM as classifier because of its good performance in many applications and theoretical soundness, and recent study [34] also shows that it is a state-of-art method in computing query similarity. The whole labeled dataset was split into 5 folds for cross validation. Each time 3 folds were used for training, 1 fold was used for tuning parameter (C in SVM), and the rest 1 fold was used for testing.

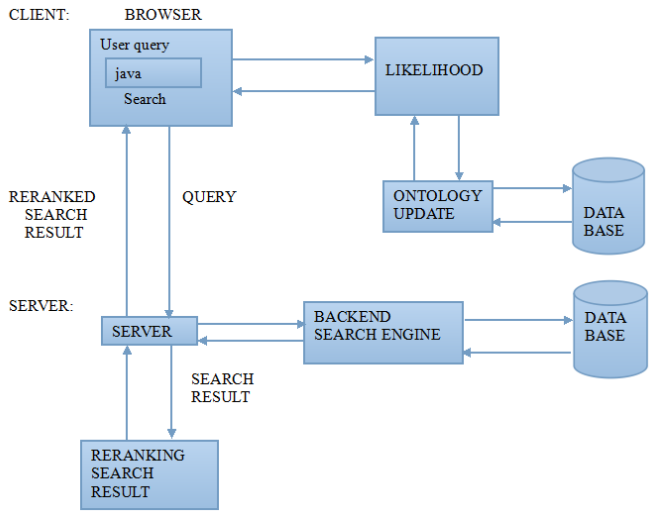


Fig: ARCHITECHTURE DIAGRAM

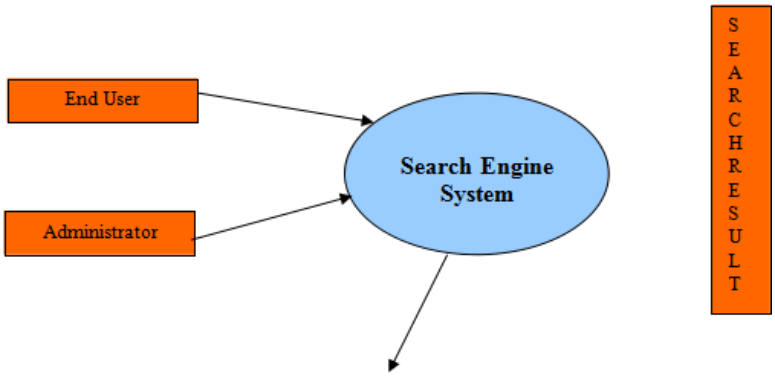
In this section, we present detailed experimental observations and results on evaluating the effectiveness of task trails in real applications. We first show the statistics on the datasets used in the experiments, and we present the methods, metrics and findings in three search applications (determining user satisfaction, predicting user search interests, and suggesting related queries). We also present the application of measuring ranking functions in the section of analyzing user satisfaction.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 3, March 2015



Data flow diagram shows how the personalized browser works

We extracted two log datasets for the experiments. The first dataset D0 consists of user browsing logs from a widely used browser plug-in (e.g., toolbar). It contains URL visits by anonymous users who opted in to provide data. The second dataset D1 consists of web search logs from Bing. The information in these datasets contains: (1) user anonymous unique identifier (machine ID). (2) A unique browser identifier. (3) User clicked/visited URLs as well as queries related to user clicks. (4) A referrer URL where current URL comes from. (5) Time stamps of user events. For preserving user privacy, intra-net and secure URLs (such as URLs beginning with https are not recorded. Both datasets are from May to June 2011 in United States search market where main language is English.

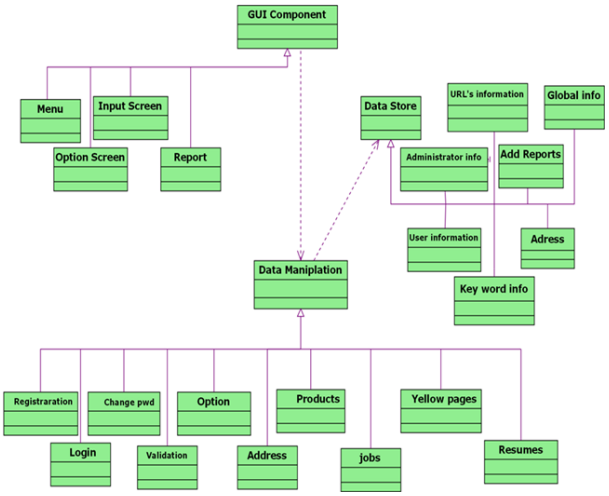


Fig: Uml Class diagram on how the personalized search works

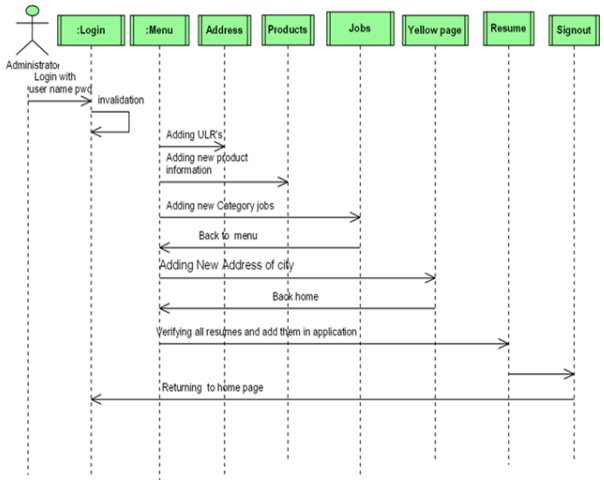
To further clean the data, we pre-processed datasets as follows: (1) filtering sessions which have no search event (such as checking emails) or too many search events (which are likely generated by robots); (2) filtering entries with non-English language settings, e.g., users searched in other languages; (3) keeping those sessions with search events in "Web" search vertical and filtering verticals like "Image", "Video", etc. (4) keeping sessions with search events from Google, Bing and Yahoo! since these are main search engines in U.S. market. The dataset D0 contains 2, 673, 335 unique users, and dataset D1 contains 159, 668, 543 unique users. We tried different thresholds θ to extract session ranging from 1 minute, 2 minutes, 5 minutes, 10 minutes, 20 minutes, 30 minutes, 60 minutes to a day.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 3, March 2015



Sequence diagram shows the step by step process of function

The purpose of using different thresholds is to illustrate the phenomenon of multi-task and interleave-task behavior in user search behavior. We report the multi task and interleave-task rate of extracted sessions in Table 5. From the table, we have the following ob servations: (1) the multi-task behavior always exists in the search process, even if we set the timeout of session as one minute. (2) The interleave-tasking behavior is not obviously while the threshold θ is less than 5 minutes.

IV. ALGORITHM AND DESCRIPTION

#Algorithm 1: Spread Query Task Clustering (QC-SP).
Input: Query set *Q*, cut-off threshold *b*;
Output: A set of tasks _ ;
Initialization: _ = ∅; Query to task table *L*=∅;
1: **for** *len* = 1 : |*Q*| − 1 **do**
2: **for** *i* = 1 : |*Q*| − *len* **do**
3: // if two queries are not in the same task
4: **if** *L*[*Qi*]≠ *L*[*Qi*+*len*] **then**
5: // compute similarity takes *O*(*k*)
6: *s* ← sim(*L*[*Qi*];*L*[*Qi*+*len*]);
7: **if** *s* ≥ *b* **then**
8: merge _(*Qi*) and _(*Qi*+*len*);
9: modify *L*;
10: // break if there is only one task
11: **if** |_| = 1 break;
12: return _;

#Algorithm 2: Bounded Spread Query Task Clustering (QC-BSP).
Input: Query set *Q*, cut-off threshold *b*, bounded length *bl*;
Output: A set of tasks _;
Initialization: _ = ∅; Query to task table *L*=∅, *M*=∅;
1: // initialize same queries into one task
2: *cid*=0;
3: **for** *i* = 1 : |*Q*| − *len* **do**
4: **if** *M*[*Qi*] exists **then**
5: add *Qi* into _(*M*[*Qi*]);
6: **else**
7: *M*[*Qi*]=*cid*++;
8: **if** |_| = 1 **return** _;
9: **for** *len* = 1 : *bl* **do**
10: **for** *i* = 1 : |*Q*| − *len* **do**
11: // if two queries are not in the same task
12: **if** *L*[*Qi*]≠ *L*[*Qi*+*len*] **then**
13: // compute similarity takes *O*(*k*)
14: *s* ← sim(*L*[*Qi*];*L*[*Qi*+*len*]);
15: **if** *s* ≥ *b* **then**
16: merge _(*Qi*) and _(*Qi*+*len*);
17: modify *L*;
18: // break if there is only one task
19: **if** |_| = 1 break;
20:**return**_;

A Genetic Algorithm searches for the global optimum by mimicking the process of selection in search result. A GA iteratively generates a set of solutions known as a search data's in database. At each iteration, a GA selects a subset of solutions to form a new data search.

Step 1: [Start] Generate random search (suitable solutions for the problem)
Step 2: Already search performed or not.
Step 3: If not create a new search by repeating following steps until the new search is complete
Step 4: Select two queries that relate to each other.
Step 5: Store the query in the database

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 3, March 2015

Step 6: Perform ontology update for priority.
Step 7: If yes
Fetch the data in database
Perform ontology update
Step 8: Show the result.
Step 9: Stop.

V. SIMULATION RESULTS

Finally, we studied several queries from high, middle, and low frequent parts with their suggestions. From the table, we can find that: (1) ontology models often generate related queries in a broad range such as provide "Verizon" as a suggestion to "att". (2) Click through approach may generate suggestions which are too similar to test queries, such as providing "at & t" as a suggestion to "att". That is why random walk approach does not perform best on high and middle frequent queries. (3) For low frequent queries, task-based and session-based methods generate nearly same suggestions. This is because low frequent queries exist in fewer sessions and tasks and they do not have many alternative searches from sessions. (4) Ontology methods often generate more specific queries for further narrowing down user's information need, which are different from ontology and personalized search approaches. As a result, suggestions provided by task-based methods can be treated as a complementary source to the results provided by ontology based and click through approaches

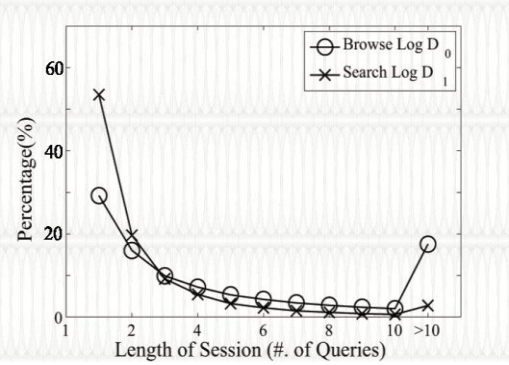
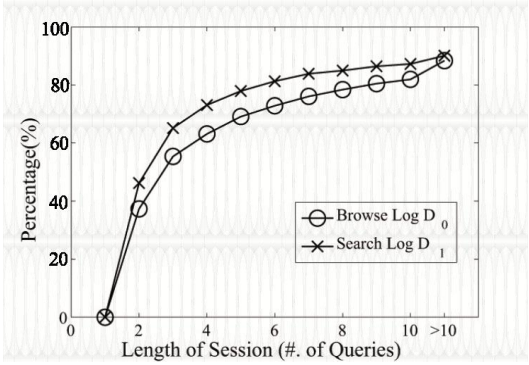
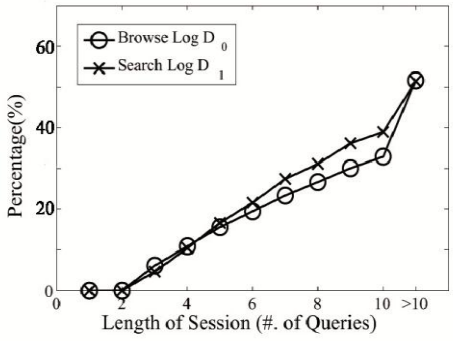


Fig a, b & c: Search and task distribution in browse and search Logs: (a) distribution of session length, (b) distribution of multi-task, (c) distribution of interleave-task. Note that the percentage of sessions is computed in total but the percentages of multi-task and interleaved-task sessions are



(b)



(c)

VI. CONCLUSION AND FUTURE WORK

In this paper we proposed to use task trail as a useful segmentation of user search behaviors. Users often perform multiple tasks during their search processes. Statistical results on 0:5 billion sessions from web search logs showed that: (1) about 30% of sessions contain multiple tasks, and (2) about 5% of sessions contain interleaved tasks. To evaluate the effectiveness of task trails, we compared task, session and query trails in determining user satisfaction, predicting user search interests, and suggesting related queries. First, comparing to session and query trails, task trail is more precise to determine user satisfaction. Second, users are more likely to find useful information following the task trails. Third, we found that measuring ranking functions at task level is comparable to query level and more sensitive

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 3, March 2015

than session level. Forth, since tasks represent atomic user information needs, they can well preserve topic similarity between query pairs. Last but not least, we found that task-based query suggestion can provide complementary results to other models. These findings verify the need to extract tasks from web search logs and suggest potential applications of using task trails in search and recommendation systems.

REFERENCES

1. White, R., Bennett, P. and Dumais, S., "Predicting short-term interests using activity-based search context," ser. CIKM '10, 2010, pp. 1009–1018.
2. White, R. and Huang, J., "Assessing the scenic route: measuring the value of search trails in web logs," ser. SIGIR '10. ACM, 2010, pp. 587–594.
3. Radlinski, F. and Craswell, N., "Comparing the sensitivity of information retrieval metrics," ser. SIGIR '10, 2010, pp. 667– 674.
4. Xiang, B., Jiang, D., Pei, J., Sun, X., Chen, E. and Li, H. "Context-aware ranking in web search," ser. SIGIR '10. ACM, 2010, pp. 451–458.
5. Liao, Z., Song, Y., He, L.-w. and Huang, Y., "Evaluating the effectiveness of search task trails," ser. WWW '12, 2012, pp.
6. 489-498.
7. Lucchese, C., Orlando, S., Perego, R., Silvestri, F., and Tolomei, G., "Identifying task-based sessions in search engine query logs," ser. WSDM '11, 2011, pp. 277–286.
8. Kotov, A., Bennett, P., White, R., Dumais, S. and Teevan, J., "Modeling and analysis of cross-session search tasks," ser. SIGIR '11, 2011, pp. 5–14.
9. Wang, H., Song, Y., Chang, M.-W., He, X., White, R. and Chu W., "Learning to extract cross-session search tasks," in *WWW*, 2013, pp. 1353–1364.

BIOGRAPHY

Hemamalini N is a final year student in the Information Technology Department at Velammal Institute of Technology., Chennai.

Gomati M is a final year student in the Information Technology Department at Velammal Institute of Technology., Chennai.

Indumati V is a final year student in the Information Technology Department at Velammal Institute of Technology., Chennai.

Guided By: S.Jegadeesan, received B.Tech Information Technology from Anna University in 2005 and M.E Computer Science and Engineering from Anna University in 2010. He is working as an Assistant Professor, Velammal Institute of Technology, Chennai.