International Journal of Computer Applications Technology and Research
Volume 4– Issue 4, 229 - 234, 2015, ISSN:- 2319–8656

# Quest Trail: An Effective Approach for Construction of Personalized Search Engine

M. Therasa
Anna University
Panimalar Institute of Technology
Chennai, India

S.M. Poonkuzhali
Anna University
Panimalar Institute of Technology
Chennai, India

S. Hemamalini
Anna University
Panimalar Institute of Technology
Chennai, India

**ABSTRACT:** Personalized search refers to search experiences that are tailored specifically to an individual's interests by incorporating information about the individual beyond specific query provided. Especially people working in a software development organization (analysts, developers, testers, maintenance team members), find it increasingly difficult to get relevant results to their searches. We propose methods to personalize searches by resolving the ambiguity of query terms, and increase the relevance of search results in order to match the user's interests. Difficulty in web searches has given rise to the need for development of personalized search engines. 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, the search engine faces the entire burden of predicting the user's preferences and intentions behind a query in order to yield more relevant search results. In this paper we define a QUEST to be the objective of user's search; here we combine quest level analysis of user's search logs and semantic analysis of the user's query in order to personalize user's search results. Most personalization methods focus on the creation of one single profile for a user and apply the same profile to all of the user's queries. Hence we propose a personalized search for a software development organization by creating QUEST or domain based profile rather than individual user based profile.

**Keywords: Metasearch, Content Based Filtering, Query Bundle, QB-C, Quest Trail**

## 1. INTRODUCTION

Personalized search refers to search experiments that are tailored specifically to an individual's interests. It aims to resolve the ambiguity of query terms. To know more about the ambiguity that arises in search engines let us take the instance of "Java". When the user searches about Java there are three possibilities of results (i.e.) the results can be about Java Sea in Indonesia or about the Java coffee bean or the programming language. This is an example for ambiguity.

Difficulty in web searches has given rise to the need for development of personalised search engine. It is important to introduce personalization in a software organization where the employees are reluctant to provide information. There are two types of user behaviour (i.e.) search behaviour and browser behaviour. Search behavior [22] is everything the user enters in the search engine to search for the information needed. Browser behaviour involves surfing; user types a URL address in the browser, king a bookmark or forward page in the browser etc.

Searches can be analysed in three levels, (a) query level, (b) quest level and (c) session level. In query level it fails to capture the interleaving relationships between different quests [18]. If we analyse the search logs based on session (i.e. session level) [6] [11] [13][21] the quests will be interleaved. It is difficult to identify what the user is doing because the sessions are chronologically ordered.

If we analyse in quest level the topics will be more consistent and relevant to each other. This will help us to understand the intentions behind a user's search. Query is the search entry made by the user into the search engine (e.g.) the user types "jython versus swings" into the search box and searches. A Query Trail can be defined as sequence of user behaviour (a query followed by sequence of browsing behaviour) [14][16][28]. Quest (task) is an atomic information need (e.g.) the user needs to know what "jython" is? And compare its features with swings in "java". A quest trail represents all user activities within that particular task, such as query reformulations, URL clicks [17]. Sessionis defined as "a series of queries by a single user made within a small range of time" and the activities done by the user in that time period in a browser is known as session trail [23].

Consider the example shown in Table 1, which is a real user search session from Google (http://www.google.com). This session contains 4 different search quest: Twitter, Flipkart Kindle Books, Yahoo, and lyrics of a song. The "Yahoo" task is interleaved with the "Flipkart Kindle Books" task. The reasons causing the interleave phenomenon [18] are: (1) web search logs are ordered chronologically; (2) users often open several tabs or browsers and conduct multiple tasks at the same time.

**Table1: A sample session from web search logs.**

| Time | Event | Value | QUEST |
|---|---|---|---|
| 09:03:26 | Query | Twitter | 1 |
| 09:03:39 | Click | www.twitter.com | 1 |
| 09:06:34 | Query | Flipkart | 2 |
| 09:07:48 | Query | Twitter | 1 |
| 09:08:02 | Click | twitter.com/login.php | 1 |
| 09:10:23 | Query | flipkart kindle | 2 |
| 09:10:31 | Click | kindle.flipkart.com | 2 |
| 09:13:13 | Query | yahoo log in | 3 |
| 09:13:19 | Click | mail.yahoo.com/mail | 3 |
| 09:15:39 | Query | flipkart kindle books | 2 |
| 09:15:47 | Click | flipkart.com/Kindle-eBooks... | 2 |
| 09:15:59 | Click | astore.flipkart.com/ Flipkart.. | 2 |
| 09:17:51 | Query | You belong to me | 4 |
| 09:18:54 | Query | You belong to me lyrics | 4 |
| 09:19:28 | Query | Belong to me lyrics | 4 |

In this paper we bring in two studies semantic analysis and genetic algorithm for personalizing the search process in the search engine. To get a clear picture of our study we also discuss about Meta search engine, personalization and search.

## 1.1 Search Engine

A search engine is a type of computer software used to search data in the form of text or a database for specified information. Search engines normally consist of spiders (also known as bots) which roam the web searching for links and keywords. They send collected data back to the indexing software which categorizes and adds the links to databases with their related keywords. When you specify a search term the engine does not scan the whole web but extracts related links from the database. Search is the heart of the web. It is how we navigate the web. All the information available in the web will become inaccessible if we don't have a search engine to enter our queries.

Search is the means through which we discover information, access services, increase our store of knowledge, and broaden our horizon .Until recently we had to rely on Boolean search. A statistical and analytical technique that uses the operators AND, OR, NOT and NEAR to create a probability model of the answers to our search query. It relies on keywords. E.g. If our query has HELP and SEO, websites having these keywords will be given as answers to the query and also because the contents in the site have the keywords HELP and SEO that are strategically located. Boolean search does not work that simply. It relies on a lot of statistical data. The good thing is that search is changing. It is changing from Boolean search that provides the 10 best probable answers in response to search query which we then have to shortlist visiting each site to a more accurate computational type of search that is typified by search query like "How old is President Obama ?" which provides the correct answer right on the search page.

Search engines on the websites are enriched with facility to search the content stored on other sites. There is difference in the way various search engines work, but they all perform three basic tasks.

Finding and selecting full or partial content based on the keywords provided.

Maintaining index of the content and referencing to the location they find the information.

Allowing users to look for words or combinations of words found in that index.

Semantic Analysis

Semantic analysis is nothing but a process of filtering that progressively eliminates more and more input strings until you are left with only valid data. Semantic search is different from Boolean search as apples are different from oranges. The transition to semantic search also marks the transition on the web as we go from websites to people [7]. The web continues to be made of websites. In websites we get to find information, consume news and buy stuff. In order to understand natural language and search queries, it has to understand what these words really mean.

Metasearch Engine

Metasearch engine is a search tool that uses other search engine's data to produce their own results from the Internet. Metasearch engines take input from a user and simultaneously send out queries to third party search engines for results. Sufficient data is gathered, formatted by their ranks and presented to the users.

Information stored on the World Wide Web is constantly expanding, making it increasingly impossible for a single search engine to index the entire web for resources. Metasearch engine is a solution to overcome this limitation. By combining multiple results from different search engines, metasearch engine is able to enhance the user's experience for retrieving information, as less effort is required in order to access more materials. A metasearch engine is efficient as it is capable of generating a large volume of data, however, scores of websites stored on search engines are all different: this can draw in irrelevant documents. Other problems such as spamming also significantly reduce the accuracy of the search. This issue is tackled by the process of fusion which improves the engineering of metasearch engine. There are many types of metasearch engines available to allow users to access specialised information in a particular field. These include Savvy search engine and Meta seek engine. The advantage of using a metasearch engine is that by sending multiple queries to several other search engines this extends the search coverage of the topic and allows more information to be found. They use the indexes built by other search engines, aggregating and often post-processing results in unique ways. Metasearch engine has an advantage over a single search engine because more results can be retrieved with the same amount of exertion. It also reduces the work of users from having individual type searches from different engines to look for resources [5].

## 1.2 Genetic Algorithm

In the field of artificial intelligence, a genetic algorithm (GA) is a search heuristic that mimics the process of natural selection. This heuristic (also sometimes called a Meta heuristic) is routinely used to generate useful solutions to optimization and search problems. Genetic algorithms belong to the larger class of evolutionary algorithms (EA), which generate solutions to optimization problems using techniques inspired by natural evolution, such as inheritance, mutation, selection and crossover [24]. It is a very powerful and non-traditional optimization technique. It is based on the Darwinian Theory "Survival of the Fittest". Only the fittest will survive and reproduce and successive generations will become better and better compared to previous generations.

They are stochastic algorithms as they can come up with a different solution every time it is run on the same problem and not deterministic (an algorithm which gives the same answer for a given problem how many times it is run). GA's are creative algorithms in the sense that they make use of the concept of interaction of thousands of probabilities with each other and eventually come up with a solution. GA's are unique in that they operate from a rich database of many points simultaneously and can be incredibly efficient if programmed correctly.

The performance of the Genetic Algorithms for a particular problem can be made with regard to best fitness values obtained from it or the time taken to converge with the fairly optimal solutions because the problem might be time critical or it can also be measured in terms of diversity measures [2]. The performance can also be measured by the number of fitness functions evaluations done during the course of the run. For fixed population sizes the number of fitness function evaluations is given by the product of population size and the number of generations .The efficiency of GA varies from problem to problem and from generation to generation because some genes are solved during the first few generations but others take more time to do so, as the contribution of the genes of one individual towards the fitness function is not the same as some other genes in the same individual i.e. some genes is responsible for a high variance while others change the fitness value only minimally.

# 2. PROPOSED WORK

In our paper we mainly focus on personalizing the searching process for people working in a software development organization (analysts, developers, testers, maintenance team members), who find it increasingly difficult to get relevant results to their searches. We build group profiles based on either the domain in which software product is to be developed or on project basis [15].

Till now not much development is observed in web personalization field because individual web search behavior has not changed much. The main challenge in web personalization is to read the mind of the users [4]. This imposes a very big challenge because the words used for any search are limited to two or three words. Some of the issues in Web searching are (1) Structuring Queries i.e. the difficulty faced by users are properly structuring queries, namely applying the rules of a particular system, especially Boolean operators e.g., AND, OR, NOT and term modifiers e.g. '+', '!'. (2) Spelling i.e. the user tends to misspell their queries without even realizing it. (3) Query Refinement i.e. many times the users do not refine their query, even if there may be other terms that relate directly to their needed information. (4) Managing Results i.e. mostly, the user queries are extremely broad, resulting in an unmanageable number of results. Few users view more than the first ten or twenty documents from the result list.

## 2.1 Semantic Analysis

Analysis of the user's queries at a semantic level using vocabulary or ontology based system like ODP [8][29] or yahoo Directory [9] is semantic analysis. Optimal results from semantic analysis are chosen using genetic algorithm, where only the results that are most suitable to the users profile and interests are presented to the user [5]. Genetic algorithm aids with machine learning and supports the search engine to understand the user's mind while searching. Optimality of the results from semantic analysis is based on the user's profile that is built and the results of task analysis.

It helps in addressing the two most significant problems which is encountered during traditional content based filtering.

1. Cold start problem

2. Filter Bubble

### 2.1.1 Cold Start Problem

The lack of user rating leads to "cold start" problem. Initially when a user searches in a new domain he will not have the luxury of tracing recommended searches. Using semantic content based filtering and retrieving more semantically related concepts this problem can be solved.

### 2.1.2 Filter Bubble

Semantic analysis helps is overcoming the problem of over specialization. It means that the user is restricted to get recommendations which have strong resemblance to the one he already knows. This problem is also referred to as "Filter Bubble".

## 2.2 Quest Analysis

We define a quest to be the objective of the user's search (or) an atomic user information need (goal of a user's search), whereas a quest trail represents all user activities within that particular quest, such as query reformulations, URL clicks. Previously, Web search logs have been studied mainly at session[3] or query level where users may submit several queries within one quest and handle several quests within one session[6][26]. Quest level analysis of search log provides a better understanding of user's interests or goal, since it performs better in modelling user's profile. Thus the user behavior [22] can be studied and noted from the tasks he performs in the search engine. Thus task identification is important. We make use of the same task elicitation algorithm called Query Bundle - QUEST.

### 2.2.1 Bundling Queries into Quest

Some of the previous methods used for bundling queries into quest were WCC (weighted connected component), HTC (Head Tail component). In WCC an undirected graph for queries within a session was built. The vertices of the graph were queries and the edges were 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 query bundle. WCC outperformed other popular clustering algorithms like Query Flow Graph [1], K-means, and DB-Scan in bundling queries into quest, as indicated in [26]. WCC was found to be better than any other previous bundling algorithms because, every query was compared with every other query before bundling queries into Quests. But the time complexity of WCC is O(k·N2), where N is the average number of queries of a session and k is the dimension of features. The overall time complexity in intolerable for search logs of massive volume.

To overcome this Orlando .S. [26] proposed another head-tail component query clustering approach (HTC) to reduce the time complexity. In this approach only the similarity between head tail components were considered for bundling queries into quest. This fails to address cases of interleaved quests. We are proposing a new approach that could reduce the time complexity while addressing the interleaving quests. We name this algorithm as QB-Q.

This algorithm bundles queries belonging to a quest or relevant quest. Say for instance there are 4 queries in a search log A, B, C, and D. WCC would have needed 6 pairs of relevance computation [4] [12] [19][25], whereas our proposed method will lesser number of relevance computation unless it is a worst case where every query is irrelevant to every other query in the search log. To be more precise if A is similar to B and B is similar to C, there is no need to compute the relevance between A and C any more. If A is similar to B but B is not similar to C, QB-Q still has to compute the relevance between A and B to avoid the quest interleaving.

QB-Q is efficient approach to bundle queries of related quests or same quest. Extracting such information regarding the user's objective of searching helps us to model a stronger and dynamic user profile. Thus we can develop a more accurate profile to reflect the user's requirements than the ones that are statically created at the time of registration.

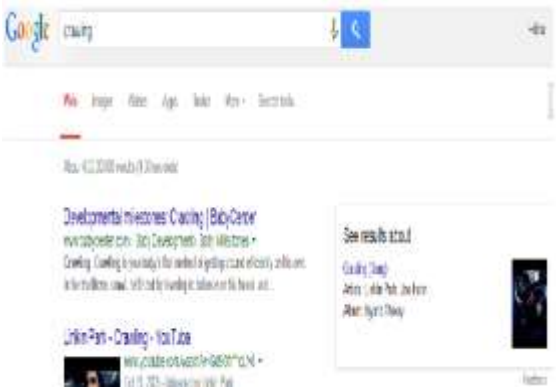
**Algorithm 1: Query Bundle – QUEST**

**Input**: Query set *Q*, cut-off threshold *b*;
**Output**: A set of Quest q;
**Initialization**: q= *null* ;
Query to Quest table *L [ ] =null* ;
1: **for** *len* = 1 **to** |*Q*| − 1 **do**
2: **for** *i* = 1 **to** |*Q*| − *len* **do**
3: **if** *L* [*Qi*] != *L* [*Qi+len*] **then** // if two queries are not in the same quest
4: *s* ← compare (*L* [*Qi*], *L* [*Qi+len*]); // compute similarity takes *O*(*k*)
5: **if** *s* ≥ *b* **then**
6: merge q (*Qi*) and q (*Qi+len*);
7: modify *L*;
8: **if** |q| = 1 break; // break if there is only one task
9: return q;

Let us now see a comparison of search result from our proposed work and a search result from Google. Assume the user is working in a cloud computing domain and he issues a search for the word "crawling". The same word will have different meaning in different context. That is the reason for the different results observed when the same word is searched in Google and our personalized metasearch engine "QUEST TRAIL".

In Google we observe results regarding, a baby's first movements, a Linken Park's song, and insects crawling and so on. The results do not seen to match the user's quest. Whereas the results of "QUEST TRAIL" are all relevant to the user's domain and hence is much more relevant to the user's quest.


/settings/ads/preferences%3Fhl%3Den
http://link.springer.com/chapter/10.1007%252F978-3-642-35864-7_53
http://link.springer.com/content/pdf/10.1007%252F978-3-642-35864-7_53.pdf
http://yourstory.com/2012/05/promptcloud-data-crawling-and-cloud-computing-solutions/
http://www.researchgate.net/publication/216965265_Service_Crawling_in_Cloud_Computing
http://www.connotate.com/4-strategies-everyone-web-crawling-industry-using-6414
https://www.promptcloud.com/
http://www.slideshare.net/ideseditor/an-efficient-cloud-based-approach-for-service-crawling



## 3. OTHER TECHNIQUES

Some of the other techniques that are currently being used for personalization are briefly listed below. Query rewriting, semantic content filtering, re-ranking [10], semantic celebrative filtering, user modelling or profiling and analysis of search logs are techniques that are used to improve the relevancy in the search results for the user, while reducing their effort[20].

### 3.1 Query Reformulation

Here the query is elaborated by the user to personalize the search result [30]. For example the query is to find the Thai restaurants located in the city of Chennai. Here in this technique Chennai is added to the search query and taken as "Thai restaurants Chennai "and the search results are given for this query. The search engine will now give the results for Thai restaurants that are in Chennai. The problem over here is there are chances where the user may not be clear about the location.

Let us say for example, a website organizer might use a word which he likes most but an individual looking for the same information might not go for the same word, instead he might use its synonym. Then tracing of such web pages will be quite a challenge for the search engines. Synonyms refer to many words expressing same meaning and poli-semis refer to one word with different meanings. Owing to this kind of language richness and the context sensitive sense a word assumes, the keyword method used by search engines faces quite a lot of issues. A user seeking for information is expected to keep reframing their query until it matches the form that is expected by the search engine.

### 3.2 URL Re-Ranking

Re-Ranking[10] the results for a user based on his profile is one of the conventional approaches for personalization. Page re-ranking is used mainly to take the advantage of user's profile. Initially some 'n' documents are taken that are reordered as per the preference from the user profile [22]. The re-ranking occurs by scores assigned to each SERP that checks with the user profile.

### 3.3 User Modeling in Personalized Systems

Traditional methods for modelling user profile were focused on creating a single static profile at the time of registration. It does not address the problem of different queries being needed to be handled differently. Collecting the user preferences and choices of the user at the time of registration helps in predicting the needs of the user.

### 3.4 Google's Approach

Google is taking several steps to improve the search results that are provided to the user. It provides the user with personalized results if the preferences are given by the user initially. For this enhancement it is required by the user to create a profile. The user is required to give the details of his preferences and based on it Google retrieves the personalized results to the user and also updates the user of any new information through mail.

## 4. CONCLUSION

In a software development organization there is a special need for quest – specific or domain – specific ranking. Applying QUEST level analysis of the search log for constructing a personalized search engine for software developers will make the searching process much easier. Our system proved an effective performance in personalizing the searching process especially for software developers. It is clearly seen that the combination of quest analysis and semantic analysis is an effective approach for personalization of searching process and it is better than any of the currently used techniques for personalization.

## 5. REFERENCES

[1] Boldi, P., Bonchi, F., Castillo, C., Donato, D., Gionis, A. And Vigna, S., "The query-flow graph: model and applications," ser. CIKM '08, 2008, pp. 609–618.

[2] Radlinski, F. and Craswell, N., "Comparing the sensitivity of information retrieval metrics," ser. SIGIR '10, 2010, pp. 667–674.

[3] Catledge, L.D. and Pitkow, J.E., "Characterizing browsing strategies in the world-wide web," Computer Networks and ISDN Systems, vol. 27, no. 6, pp. 1065–1073, 1995.

[4] Huang, C.K., Chien, L.F. and Oyang, Y.J., "Relevant term suggestion in interactive web search based on contextual information in query session logs," Journal of the American Society for Information Science and Technology, 2003.

[5] White, R., Bailey, P. and Chen, L., "Predicting user interests from contextual information," ser. SIGIR '09, 2009, pp. 363– 370

[6] Cao, H., Jiang, D., Pei, J., He, Q., Liao, Z., Chen, E. and Li, H., "Context-aware query suggestion by mining click-through and session data," in KDD '08, 2008, pp. 875–883.

[7] Song, Y., Zhou, D., and He, L.-w., "Query suggestion by constructing term-transition graphs," ser. WSDM '12, 2012, pp. 353–362.

[8] 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.

[9] Donato, D., Bonchi, F., Chi, T. and Maarek, Y., "Do you want to take notes? identifying research missions in yahoo! Search pad," ser. WWW '10, 2010, pp. 321–330.

[10] 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.

[11] Jain, A., Ozertem, U. and Velipasaoglu, E., "Synthesizing high utility suggestions for rare web search queries," ser. SIGIR '11, 2011, pp. 805–814.

[12] Craswell, N. and Szummer, M., "Random walks on the click graph," ser. SIGIR '07, 2007, pp. 239–246.

[13] Jones, R. and Klinkner, K.L., "Beyond the session timeout: automatic hierarchical segmentation of search topics in query logs," ser. CIKM '08, 2008, pp. 699–708.

[14] Liu, Y., Gao, B., Liu, T.-Y., Zhang, Y., Ma, Z., He, S. and Li, H., "Browserank: letting web users vote for page importance," ser. SIGIR '08, 2008, pp. 451–458.

[15] Beeferman, D. and Berger, A., "Agglomerative clustering of a search engine query log," ser. KDD '00, New York, NY, USA,2000, pp. 407–416.

[16] 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.

[17] Fox, S., Karnawat, K., Mydland, M., Dumais, S. and White, T., "Evaluating implicit measures to improve web search," ACM Trans. Inf. Syst., vol. 23, pp. 147–168, 2005.

[18] Chapelle O., Joachims T., Radlinski F. and Yisong Yue, "Largescale validation and analysis of interleaved search evaluation," ACM Trans. Inf. Syst., vol. 30, no. 1, p. 6, 2012.

[19] Gao, J., Yuan, W., Li, X., Deng, K. and Nie, J.-Y., "Smoothing clickthrough data for web search ranking," ser. SIGIR '09. ACM, 2009, pp. 355–362.

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

[21] Silverstein, C., Henzinger, M.R., Marais, H. and Moricz, M., "Analysis of a very large web search engine query log," SIGIR Forum, vol. 33, pp. 6–12, 1999.

[22] Hassan, A., Jones, R., and Klinkner, K., "Beyond dcg: user behavior as a predictor of a successful search," ser. WSDM '10, 2010, pp. 221–230.

[23] Jansen, B., Spink, A. and Kathuria, V., "How to define searching sessions on web search engines," ser. WebKDD '06, 2007, pp. 92–109.

[24] 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.

[25] Jones, R., Rey, B., Madani, O. and Greiner, W., "Generating query substitutions," ser. WWW '06. ACM, 2006, pp. 387–396.

[26] 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.

[27] Hassan, A., Song, Y. and He, L.-w., "A task level user satisfaction metric and its application on improving relevance estimation," ser. CIKM '11, 2011.

[28] Olston, C. and Chi, E.H., "Scenttrails: Integrating browsing and searching on the web," ACM Trans. Comput.-Hum. Interact., vol. 10, pp. 177–197, September 2003.

[29] Shen, X., Tan, B. and Zhai, ChengXiang, "Context-sensitive information retrieval using implicit feedback," ser. SIGIR '05, 2005, pp. 43–50.

[30] Mei, Q. and Zhou, D. and Church, K., "Query suggestion using hitting time," ser. CIKM '08. ACM, 2008, pp. 469–478.

[31] IEEE TRANSACTIONS O,NV OKLN.O26W, LNEOD.G2E, FAENBDR EUNAGRINYE 2E0R1I4NG,VOL.26, NO.12, APRIL 2014 "Task Trail: An Effective Segmentation of user search behaviour" Zhen Liao, Yang Song, Yalou Huang, Li-wei He, Qi He.