Online Behavioral Genome Sequencing from Usage Logs: Decoding the Search Behaviors

Yang Song, Weiwei Cui, Shixia Liu, Kuansan Wang
Microsoft Research
{yangsong,weiweicu,shliu,kuansanw}@microsoft.com

ABSTRACT

We present a system to analyze user interests by analyzing their online behaviors from large-scale usage logs. We surmise that user interests can be characterized by a large collection of features we call the *behavioral genes* that can be deduced from both their explicit and implicit online behaviors. It is the goal of this research to sequence the entire behavioral genome for online population, namely, to identify the pertinent behavioral genes and uncover their relationships in explaining and predicting user behaviors, so that high quality user profiles can be created and the online services can be better customized using these profiles. Within the scope of this paper, we demonstrate the work using the partial genome derived from web search logs. Our demo system is supported by an open access web service we are releasing and sharing with the research community. The main functions of the web service are: (1) calculating query similarities based on their lexical, temporal and semantic scores, (2) clustering a group of user queries into tasks with the same search and browse intent, and (3) inferring user topical interests by providing a probability distribution over a search taxonomy.

Categories and Subject Descriptors

H.3.4 [**Information Storage and Retrieval**]: User Profiles and Alert Services

Keywords

Search task identification, user behavior modeling, personalization

1. INTRODUCTION

Human-beings are characterized by our biological genes which determine how proteins are synthesized in our bodies and eventually lead to characterize our physical features. We are testing an analogous hypothesis that user behaviors are governed by their innate interests that can be encoded by what we call the *behavioral genes* that partially manifest themselves as patterns in online behaviors, e.g., websites visited, online shopping history, search queries, etc. Our research aims at sequencing the entire behavioral genome for Internet users so that online services can be better customized to fit their user profiles. In this work, we leverage a portion of the user online activity data from search engine logs to deduce user interests and demonstrate our research in this area.

User search logs have become the primary source for search engines to better understand user search interests and therefore improve search relevance. Traditionally, search logs are commonly organized based on the timestamps of user search activities such as submitting a query or clicking a document on result page. Moreover, search logs are usually divided into sessions based on some pre-defined inactivity threshold (e.g., 30-minute inactivity). Recently, given rise to arguments among researchers whether such representation of user search logs might not be ideal to capture the genuine user information need. In particular, our previous research has indicated that (1) users often perform multi-tasking simultaneously when using search engine [4] (e.g., checking on facebook feed while planning a trip), and (2) a lot of user information need actually spans multiple sessions, or even multiple days [6, 3]. Consequently, researchers have urged a new definition of *search tasks* to overcome the drawbacks of the previous session definition, hoping to better understand user search interests.

Our previous research on search task identification has demonstrated the power of task, which can capture user information need more precisely than session [4, 6, 5]. It has also shown to be very effective on improving search relevance [2], as well as the applications on query suggestions [4], user behavior analysis [3] and personalization [7].

The objective of this demo is to unify our previous effort on task identification and user interests modeling, by providing several web services that takes user queries as input and output (1) identified user search tasks with semantic information, and (2) user topical interests. By doing so, we hope to *decipher* user search genome, provide the community with an open platform of resources with models learnt from large corpus of commercial search engines, as well as to trigger a new horizon of research effort towards this direction.

2. DESCRIPTION OF OUR SERVICES AND DEMO

We provide three services that correspond to three most important stages of user preference modeling, so that re-

WWW'14 Companion, April 7–11, 2014, Seoul, Korea.
ACM 978-1-4503-2745-9/14/04.
http://dx.doi.org/10.1145/2567948.2577015.

searchers who use our services can have full control of the intermediate results and the final outcome. To be specific, our three services are:

1. QSIM service. Compute a similarity score given any two query strings and (optionally) the time-gap between them.
2. QCluster service. Cluster a set of queries into tasks based on user search intent. This includes both within and cross session task segmentation models.
3. TopicInf service. Inference user topic interests based on user search tasks. The prediction is a probability distribution over a pre-defined taxonomy.

We briefly explain how our services work before presenting the technical details. In Figure 1, we present a real-user example mined from search logs. In this example, the user issued a total of 8 queries in three different topics (determined by our model). In the first Query Similarity Service, each pair of queries is assigned with a similarity score between 0 and 1. A strongly connected graph is then formed where edges exist between any pairs of queries. After that, the Query Clustering Service is used to split the graph into individual components called tasks by dropping edges that are below a certain threshold. In this example, we can see that queries about NFL sports teams are grouped into one task, queries about reading devices are grouped into another task, while the query "facebook" are left as a singleton task. To predict the user search interests, the Topic Inference Service is then utilized to estimate user topical distribution on a set of categories. In this example, the user is clearly mostly interested in Sports (0.59), followed by Computers (0.21) and Shopping (0.13).

3. TECHNICAL DETAILS OF OUR SERVICES

3.1 QSIM Service

The goal of this service is to compute a similarity score given two query strings and (optionally) the time-gap between them. Our model is a modified version of our previous WWW paper [4]. Specifically, a logistic regression model was trained using three sets of features: temporal features, word features and semantic features. In particular, the logistic regression model outputs a score between [0, 1] for each query pair

$$S(x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}}, \tag{1}$$

where β_0 is the bias term and β_1 are coefficients for the feature vector x that contains the features specified in Table 1. The model was trained on 17,924 query pairs, annotated by 26 human judgers. For the details of the model, readers are referred to [4].

Note that the model used in [4] only considered lexical similarities from queries. To add semantic meanings to queries so that those queries that do not share common words but indeed referred to the same user tasks, we leverage a large knowledge base from Microsoft Research Asia named Probase[1] [8]. Probase contains knowledge which is extracted from a large web corpus of billions of web documents. It defines over two million concepts (comparing to Freebase that has 1,450 concepts) with entities and attribute information. For example, query "amazon" is in the classes of *company*, *technology company* and etc, with the attributes including *location*, *hompage* and etc.

[1]http://research.microsoft.com/en-us/projects/probase/

Feature Description
Word Features
lv_1: Levenshtein distance of two queries
lv_2: lv_1 after removing stop-words
prec_1: average rate of common terms
prec_2: prec_1 after removing stop words
prec_3: prec_1 (If term A contains B, A=B)
rate_s: rate of common characters from left
rate_e: rate of common characters from right
rate_l: rate of longest common substring
b_1: 1 if one query contains the other, else 0
Temporal Features
timediff_1: time difference in seconds
timediff_2: category for 1/5/10/30 mins
Semantic Features
Probase_sim: category similarity from Probase

Table 1: List of features used in QSIM service.

For query understanding, we first segment queries into the longest possible entity names, and then calculate their category similarity. Take two queries "Peyton Manning" and "Chris Johnson Injury" from Figure 1 for example. The longest possible entity name for the first query is itself, while for the second query it is "Chris Johnson". We then pass these two entities to the function `GetClassByInstance` provided by `ProbaseAPI` and get the following categories and scores,

`PossibleInstanceName:`	`Peyton Manning`
`player`	0.143836
`athlete`	0.109589
`star`	0.027397
`nfl player`	0.020548
`great athlete`	0.013699
.....	
`PossibleInstanceName:`	`Chris Johnson`
`player`	0.347826
`athlete`	0.043478
`nfl standout`	0.043478
`real estate master`	0.043478
`running back`	0.043478
.....	

(2)

For space concern, we only list top 5 categories for each entity. In practice, we leverage top-50 categories to increase the coverage. Finally, to compute the similarity between two entities, we use Jaccard similarity that takes the score as the weight for each common category:

$$J(C(e_1), C(e_2)) = \frac{|C(e_1) \cap C(e_2)|}{|C(e_1) \cup C(e_2)|}. \tag{3}$$

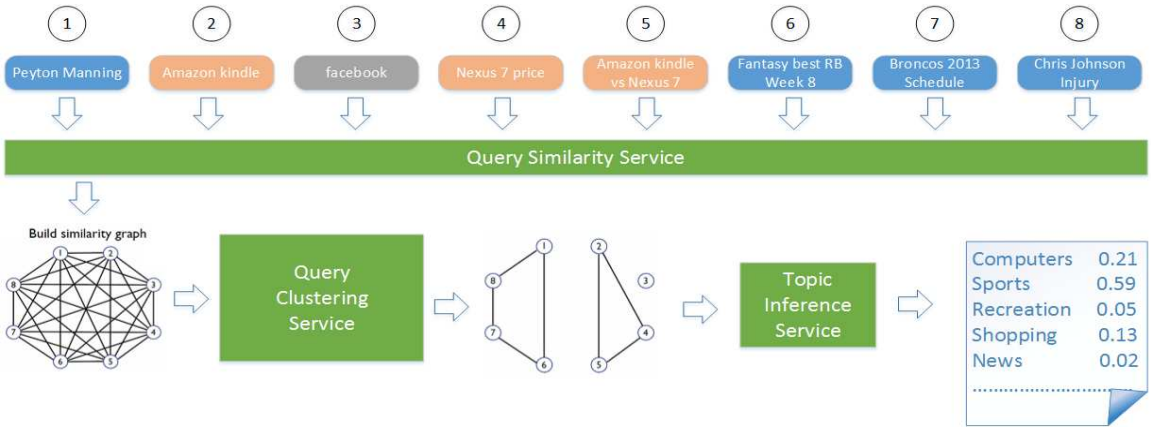

Figure 1: An example of our services to decipher user search genome. QSim service calculates pairwise query similarity for each user session. QCluster service groups semantically related queries into tasks. TopicInf service infers user interests from search tasks.

3.2 QCluster Service

The goal of this service is to cluster a set of queries into tasks based on user search intent. The method works in two steps: first, measure the similarities between query pairs (by using the QSIM service mentioned above); second, cluster queries into tasks based on their similarity scores. Specifically, given a set of queries, the model will determine the optimal number of clusters, and assign each query into one of the clusters. Each cluster has a unique cluster ID. The cluster ID starts from 1 and can be any arbitrary positive integer depending on the size of the data set.

In particular, the cluster method QTC [4] works as follows: using the learned query similarity function, QTC then builds an undirected graph of queries for each user's search history, where the vertices of the graph are queries and the edges represent similarities between queries. By dropping the weak edges where the similarities are smaller than a threshold which is determined using cross validation, the algorithm extract all connected components of the graph as tasks.

Note that our model deals with cases where tasks can interleave with each other. Again, take Figure 1 for example, the 8 queries are clustered into three tasks: task 1 regarding sports, task 2 regarding computer hardware and task 3 containing a single query.

3.3 TopicInf Service

Profiling users search behavior and interests into a predefined categories has shown to be very useful for personalized search [7]. Thus, our TopicInf Service aims at providing such user profiling service. Since the Probase knowledge base contains over 2 million concepts, profiling users in such big taxonomy is quite unrealistic. Therefore, we seek a much more compact representation of user interests taxonomy and decide to use the ODP categories[2]. The Open Directory Project (ODP) is a web document directory, which is maintained by a group of volunteer editors. ODP leverages a tree hierarchy to display web link categories where top-level categories are more abstract than lower-level ones. For example, the web link *http://microsoft.com* is classified under *Computers/Companies*. In our work, we leverage the top-2 levels of ODP tree which consists of 221 categories to profile user search interests. We start from a search task for a particular user. To profiling each task of the user, we use a content-based ODP classifier [1] that takes a URL as input and outputs its topical labels. For the demo purpose, we do not require user clicks to be part of the input. Thus, with the absence of user clicks, we use the idea of *pseudo relevance feedback* by assuming that the top-N returned documents are relevant to the user query. Specifically, for each query in the task, we set $N = 5$ to take the top-5 returned documents and aggregate their topical labels to represent the topical distribution for that query. We then aggregate the query topics for each task by normalizing them to be a distribution. Similarly, each user's topical interests is determined by aggregating the topics from all his/her search tasks.

Take Figure 1 for example. By aggregating the topic distribution of the three tasks, the user's favorite topics are *Sports*, followed by *Computers* and *Shopping*.

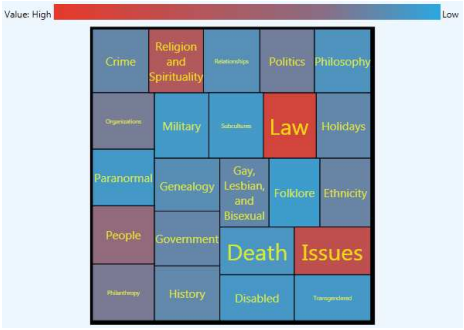

Figure 2: TreeView of the second-level topics after zooming-in the tree view box.

4. THE DEMO SYSTEM

Our demo system[3] showcases the three services we mentioned in the previous section. The system contains mainly three components: (1) a sliverlight-based front-end user

[2]http://dmoz.org

[3]Demo video available at https://vimeo.com/82216211.

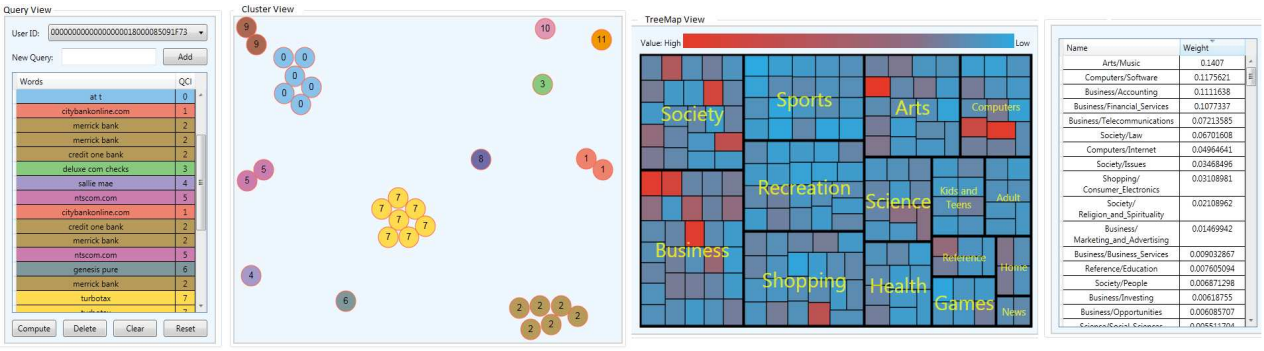

Figure 3: The main front-end UI of our demo system. It contains three major components: a query view box, a cluster view box and a treemap view box.

interface, (2) a back-end web service that processes all requests, and (3) a back-end http server that holds Probase database and topic inference classifier into the memory. The front-end UI only communicates with the back-end web service, which responses to the request by reading modules and data from the http server.

Figure 3 shows the front-end UI of our demo system. It is composed of three major components: a query view (QV) box, a cluster view (CV) box, and a treemap view (TV) box. The demo works by first filling in queries in the QV box, by either choosing a random search user from our database, or typing arbitrary queries into the QV box. Then by clicking the *action* button, the system shows the clustering results in the CV box, highlighting each cluster with different colors. Meanwhile, queries in the QV box are also colored accordingly to indicate their corresponding tasks (clusters).

On the other hand, the TV box illustrates a two-dimensional hierarchical topic distribution of the current user. It uses nested rectangles to illustrate the tree structure and relative sizes of nodes. Recall that we have a two-level hierarchy from ODP. The topics from the first level is shown in different sized rectangles based on their number of second-level topics. The colors in each rectangle demonstrates the popularity of that topic, where more popular topics have colors that more towards red. Right beside the TV box, we also depicts the most popular topics as well as their probabilities. A left-click on any rectangles in TV box will zoom-in the view into the second-level topics of that rectangle, as shown in Figure 2. Similarly, a right-click will zoom-out the current view to show the first-level topics.

The system also provides rich interactions to allow users to examine the clustering results from multiple perspectives. For example, right-clicking on any query in the QV box will bring up the topic distribution of that particular query in the TV box. From the existing query list, users can choose to add/remove queries and re-estimate the topic distribution and query clusters by clicking the *Compute* button. Likewise, users can also select one or more clusters (user tasks) in the CV box and re-plot the topics in TV box. From the CV box, users can left-click to choose a cluster/task and the corresponding queries in the left QV box will also be highlighted. The *Reset* button in QV box provides a remedy to undo any changes to the current query list. Note that some of these operations require real-time computation with communication to the back-end server, which may cause some delay in the UI response.

5. CONCLUSIONS

We presented a demo to help analyzing and understanding user search and browse behavior by mining user search logs from search engines. The demo showcased features including calculating query similarity, clustering queries into similar search tasks, and modeling user topical interests in ODP categories. We hope to leverage this demo to advocate the research area of deciphering user behavior genome from the entire user online activities, while in this paper a portion of search activities were used to represent the entire behavioral genome. We will also provide an open access to our web services in this paper via APIs, which will be available soon.

6. REFERENCES

[1] P. N. Bennett and N. Nguyen. Refined experts: Improving classification in large taxonomies. In *SIGIR '09*, pages 11–18, New York, NY, USA, 2009. ACM.

[2] A. Hassan, Y. Song, and L.-w. He. A task level metric for measuring web search satisfaction and its application on improving relevance estimation. In *CIKM '11*, pages 125–134, New York, NY, USA, 2011.

[3] A. Kotov, P. N. Bennett, R. W. White, S. T. Dumais, and J. Teevan. Modeling and analysis of cross-session search tasks. In *SIGIR '11*, pages 5–14, 2011.

[4] Z. Liao, Y. Song, L.-w. He, and Y. Huang. Evaluating the effectiveness of search task trails. In *WWW '12*, pages 489–498, New York, NY, USA, 2012. ACM.

[5] C. Lucchese, S. Orlando, R. Perego, F. Silvestri, and G. Tolomei. Identifying task-based sessions in search engine query logs. In *WSDM '11*, pages 277–286, New York, NY, USA, 2011. ACM.

[6] H. Wang, Y. Song, M.-W. Chang, X. He, R. W. White, and W. Chu. Learning to extract cross-session search tasks. In *WWW '13*, pages 1353–1364, Republic and Canton of Geneva, Switzerland, 2013. ACM.

[7] R. W. White, W. Chu, A. Hassan, X. He, Y. Song, and H. Wang. Enhancing personalized search by mining and modeling task behavior. In *WWW '13*, pages 1411–1420. ACM, 2013.

[8] W. Wu, H. Li, H. Wang, and K. Q. Zhu. Probase: A probabilistic taxonomy for text understanding. In *SIGMOD '12*, SIGMOD '12, pages 481–492, New York, NY, USA, 2012. ACM.