Enabling Direct Interest-Aware Audience Selection

Ariel Fuxman
Microsoft Research
Mountain View, CA
arielf@microsoft.com

Anitha Kannan
Microsoft Research
Mountain View, CA
ankannan@microsoft.com

Zhenhui Li[*]
University of Illinois
Urbana-Champaign, Illinois
zli28@uiuc.edu

Panayiotis Tsaparas[†]
University of Ioannina
Ioannina, Greece
tsap@cs.uoi.gr

ABSTRACT

Advertisers typically have a fairly accurate idea of the interests of their target audience. However, today's online advertising systems are unable to leverage this information. The reasons are two-fold. First, there is no agreed upon vocabulary of interests for advertisers and advertising systems to communicate. More importantly, advertising systems lack a mechanism for mapping users to the interest vocabulary.

In this paper, we tackle both problems. We present a system for direct interest-aware audience selection. This system takes the query histories of search engine users as input, extracts their interests, and describes them with interpretable labels. The labels are not drawn from a predefined taxonomy, but rather dynamically generated from the query histories, and are thus easy for the advertisers to interpret and use. In addition, the system enables seamless addition of interest labels provided by the advertiser.

The proposed system runs at scale on millions of users and hundreds of millions of queries. Our experimental evaluation shows that our approach leads to a significant increase of over 50% in the probability that a user will click on an ad related to a given interest.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Clustering*; H.2.8 [**Database Management**]: Database Applications—*Data mining*

General Terms

Algorithms, Experimentation

Keywords

Online Targeted Advertising, Query-Log Clustering

1. INTRODUCTION

In online advertising, advertisers want to target a specific audience that is more likely to engage with their campaign. Typically, advertisers are capable of describing this audience fairly accurately. However, in today's online advertising systems, they do not have the option to *explicitly* specify the characteristics of the users that they wish to target, except for broad demographic information. Instead, they bid on query terms, which act as a proxy for the user interests. But queries can be misleading when taken out of context. For example, if a user queries for "helmets", it is not obvious if she is looking for bike helmets or motorcycle helmets. Ads for both will appear, since this is a term related to both bike and motorcycle helmet companies. If we knew that the user who posed the query has a long-term interest in biking, then it would become clear that the query is more likely to be about bike helmets. The techniques that we present in this paper allow the bike helmet advertiser to *directly* specify that she prefers users who are interested in biking and the advertising system to identify the users who are interested in biking. Thus, a match between users and advertisers with intersecting interests can be easily made. We call this capability *direct interest-aware audience selection*.

Enabling advertisers to directly specify user interests is extremely powerful. For instance, part of the appeal of advertising on social media sites such as Facebook is the ability for advertisers to directly select their audience based on their expressed interests, as well as their "likes" and friends[1]. An expensive restaurant can select users who have specified interest in "Food and Wine", while a company that sells outdoors equipment can advertise to users who have declared interest in "Camping". This option is not available when advertising on search engines. The aforementioned restaurant would have to guess the terms with which a user will express their interest, and bid on these terms. In the case of "Food and Wine", this translates to a large set of terms related to restaurants, fine dining, wine selection, entertainment arrangements, etc. This places a huge burden on the advertisers to come up with the right terms, and they still run the risk of triggering incorrectly, or missing an important term.

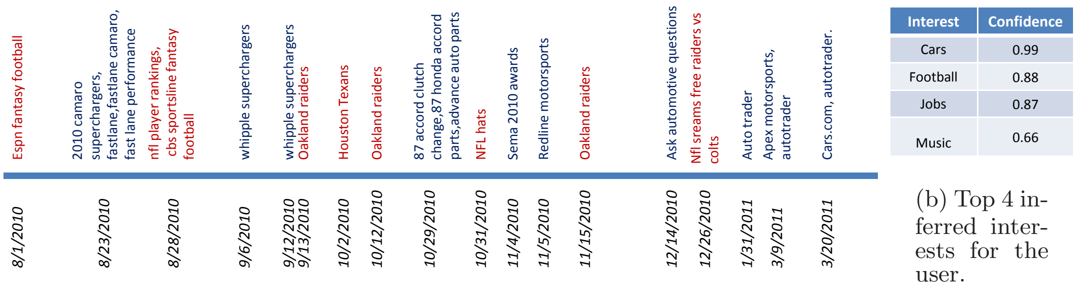
Unlike social media users, search engine users do not explicitly state their interests and preferences. However, they give abundant *implicit* information about their interests

[*] Work done while the author was an intern at Microsoft Research.

[†] Work done while the author was at Microsoft Research.

[1] See https://www.facebook.com/business#!/business/ads/.

CIKM'12, October 29–November 2, 2012, Maui, HI, USA.
Copyright 2012 ACM 978-1-4503-1156-4/12/10 ...$15.00.



(a) A portion of the user history corresponding to queries for *only* two of the identified clusters, "football" and "cars". The clusters spread over a long period of time. Queries of the clusters are semantically related but do not necessarily share terms. For instance, the "football" cluster has queries "oakland raiders" and "houston texans" with no overlapping terms. Similarly, the "cars" cluster has queries including "Sema 2010 awards" and "redline motorsports". For best visualization, please see this figure in color.

Interest	Confidence
Cars	0.99
Football	0.88
Jobs	0.87
Music	0.66

(b) Top 4 inferred interests for the user.

Figure 1: Example of inference of user interests

furniture	travel	movies	basketball	games	fishing	coupons	wedding
airlines	dining	diet	food	recipes	hotels	cars	games
cruises	baby	chiropractic	music	health	lyrics	baking	commodities
brewers	divorce	timeshare	cycling	arthritis	roofing	orthopedic	flu

Table 1: A subset of interest labels identified using our approach

through their actions, and more specifically their queries. Users query about anything and everything that is on their mind. Compiling the long-term (*e.g.,* year-long) query history of a user reveals a variety of interests: ephemeral interests that correspond to short-term tasks such as buying a new washing machine; routine interests that correspond to queries that enable everyday tasks such as reading a newspaper or checking email; activities that correspond to long-term interests of the user such as diet, sports, gaming, and health care.

In this paper, we consider the problem of extracting user interests from query histories for enabling *direct* interest-aware audience selection. Given a collection of multiple user histories, we will produce a set of *interest labels*, and train a model that assigns interest labels to users. The labels are not drawn from a predefined taxonomy, but rather dynamically generated from the query histories, and thus easy for the advertisers to interpret and use for targeting specific users. A direct interest-aware audience selection capability is important for both sponsored search and display advertising. In the former, the targeted user interests would be provided by the advertisers alongside the usual bid terms; in the latter the interests would be specified together with demographic and other behavioral targeting information.

We contrast our approach to other audience selection approaches, where users may be associated to interests, but these interests cannot be directly used by the advertisers. For example in the work of Ahmed et al. [1] interests are represented by topics produced by a topic model, and used for ad click prediction. Advertisers do not have the option to directly specify the interests they want to target. Furthermore, the produced interests are not easily interpretable, and thus they cannot be used for direct targeting.

In a nutshell, our approach is as follows: First, we cluster the query history of each individual user in order to identify groups of queries that are about the same interest. For the clustering, we use a measure of semantic similarity between terms that we obtain by exploiting temporal relationships between queries. Figure 1(a) shows an illustrative example of two clusters obtained using our approach from a user history in our data set. Notice, for instance, the queries "oakland raiders" and "houston texans" being clustered together but having no terms in common.

Given the clustering of the query history, we extract a short description for each cluster consisting of the most popular query terms present in the cluster. Then, we generate the interest labels by finding terms that occur frequently across multiple user histories, and selecting a subset of these terms as our interest vocabulary. Table 1 lists some of the terms that our approach extracted as part of the interest vocabulary. We can see that the interests are represented using commonly used vocabulary found in search queries. In order to map the clusters into this vocabulary, we train a classifier using massive amounts of *automatically* created training data constructed from the queries in the labeled clusters. The classifier can then be applied to new users to map them to the set of interest labels. For the same user shown in Figure 1(a), Figure 1(b) shows the top four interests inferred by our approach.

Our contributions include the following:

- We address the problem of direct interest-aware audience selection. Our approach distinguishes itself from previous work on learning interests [1] by the fact that users are assigned a concise set of interpretable interest labels, empowering the advertisers to directly target users using these labels.

- At the core of our approach is a component for clustering queries within a user history that are thematically related. Our clustering algorithm uses a novel similarity measure, which makes use of the semantic relationships between terms defined by the temporal co-occurrence of queries across multiple user histories. Thus, our clustering approach exploits both the local (within a single user history) and global (across user histories) relationships between queries for deriving query clusters.

- We implement our approach on a distributed data storage and processing system. Our system runs at scale for millions of users and hundreds of millions of queries. We perform a thorough experimental evaluation that shows that our approach leads to a significant increase of over 50% in the probability that a user will click on an ad related to a given interest. The evaluation was performed at a large scale, on 150,000 users using 2 months of ad data and user histories consisting of 16 months of query activity.

We note that although in this work we consider the problem of audience selection, our work can also be applied to other tasks, such as personalization of search user experience. In this case, the user interests could be used to provide context for a query, and tailor the search engine response to the needs of the specific user.

The rest of the paper is structured as follows. In Section 2, we present related work. In Section 3, we provide an overview of our approach. In Section 4, we present the details of the modeling phase of our approach. In Section 5, we present experimental results. Finally, in Section 6, we make concluding remarks and give directions for future work.

2. RELATED WORK

Computational advertising is an emerging research field that considers the application of computational and algorithmic techniques to online advertising. We refer the reader to the course notes of *Introduction to Computational Advertising*[2] for a thorough review of the field. Behavioral targeting, the use of prior user history for improving the effectiveness of an online campaign, is a prominent research topic within this field and has received considerable attention [1, 5, 9, 14, 21, 22]. Pandey et at. [14], Chen et al. [5], and Yan et al. [22] model the user as a bag of events, such as clicks to pages or queries. Jaworska et al. [9] represent users as a vector of categories, by mapping their web page visits to a predefined taxonomy. A machine-learning model is then trained in order to predict whether a user will click on an ad. Tyler et al. [21] model the problem of audience selection as an information retrieval problem, where there is a repository of users, and some users that are known to respond well to a campaign are used as queries over the repository. Users are modeled again as a bag of events, queries, and web page clicks. Recent work [12, 17, 2] has also considered the use of social network information (friendships, email communication) for improving behavioral targeting. The scalability of the behavioral targeting problem has been addressed either with Map-Reduce implementations [5, 14] or sampling [1].

The closest work to our approach is the recent work by Ahmed et al. [1]. They consider the problem of behavioral targeting in display advertising, and use a generative topic model to define interests over histories of multiple users. Then, they use the interests of users who have clicked on an ad as features in a classifier that predicts whether a user will click on the ad. Their technique assumes the existence of previous ad click activity for the given ad. Furthermore, their interest topics are not directly used by the advertisers. In contrast, we associate users to a concise set of interpretable labels and empower the advertisers to directly specify such interest labels together with their ads.

Query logs are instrumental in the improvement of search engines, and they have been under intense analysis in the past few years. There is a voluminous literature on different aspects of query-log analysis. One important problem is that of breaking up a query history into sessions [7, 10, 11, 13, 16, 18], dealing with the fact that temporal coherence does not necessarily imply thematic coherence. This is a challenging task, since different tasks tend to be interleaved or span long periods of time. Temporal correlation between queries over large number of sessions has been exploited to define semantic correlations between queries [3, 8, 20] for tasks like reformulations or query suggestions. One key differentiation of our work is that we use temporal correlations between queries to define similarity between *terms*, and then we utilize this similarity to cluster queries. In contrast, previous works define relationships directly between queries. Related to our approach is the work by Richardson [19] that discovers long-term relationships between terms in query histories.

3. OVERVIEW OF OUR APPROACH

In this paper, we address the problem of identifying user interests from search query histories, and describing them using a concise vocabulary. Given a user u and their query history Q_u, we want to assign user u a set of labels L_u, drawn from a larger vocabulary $\mathcal{L}$ of possible interests. The choice of the vocabulary $\mathcal{L}$ is of paramount importance in enabling the advertisers to select the appropriate audience for their campaign. We propose a methodology for generating the interest vocabulary $\mathcal{L}$, and an algorithm for mapping the user history to this interest vocabulary.

Our approach has two phases: the modeling phase, and the inference phase. In the modeling phase we use query histories from multiple users to generate the vocabulary of interests, and train a machine learning model that maps collections of queries to labels in our vocabulary. In the inference phase, we apply our labeling algorithm to user query histories to obtain a labeling of the users in our interest vocabulary. We now discuss the details of the two phases.

3.1 Modeling User Interests

The modeling phase takes as input a collection $\mathcal{Q} = \{Q_1, ..., Q_m\}$ of query histories of m users, and produces a vocabulary of interest labels $\mathcal{L} = \{\ell_1, ..., \ell_K\}$, and a model $\mathcal{M}$ that assigns interest labels to collections of queries. This phase can be decomposed into three steps: First, we cluster the individual query histories in order to extract *themes*; Then we use the produced clusterings to generate the label vocabulary. When available, we also augment the interest vocabulary with advertiser provided interest labels; Finally, we train a machine learning model that maps themes into the label vocabulary. The model itself is trained using data obtained *automatically* from the clusters. The pipeline of these three steps is shown in Figure 2.

[2] http://www.stanford.edu/class/msande239/

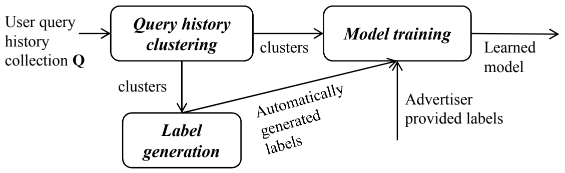

User query history collection Q
Query history clustering
clusters
Model training
Learned model
clusters
Label generation
Automatically generated labels
Advertiser provided labels

Figure 2: Modeling phase pipeline.

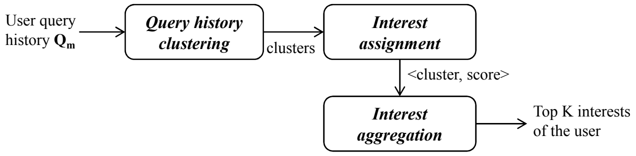


Figure 3: Inference phase pipeline.

Query History Clustering: Users express their interests in their search queries, but not necessarily in a temporally and syntactically coherent way. Queries pertaining to wildly different interests are interleaved over short intervals of time, while queries that refer to the same interest recur over the span of weeks, months, or even years, each time slightly mutated, using different terms and touching different aspects. As a first step towards extracting interests from query histories, we organize queries of individual users into *themes*: semantically coherent clusters that are potentially related to the same interest. We extract these themes by clustering the user history. For our clustering, we use a similarity measure that captures the semantic correlation between queries, as this is observed over the query histories of millions of users. We discuss our similarity measure, and clustering algorithm in detail in Section 4.1.

Formally, given the history Q_u of an individual user u, the query history clustering step produces a clustering $C_u = \{c_1, ..., c_{T_u}\}$, where each cluster of queries $c \in C_u$ corresponds to a semantically coherent theme that is candidate for capturing an interest. Given a collection of user query histories $\mathcal{Q} = \{Q_1, ..., Q_m\}$ the output of the clustering step is a collection of clusterings $\mathcal{C} = \{C_1, ..., C_m\}$, one for each individual user.

Label Generation: The clustering step does a good job in bringing together queries that are semantically related, and organizing the user query history into themes. Manual inspection reveals some clearly defined interests: an long-term engagement in online gaming, a prolonged search for a new house, or a long-standing quest for medical advice. These groups of queries make intuitive sense to a human observer, but they are not actionable for advertisers who cannot afford to go through millions of query clusters to find the ones that are of interest to their campaign. We thus need a concise way to describe the interests we observe. Using the themes we have identified, we will extract a set of interest labels, which will define the vocabulary with which advertisers can select the users they are interested in.

The label extraction process identifies key terms that can be used for labeling query clusters of individual users. It then aggregates these terms over the full history collection to identify terms that pertain to a large number of users. These terms will define the interest label vocabulary. We describe the details of this process in Section 4.2. In summary, given the collection of clusterings $\mathcal{C}$ output from the history clustering step, the label generation step will produce an interest label vocabulary $\mathcal{L} = \{\ell_1, ..., \ell_K\}$ that describes the space of possible interests of a search user.

Optionally, we can also incorporate advertiser-provided interest labels to the label set, thus allowing the advertisers to dynamically modify the set of interest labels. In Section 5.3, we show the performance of our system when advertiser-provided interest labels such as "webkinz" and "lego" are included.

Model training: We observed that the themes extracted from the clustering step align well with intuitively defined interests, and we used this fact to create our interest vocabulary $\mathcal{L}$. Given a new user u, with query history Q_u, which is clustered into a set of themes C_u, we want to be able to map these themes into the space of interest labels $\mathcal{L}$. In this step we train a discriminative model $\mathcal{M}$ that performs this task: given a cluster of queries c, it produces a probability distribution $P(\ell|c)$ over the interest labels $\ell \in \mathcal{L}$.

In order to train the model we need training data: clusters that are labeled within our label vocabulary $\mathcal{L}$. We obtain this data automatically from the clustering collection $\mathcal{C}$ we produced in the clustering step.

In summary, in the model training step, we take as input the interest vocabulary $\mathcal{L}$ produced in the label generation step, and the clustering collection $\mathcal{C}$ produced in the clustering step, and we produce a machine learning model $\mathcal{M}$ that maps a cluster of queries c into the label vocabulary $\mathcal{L}$. We describe the details of this step in Section 4.3.

3.2 Inferring User Interests

At the inference phase, given a user u, with query history Q_u, we will assign a set of labels $L_u \subseteq \mathcal{L}$ from the vocabulary of labels $\mathcal{L}$ produced in the modeling phase. This phase can be decomposed into three steps: the history clustering step, the interest assignment step, and the interest aggregation step. The pipeline for the inference procedure is shown in Figure 3.

Query History Clustering: In this step we extract the main themes in the query history of the user, using the same clustering techniques that we described in the modeling phase, which we describe in detail in Section 4.1. Given the input history Q_u we produce a clustering $C_u = \{c_1, ..., c_{T_u}\}$, where each cluster $c \in C_u$ corresponds to a theme in the user history.

Interest Assignment: In this step we apply the machine learning model $\mathcal{M}$ that we trained in the modeling phase to the clustering C_u. For every cluster $c \in C_u$, we obtain a probability distribution over the label set $\mathcal{L}$. That is, for each label $\ell \in \mathcal{L}$, we obtain the probability $P(\ell|c)$ that the cluster c should be labeled with label ℓ.

Interest Aggregation: Given the clustering C_u and the probability distributions $P(\ell|c)$ defined for each cluster c, we can aggregate them in a number of ways to obtain the consolidated user interest profile. In our case, we associate the user u with the set of labels L_u that have probability $P(\ell|c)$ above a certain threshold θ_p (set to 0.75 in our experiments), for some cluster $c \in C_u$. While being simple, the experimental results indicate that this aggregation scheme is robust and works well in practice. It is possible to exploit a variety of other signals for weighting the probability scores of the clusters such as the cluster size, the time interval over which the queries were asked, etc., but we leave this as potential problem for future investigation.

4. MODELING USER INTERESTS

We will now describe in details the three steps of the modeling phase: query history clustering, label generation, and model training.

4.1 Query History Clustering

An interest manifests itself in the query history over multiple queries that cover different aspects of the interest. For example, a user who has an interest in "football" will pose multiple queries about different football teams, NFL, or game schedules. All these queries are thematically related around the interest "football". Identifying such groups of thematically related queries poses the challenge of defining a suitable similarity measure between queries. Typical syntactic similarity measures such as Jaccard coefficient or edit distance are not sufficient, since they do not capture the diversity in the way people query about a topic. Queries like "wedding gown" and "floral arrangements" are semantically close under a "wedding" interest, yet far apart under any measure of textual similarity. Temporal affinity is a popular method for capturing such semantic correlations: related queries are likely to appear close in the user history. There is considerable amount of work in partitioning a user history into *sessions*, temporally coherent clusters of queries [4, 7, 10]. However, temporal coherence does not always guarantee thematic coherence: thematically diverse interests may be interleaved over a short period of time. Conversely, a thematically coherent interest may span several days, weeks or months in the history of a user. Therefore, sessions do not necessarily capture interests fully and accurately.

Although temporal affinity is not sufficient to capture interests in a single user history, when aggregating millions of user histories, it provides a strong signal for semantic similarity. This idea has been previously explored to extract correlations between queries for tasks such as query suggestions and query reformulations [23, 3]. In our approach we will use temporal co-occurrence of queries over multiple user histories in order to define semantic similarity between *terms*. We will then use this measure of similarity to group queries into clusters, which capture themes in the user history, and are candidates to be mapped to user interests.

Formally, let $\mathcal{Q}$ denote a collection of user histories. A user history Q_u is a sequence $Q_u = \langle (q_1, t_1), ..., (q_{n_u}, t_{n_u}) \rangle$ of query, time-stamp pairs (q_i, t_i), where query q_i was posed at time t_i. We partition the query history into *sessions* using the usual 30-minute timeout rule: a timeout of more than 30 minutes between two queries defines the beginning of a new session. The session contains the set of queries between two timeouts. Formally, a session is a maximal subset $S \subseteq Q_u$ of the query history, such that for any two query-timestamp pairs $(q_i, t_i), (q_j, t_j) \in S$, $|t_i - t_j|$ is less than 30 minutes.

Let $\mathcal{S}$ denote the set of all sessions defined over the history collection $\mathcal{Q}$. Each session $S \in \mathcal{S}$ can be thought of as a bag of words, $S = \{w_1, ..., w_k\}$, consisting of all the terms of the queries contained in S. Let $P(w_i, w_j)$ be the number of sessions where words w_i and w_j occur together. Let N be the total number of co-occurrences, that is, $N = \Sigma_i \Sigma_j P(w_i, w_j)$. For a pair of terms (w_i, w_j) we define the co-occurrence frequency $f(w_i, w_j)$ as the fraction of co-occurrences that contain both terms w_i and w_j. That is, $f(w_i, w_j) = \frac{P(w_i, w_j)}{N}$. Similarly, for a term w_i, we define the *frequency* $f(w_i)$ of term w_i to be the fraction of co-occurrences that contain term w_i. That is, $f(w_i) = \frac{\Sigma_j P(w_i, w_j)}{N}$.

In order for two terms to be similar we would like them to have high co-occurrence frequency. However, high co-occurrence frequency by itself is not sufficient to determine similarity. Terms that have high frequency on their own are likely to participate in pairs with high co-occurrence frequency. For example, queries "facebook" and "google" are prominent in the search logs, and they exhibit high co-occurrence frequency with each other and with other terms, yet this does not imply semantic similarity. To normalize for this effect, we divide the co-occurrence frequency with the probability that the two terms co-occur in the same session by chance. This ratio, or more precisely the log of this ratio, is the *point-wise mutual information* (PMI) between the two terms, a commonly used similarity measure in text mining and natural language processing [6]. Formally, it is defined as follows:

$$\text{PMI}(w_i, w_j) = \log \frac{f(w_i, w_j)}{f(w_i) f(w_j)}$$

A known drawback of PMI is that it favors rare co-occurrences. Two terms that appear only once in the the query histories in the same session, have the highest possible PMI. This is undesirable, since we would like the pair of terms to have some support in order to be deemed similar. We address this issue by using the discounted PMI (dPMI) measure [15]:

$$\text{dPMI}(w_i, w_j) = \\ \text{PMI}(w_i, w_j) \frac{f(w_i, w_j)}{f(w_i, w_j) + 1/N} \frac{\min\{f(w_i), f(w_j)\}}{\min\{f(w_i), f(w_j)\} + 1/N}$$

Given the similarity measure between terms, we can extend it to queries, or collection of queries. We represent those as bags of terms. Given two bags of terms $X = \{x_1, ..., x_{k_x}\}$ and $Y = \{y_1, ..., y_{k_y}\}$, we define their similarity as follows:

$$\text{sim}(X, Y) = \frac{1}{|X||Y|} \sum_{(x,y) \in X \times Y} \text{dPMI}(x, y)$$

That is, the similarity of the two bags of terms is the average dPMI similarity of the pairs of terms in the cross-product between the two bags.

Note that a collection of terms may contain the same term multiple times. According to our similarity definition this term will appear multiple times in the sum, and thus contribute more to the similarity. This follows the intuition that terms that are frequent should have more impact on the similarity of the collection of queries. It is also possible that X and Y share a term w. In this case we need to define a measure of similarity of a term to itself. We compute this using the definition of PMI, where we define $f(w, w) = f(w)$. Therefore, we have:

$$\text{PMI}(w, w) = \log \frac{1}{f(w)}$$

This definition captures nicely the intuition that two collections that share a rare term (*e.g.,* "aquarium") are more similar than two collections that share a frequent term (*e.g.,* "facebook").

Note that our final query similarity measure makes use of both semantic and syntactic similarity between queries. Semantic similarity is explicitly introduced by using the temporal co-occurrence of terms, while syntactic similarity is a side benefit of reducing the similarity of queries to comparisons between terms.

Equipped with a similarity measure between queries and sets of queries, we can now apply any standard clustering algorithm for grouping the queries into interests. We opt for hierarchical agglomerative clustering. The algorithm proceeds iteratively, starting with a set of singleton clusters each consisting of a single query, and at each iteration it merges the clusters with the highest similarity. It continues until the similarity of the most similar pair drops below a predefined threshold.

Therefore, given a collection of m user query histories $\mathcal{Q} = \{Q_1, ..., Q_m\}$, we have obtained a collection of m clusterings $\mathcal{C} = \{C_1, ..., C_m\}$. A cluster $c \in C_u$ in the clustering of user u is a set of queries that are thematically related, and are candidates for defining an interest of user u. In the following section, we will show how we generate the interest labels from the clusterings $\mathcal{C}$, and then label the clusters in C_u with the appropriate interest label.

4.2 Label Generation

In the label generation step, we exploit the collection of clusterings $\mathcal{C}$ that we obtained in the clustering step to automatically generate a rich set of terms that can serve as the vocabulary $\mathcal{L}$ of interest labels.

Given the collection $\mathcal{C}$, we first perform a pruning step to remove clusters with number of distinct queries below a certain threshold (set to 30 in our experiments). Such clusters are too small to capture a prevalent interest of the user. Let $\mathcal{C}_p$ denote the new clustering collection. For each cluster $c \in \mathcal{C}_p$ we produce a set T_c containing the top-q most frequent terms, where $q = 5$ in our experiments (we exclude stop words, *etc.*). These top frequently-occurring terms represent a "synopsis" of the cluster, capturing the underlying "theme" of its queries. Let $T = \cup T_c$ denote the union of all terms that are among the most frequent terms T_c of at least one cluster, for at least one user. We keep as our label set $\mathcal{L}$ the terms in T that appear in the query history of at least θ_u users, where $\theta_u = 100$ in our experiments. That is, our label set consists of terms that appear as a theme for at least 100 users. We also do minimal human inspection to remove certain labels that do not correspond to interests (*e.g.,* "map" and "store"), and canonicalize certain synonymous terms (*e.g.,* "recipes" and "recipe").

Table 1 shows a subset of the 300 labels that were generated using this approach. We can see that the labels cover a wide spectrum of interests, and at different levels of granularity. For instance, while a label like "games" tends to be more encompassing, a label such as "basketball" is more specific. There are subtle variants of similar kind of interests, *e.g.,* "food", "recipes", "baking" and "dining", to name a few. There are also time bounded interests such as "wedding" and "roofing". Furthermore, these interest labels are of high value to the advertisers. To verify this, we performed the following check: we obtained the top 1,000 unigram advertising bid terms (in terms of the revenue that they generate in a major sponsored search engine), and we computed the overlap with the list of generated interest labels. It turns out that 12.5% of these 1,000 top advertising terms are actually included in our list.

It is important to note that our approach is not restricted to using these automatically-generated interest labels. Any list of interest labels provided by the advertisers can be used, as long as it contains terms that appear in the query clusters. In Section 5.3, we experimentally show the effectiveness of our approach not only in the scenario of automatically-generated labels but also in the case of additional labels provided by the advertisers.

4.3 Model Training

Given the label set $\mathcal{L} = \{\ell_1, ..., \ell_K\}$, in this step we train a machine learning model $\mathcal{M}$ which given a cluster c, produces the probability $P(\ell|c)$ that the cluster c belongs to the interest described by label ℓ, for every label $\ell \in \mathcal{L}$. We use a multiclass logistic regression model parameterized by $\mathbf{W}$ to compute $P(\ell|c)$. The parameter matrix $\mathbf{W}$ is a collection of weight vectors $\{\mathbf{w}_k\}$, one for each label $\ell_k \in \mathcal{L}$ such that each component w_{jk} measures the relative importance of the j^{th} feature for predicting k^{th} label. The multiclass logistic regression learns a mapping from the feature vector of c, denoted by $\mathbf{z}(c)$ to the label y, using the following softmax logistic function:

$$\begin{aligned} &P(\ell_k|c) = \\ &\quad P(y = \ell_k|\mathbf{z}(c), \mathbf{W}) = \frac{\exp\left(b_k + \mathbf{z}(c)\cdot\mathbf{w}_k\right)}{1+\sum_{j=1}^{K}\exp\left(b_j + \mathbf{z}(c)\cdot\mathbf{w}_j\right)} \qquad (1) \end{aligned}$$

where b_j $(1 \leq j \leq K)$ are bias terms.

In our setting, the feature vector $\mathbf{z}(c)$ corresponds to a summary constructed from the cluster, c. In particular, we used lexical features consisting of all unigrams (terms) appearing in the queries in the cluster. We then converted them to boolean features representing the presence or absence of these unigrams. We did not consider the frequency of occurrence of these terms as that would require that the clusters are normalized for many factors such as number of queries in the cluster, number of unigrams in the cluster, *etc.* In fact, our experimental evaluation shows that these simple binary features perform effectively.

The weight vector in Equation 1 is learned from a labeled data set $\mathcal{D} = \{(\mathbf{x}^1, y^1), \ldots, (\mathbf{x}^n, y^n)\}$, where each pair $(\mathbf{x}^j, y^j)$ corresponds to the feature extracted from a cluster and its corresponding interest label. In particular, $\mathbf{W}$ is learned so as to maximize the conditional log-likelihood of the labeled data:

$$\mathbf{W}^* = \arg\max_{\mathbf{W}} \sum_{j=1}^{n} \log P(y^j = \ell_k|\mathbf{x}^j, \mathbf{W}) \qquad (2)$$

Labeled data for training: Manual construction of a labeled training set can be too expensive and time-consuming. Our approach enables an effective way to obtain large amounts of *automatically* labeled training data. For some label $\ell \in \mathcal{L}$, let C_ℓ denote the set of clusters, from any clustering $C_u \in \mathcal{C}$, such that for all $c \in \mathcal{C}_\ell$, $\ell \in T_c$, that is, the label ℓ is one of the top terms in the cluster c. We treat the clusters in $\mathcal{C}_\ell$ as positive examples for the label ℓ, and we use this data to train our model. In order for our approach to be successful, we need clusters in $\mathcal{C}_\ell$ to be homogenous, with highly frequent terms being semantically related. We manually evaluated the clusterings for their homogeneity and we confirmed that this is indeed the case. The homogeneity of the clusters follows from the way we have constructed our similarity function to capture the semantic similarity of terms.

Note that our approach generalizes naturally to the case that the set of labels we want to train against is provided from some external source (*e.g.,* provided by the advertisers). Let $\mathcal{L}'$ denote this provided set of labels. For each label $\ell' \in \mathcal{L}'$ we can use the process described above to obtain the set $\mathcal{C}_{\ell'}$ of clusters that have ℓ' in their top terms. Then we can train our model against these externally provided labels. We experiment with this case in Section 5.3.

5. EXPERIMENTAL EVALUATION

We now report the results of a large-scale, end-to-end evaluation that we performed on our system. In Section 5.1, we present the experimental setup. In Section 5.2, we describe our methodology and metrics for evaluating on advertising data. Finally, in Section 5.3, we present our key findings.

5.1 Experimental Setup

We now provide the details of the data and parameters used for the different components of our system. The clustering algorithm uses the similarity measure defined in Section 4.1, which relies on a discounted PMI computation for unigram pairs over user sessions. We computed discounted PMI over the sessions of 2.2 million users over 16 months of queries from the query log of the Bing search engine. We used the standard definition of session, where a session consists of all consecutive queries until there is a period of 30 minutes inactivity [4].

To create the set of interest labels, we ran the clustering algorithm on 580,000 users. To ensure good quality keywords, we constrained ourselves to clusters with at least 30 queries. As a result, we obtained 1,042,729 clusters mapped to their top-5 keywords. We aggregated this mapping by counting for each keyword the number of users who have at least one cluster that is mapped to the keyword. We then produced a list of all keywords that are associated to at least 100 users; after removing stop-words, plurals etc, this resulted in 5,500 keywords. The size of this list was manageable enough to be processed editorially in order to detect highly frequent terms related to interests. After editorial processing, we obtained a list of 332 interest labels. To verify that these interest labels are of high value to the advertisers, we performed the following check: we obtained the top 1,000 unigram advertising bid terms in terms of the revenue that they generate in a major sponsored search engine, and we computed the overlap with our list of interest labels. It turns out that 12.5% of these 1,000 top advertising terms are actually captured by our list. This indicates that the interest labels are, indeed, monetizable.

We note that the creation of a meaningful taxonomy, or class system, is an extremely hard undertaking that merits its own scientific field. Complete automation is nearly impossible, and probably not desirable, since introducing some human intuition can improve the quality significantly. Our approach simplifies the vocabulary creation process significantly, by offering a manageable set of labels for the data analyst to process. Processing is also simplified, consisting mostly of filtering out uninteresting or very specific terms. This is a considerably easier task compared to deriving such class labels from scratch. More importantly, the produced labels capture the underlying trends in the data, they can be updated dynamically as query histories get updated, and they come together with training data.

The Model Training component used the clustering of 120,000 users to create 116,839 (user,cluster,label) training examples. The classifier uses lexical features: we used a feature vector consisting of 56,983 binary features (these features correspond to unigrams that appeared in at least 20 queries among the 120,000 users). A logistic regression model was learned using these training examples.

At inference time, the Interest Aggregation component mapped users to the labels for which they had at least one cluster with classification score above a threshold. Unless otherwise stated, we used threshold $\theta_p = 0.75$. As we explain in the next section, the system was tested on 150,000 users using 16 month of query activity. Running the system at this scale was enabled by our implementation on large-scale Map-Reduce distributed data processing system.

5.2 Evaluation on Advertising data

The main goal of this evaluation is to study the effectiveness of our interest-aware audience selection system. Our hypothesis is that users who match advertiser-specified interests are (on average) more likely to click on ads related to the interest than the users currently selected via keyword match.

User click probability. Let A denote a set of ads, e.g., the set of ads in a specific advertising campaign, or all the ads related to a specific interest. Let U denote the set of users that are candidates to be shown the ads in A, and let $U^A \subseteq U$ denote the set of users that are actually impressed with at least one ad from the set A. Also let $C^A \subseteq U^A$ denote the subset of these users that clicked on at least one of the ads that they were impressed. We define the user click probability of the set A *with respect to the user set* U as follows:

$$P_U(A) = \frac{C^A}{U^A}$$

That is, $P_U(A)$ is the fraction of users being impressed with ads from A, that actually clicked on at least one ad from the set A.

This metric is reminiscent of the standard notion of click-through rate (CTR)[3] which, like user click probability, is also a ratio between clicks and impressions. However, CTR is the probability that, given an impression of an ad, the ad will be clicked, while user click probability is the probability that given a *user* who is impressed an ad, the user will at some point click on the ad. Although related, the two metrics capture different information. We believe that user click probability metric fits nicely with the goal of audience selection, which is to select users to whom to impress advertisements.

Now, let U_ℓ denote a set of users tagged with an interest label ℓ. Also, let A_ℓ denote a set of ads that are related to the interest ℓ (we discuss later how we obtain this set). The user click probability $P_{U_\ell}(A_\ell)$ is the probability that a user who is assigned the interest label ℓ will click on an ad related to the interest ℓ. Therefore, mathematically, our hypothesis is that on average $P_{U_\ell}(A_\ell) > P_U(A_\ell)$, that is, users associated with interest ℓ are more likely to click on an ad related to ℓ, than users drawn from the general population of all users U who are impressed with the ad.

[3] http://www.stanford.edu/class/msande239/

To test our hypothesis, we used the sponsored search logs of the Bing search engine for a 2-month period, which does not overlap with the time period used to compute user interests[4]. For each label ℓ in the vocabulary $\mathcal{L}$, we applied our algorithm to the set of users U in the 2-month query log, and we generated a subset of users $U_\ell \subseteq U$ that were assigned this label. Next, for each label ℓ we need to obtain a set of ads A_ℓ that are related to interest ℓ. It is not immediate how to obtain such a set, since currently, advertisers do not provide interest labels, and thus there exists no test set of ads labeled with interest labels that we could use for the evaluation. We tackle this problem by making the following approximation: we use the readily available, existing bid terms from advertisers as a proxy for interest labels. More specifically, let a be an ad, and let B_a denote the set of all bid keywords associated with this ad. We say that ad a is labeled with interest label ℓ if $\ell \in B_a$. We define the set A_ℓ as the set of ads that are labeled with the label ℓ.

Given the set of ads A_ℓ, we use $U_\ell^{A_\ell}$ to denote the subset of users from U_ℓ that are impressed an ad in A_ℓ. Ideally, we would like to have control over the set $U_\ell^{A_\ell}$ in our experiments. However, since we do not have such control, we define $U_\ell^{A_\ell} = U_\ell \cap U^{A_\ell}$, i.e., the subset of users labeled with ℓ, that are impressed an ad from A_ℓ in the existing logs. The set $C_\ell^{A_\ell}$ of users tagged with the interest label ℓ that clicked on the ads in A_ℓ is defined similarly. The user click probability of A_ℓ with respect to set U is $P_U(A_\ell) = |C^{A_\ell}|/|U^{A_\ell}|$ while the user click probability with respect to U_ℓ is $P_{U_\ell}(A_\ell) = |C_\ell^{A_\ell}|/|U_\ell^{A_\ell}|$.

To make it more concrete, consider the following example. We have a set of four users $U = \{Alice, Bob, Cathy, David\}$, and a set of three ads $A = \{a_1, a_2, a_3\}$. Advertisement a_1 is tagged with bid keywords $B_1 =$\{ "casino", "hotel"\}, a_2 with keywords $B_2 =$ \{"vegas", "casino" \}, and a_3 with keywords $B_3 =$ \{"vegas", "hotel"\}. In our logs, Alice is shown ads $\{a_1, a_2, a_3\}$, Bob is shown ads $\{a_2, a_3\}$, Cathy is shown ads $\{a_1, a_3\}$ and David is shown ad $\{a_3\}$. Alice clicked on ads a_1 and a_2, Bob clicked on a_3, and Cathy and David did not click on any of the ads.

Suppose now that our interest label ℓ is "casino", and that our algorithm tagged users $U_\ell = \{Alice, Bob, David\}$ with the interest label ℓ. We have that $A_\ell = \{a_1, a_2\}$, and $U^{A_\ell} = \{Alice, Bob, Cathy\}$ is the set of users that were impressed with an ad in A_ℓ. The set of users tagged with the label ℓ that are also impressed with an ad in A_ℓ is $U_\ell^{A_\ell} = \{Alice, Bob\}$. Only Alice clicked on an ad related to the interest ℓ, therefore, $C^{A_\ell} = C_\ell^{A_\ell} = \{Alice\}$. The user click probability with respect to the set of all users U is $P_U(A_\ell) = |C^{A_\ell}|/|U^{A_\ell}| = 1/3$ while the user click probability with respect to the set of users U_ℓ that we tagged with the interest ℓ is $P_U(A_\ell) = |C_\ell^{A_\ell}|/|U_\ell^{A_\ell}| = 1/2$. Therefore, in our example, among the users impressed with an ad in A_ℓ there is a 33% probability for one of them to click on an ad, while this probability increases to 50% when a user is drawn from the set of users tagged with the interest label ℓ.

User coverage. We also consider a measure of coverage which is defined as the fraction of users who are assigned an interest profile (set of labels) of a minimum given size. The goal is to show that a large fraction of users get assigned interest labels. Let U be the universe of users considered in an experiment (in our case, the users in the 2-month snapshot of the sponsored search logs). Let V_k be the users in U that are assigned a profile consisting of at least k labels by the interest-aware system. Then, the measure *user coverage for profile of size k* is defined as follows:

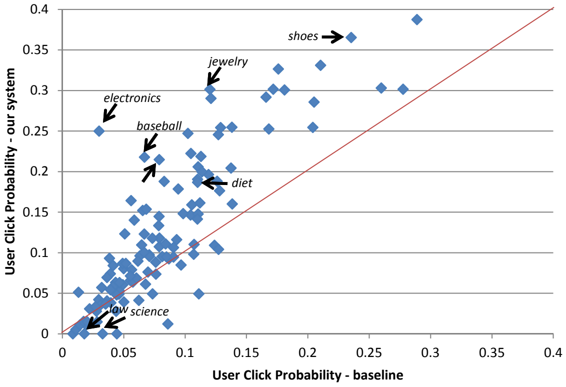


Figure 4: Scatter plot of User Click Probability of baseline vs. our system.

$$UserCoverage_k = \frac{|V_k|}{|U|}$$

Recall that in our system, we only keep the labels whose classification score for some cluster is above a classification threshold. Thus, there is a clear tradeoff between user coverage and user click probability. A high user coverage means that more users will be assigned interest labels, but this may come at the expense of retrieving "poor-quality" users who might not click on ads.

5.3 Key Findings

We now discuss the main results from our experiments.

Effect on user click probability. We first compare the user click probability of our system against that of the baseline, for every label ℓ for which there were at least 30 impressed ads in the 2 month time period that we considered for the advertising data (126 labels in total). In Figure 4, we present a scatter plot of the user click probability of the two alternatives we consider for every label ℓ. The x-axis corresponds to the user click probability $P_U(A_\ell)$ of the baseline system, and the y-axis corresponds to the user click probability $P_{U_\ell}(A_\ell)$ of our system. We can observe that the points for most labels lie above the diagonal (more precisely, for 97 out of 120 labels). This means that our system outperforms the baseline for 81% of the labels.

To get an understanding of the performance for individual interest labels, we indicate on Figure 4 some labels for which we do particularly well, and some labels for which the baseline outperforms our approach. For example, we have large gains for labels such as "baseball" and "jewelry" which represent permanent (or long-term) interests of users, and for long term tasks such as "wedding". Arguably, considering entire histories to make a determination of the user interests helps for these long-term interests. Some of the worst performing labels for our system are broad, high-level interests which do not necessarily lead to higher user click probabilities. Examples include terms such as "science" and "law".

[4] We further restricted the users to "high engagement" users, as determined by the rules of the search engine company.

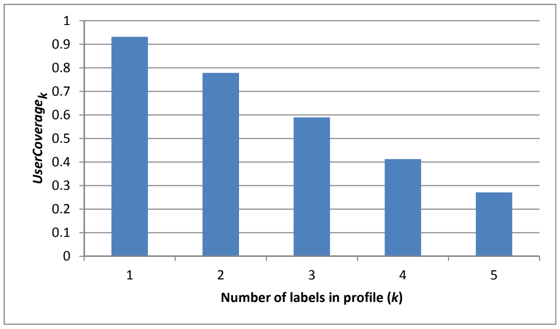


Figure 5: User Coverage for our interest-aware system.

To get an aggregated view of the results, we computed the average user click probability over all labels l. For our system, the average user click probability is 0.131; for the baseline, it is 0.087. This represents a 50.5% increase over the baseline. We used a one-tailed t-test, and verified that this difference is statistically significant with 95% confidence, thus establishing our hypothesis that on average it is more likely for a user who is tagged with an interest label to click on an ad related to that interest than a user drawn from the general population.

Our results demonstrate that given an interest of the advertiser, our technique produces an audience that has increased probability of clicking to an ad related to that interest. Of course, this audience must be of significant size. That is, $U_\ell^{A_\ell}$, the labeled users to whom the ads related to ℓ are impressed, should be a sizeable fraction of U^{A_ℓ} all the users to whom an ad related to ℓ were impressed. In our experiments $U_\ell^{A_\ell}$ is on average 10% of U^{A_ℓ}, indicating that we capture a sizeable fraction of impressed users.

Effect on user coverage. Since, in our system, a profile consists of all labels above a classification threshold, users may get profiles of different sizes (in terms of the number of labels), or even no profile at all. We measured user coverage for different minimum profile sizes for our interest-aware system. In Figure 5, we give size k of the profile in terms of number of labels on the x-axis; and the user coverage for profiles that have at least k labels, $UserCoverage_k$, on the y-axis. We can observe that the coverage is as high as 0.93 and 0.78 for histories with at least 1 and 2 labels respectively. It remains at reasonable levels even for histories of size at least five (0.27). This implies that the number of users retrieved by our system is significant for the different labels.

Sensitivity to classification threshold. We also performed a sensitivity analysis of the classification score threshold θ_p. In particular, we considered different instantiations of our system, where we varied the classification threshold θ_p and measured the corresponding average user click probability. The results are shown in Figure 6. We can observe that the average user click probability is not overly sensitive to the classification threshold: for classification threshold $\theta_p = 0.95$ it is 0.158, decreasing gently to 0.130 for threshold $\theta_p = 0.6$.

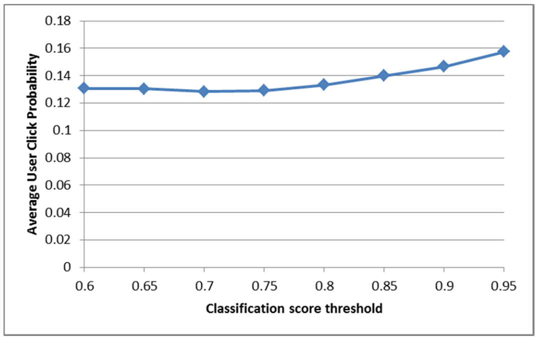


Figure 6: Average User Click Probability for different classification score thresholds.

Interest	*Baseline*	*Interest-aware*
Vegas	0.181	**0.301**
Disney	0.205	**0.286**
Hawaii	0.071	**0.097**
Lego	0.128	**0.176**
Webkinz	0.049	**0.061**

Table 2: Performance on advertiser-provided interests. The first column shows the User Click Probability of the Baseline, while the second column the User Click Probability of our method.

Enabling advertisers to extend the vocabulary of interests. So far, we have presented results on a set of interest labels derived from the clusters themselves (Label Generation component). However, our system is by no means restricted to that list: it enables advertisers to flexibly provide new interest labels and add them to the interest label set. To show that this is possible, we performed an experiment where we added some highly monetizable interests to the list of labels, and then performed the same evaluation as before but with the extended list. In particular, we added three popular tourist destinations (Vegas, Disney, and Hawaii), and two popular toys (Lego and Webkinz). We show the results on Table 2. We can observe that our system outperforms the baseline on the five interests. This means that click-through rate increases for interests that are flexibly added by the advertisers to the interest vocabulary.

6. CONCLUSIONS

In this paper, we presented a system for direct interest-aware audience selection. Our system takes the query histories of search engine users as input, extracts their interests, and describes them with interpretable labels that enable advertisers to easily target users.

We showed the effectiveness of our approach through a large-scale evaluation using advertising data. The results indicate that our system associates users to interest labels that are highly useful for advertisers to better target relevant users. Our system can lead to an increase in user click probability of over 50% compared to the baseline system.

We are planning to extend our work in multiple directions. Extensions include employing additional signals such as clicked URLs and timestamps, and studying the effect of interests on conversion rates, in addition to clicks. One particularly interesting direction involves building upon the output of the clustering algorithm to infer a "time signature" for different types of interests, based on the distribution over time of the queries in an interest cluster. This can enable us to better understand the nature of users interests. For instance, one may expect that the time signature of a time-bounded task such as planning a wedding would be different from that of a more permanent interest such as gardening.

Acknowledgements.

We would like to thank Patrick Pantel, Michael Gamon, and Maria Christoforaki for their insightful feedback. We would also like to thank Josh Bonis and Gil Lapid Shafriri for providing support and access to multiple resources needed for this project.

7. REFERENCES

[1] A. Ahmed, Y. Low, M. Aly, V. Josifovski, and A. J. Smola. Scalable distributed inference of dynamic user interests for behavioral targeting. In *KDD*, pages 114–122, 2011.

[2] E. Bakshy, D. Eckles, R. Yan, and I. Rosenn. Social influence in social advertising: evidence from field experiments. In *Proceedings of the 13th ACM Conference on Electronic Commerce*, EC '12, pages 146–161, New York, NY, USA, 2012. ACM.

[3] P. Boldi, F. Bonchi, C. Castillo, D. Donato, A. Gionis, and S. Vigna. The query-flow graph: model and applications. In *CIKM*, pages 609–618, 2008.

[4] L. Catledge and J. Pitkow. Characterizing browsing strategies in the world-wide web. In *WWW*, pages 1065–1073, 1995.

[5] Y. Chen, D. Pavlov, and J. F. Canny. Large-scale behavioral targeting. In *KDD*, pages 209–218, 2009.

[6] K. Church and P. Hanks. Word association norms, mutual information, and lexicography. In *ACL*, pages 22–29, 1989.

[7] D. He, A. Göker, and D. J. Harper. Combining evidence for automatic web session identification. *Information Processing and Management*, 38:727–742, 2002.

[8] H. Hwang, H. W. Lauw, L. Getoor, and A. Ntoulas. Organizing user search histories. *IEEE Transactions on Knowledge and Data Engineering*, 24(5):912–925, 2012.

[9] J. Jaworska and M. Sydow. Behavioural targeting in on-line advertising: An empirical study. In *WISE*, pages 62–76, 2008.

[10] R. Jones and K. L. Klinkner. Beyond the session timeout: automatic hierarchical segmentation of search topics in query logs. In *CIKM*, pages 699–708, 2008.

[11] A. Kotov, P. N. Bennett, R. W. White, S. T. Dumais, and J. Teevan. Modeling and analysis of cross-session search tasks. In *SIGIR*, pages 5–14, 2011.

[12] K. Liu and L. Tang. Large-scale behavioral targeting with a social twist. In *CIKM*, pages 1815–1824, 2011.

[13] Q. Mei, K. Klinkner, R. Kumar, and A. Tomkins. An analysis framework for search sequences. In *CIKM*, pages 1991–1994, 2009.

[14] S. Pandey, M. Aly, A. Bagherjeiran, A. Hatch, P. Ciccolo, A. Ratnaparkhi, and M. Zinkevich. Learning to target: what works for behavioral targeting. In *CIKM*, pages 1805–1814, 2011.

[15] P. Pantel and D. Lin. Discovering word senses from text. In *KDD*, pages 613–619, 2002.

[16] B. Piwowarski, G. Dupret, and R. Jones. Mining user web search activity with layered bayesian networks or how to capture a click in its context. In *WSDM*, pages 162–171, 2009.

[17] F. J. Provost, B. Dalessandro, R. Hook, X. Zhang, and A. Murray. Audience selection for on-line brand advertising: privacy-friendly social network targeting. In *KDD*, pages 707–716, 2009.

[18] F. Radlinski and T. Joachims. Query chains: learning to rank from implicit feedback. In *KDD*, pages 239–248, 2005.

[19] M. Richardson. Learning about the world through long-term query logs. *ACM Trans. Web*, 2:21:1–21:27, 2008.

[20] E. Sadikov, J. Madhavan, L. Wang, and A. Y. Halevy. Clustering query refinements by user intent. In *WWW*, pages 841–850, 2010.

[21] S. K. Tyler, S. Pandey, E. Gabrilovich, and V. Josifovski. Retrieval models for audience selection in display advertising. In *CIKM*, pages 593–598, 2011.

[22] J. Yan, N. Liu, G. Wang, W. Zhang, Y. Jiang, and Z. Chen. How much can behavioral targeting help online advertising? In *WWW*, pages 261–270, 2009.

[23] Z. Zhang and O. Nasraoui. Mining search engine query logs for query recommendation. In *WWW*, 1039-1040.