Modeling the Impact of Short- and Long-Term Behavior on Search Personalization

Paul N. Bennett[1], Ryen W. White[1], Wei Chu[2], Susan T. Dumais[1],
Peter Bailey[2], Fedor Borisyuk[3], and Xiaoyuan Cui[2]
Microsoft Research[1], Microsoft Bing[2], Microsoft Server & Tools[3]
One Microsoft Way, Redmond, WA 98052 USA
{pauben, ryenw, wechu, sdumais, pbailey, fedbor, xcui}@microsoft.com

ABSTRACT

User behavior provides many cues to improve the relevance of search results through personalization. One aspect of user behavior that provides especially strong signals for delivering better relevance is an individual's history of queries and clicked documents. Previous studies have explored how short-term behavior or long-term behavior can be predictive of relevance. Ours is the first study to assess how short-term (session) behavior and long-term (historic) behavior interact, and how each may be used in isolation or in combination to optimally contribute to gains in relevance through search personalization. Our key findings include: historic behavior provides substantial benefits at the start of a search session; short-term session behavior contributes the majority of gains in an extended search session; and the combination of session and historic behavior out-performs using either alone. We also characterize how the relative contribution of each model changes throughout the duration of a session. Our findings have implications for the design of search systems that leverage user behavior to personalize the search experience.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval – *search process*; *selection process*.

Keywords

Personalization; Web search.

1. INTRODUCTION

Search personalization improves retrieval effectiveness by tailoring the ranking of results for individual users based on models of their interests [24][28][29]. To construct the profiles necessary for search personalization, evidence of a user's interests can be mined from observed past behaviors. This behavior can be sourced from the short-term (e.g., the current search session) [34] or the long-term (e.g., across many previous sessions) [25]. These studies have shown that personalization is important but often care must be taken in how it is applied, e.g., we may only want to personalize queries which have high click entropy [11][30].

An important determinant of the success of personalization is the behavioral information that is used to construct user profiles. Although there has been some work examining the effect of different contextual sources for modeling user interests [22][33], another critical aspect of personalization is the timespan of the behavioral information used for profile construction. Short-term profiles capture recent interactions but lack users' long-term interests. Long-term profiles represent long-term interests but may not adequately represent searcher needs for the current task. Earlier attempts to address this challenge leveraged different representations for each source [18] or made ad hoc decisions around how to weight distant actions [27]. A principled investigation of the impact of short- and long-term behavior on search personalization is lacking and we address that shortcoming with the research presented here.

In this paper, we investigate how users' long-term search activity history interacts with their short-term search session behavior. We characterize these interactions using a framework for modeling behavior from different timespans and predicting search relevance. We explore the effectiveness of user profiles developed based on different temporal views. Although each model makes use of sets of search activity gathered over different durations, the same feature set is used for each time span to remove that source of variation, and decay factors (among other things) are studied in a principled manner. We evaluate the success of these models via a large search log containing queries, results, and clicks, enabling us to compare the performance of each personalized ranker relative to that of a high quality commercial search engine in a manner similar to previous personalization research [8][25][34]. We make the following unique contributions with this research:

- Propose a novel unified modeling framework that provides an integrative view of different parameters of personalization and controls key aspects such as the features generated from behavior and decay factors employed.
- Study dynamics in the relative contribution to personalization of short- and long-term models over the course of a session. As part of our analysis, we confirm intuitions that long-term behavior is useful at the start of a session and that short-term models yield benefit as the session proceeds.
- Provide new findings on search personalization, such as the special properties of the first query in the session, and the strong performance of models that *learn* to combine short- and long-term features for each query, rather than simply aggregating all features; suggesting that individual queries differentially benefit from short- and long-term personalization.

The remainder of this paper is structured as follows. Section 2 presents related work on model-based user behavior analysis for search personalization. Section 3 describes our unified framework for combining a user's (long-term) historical behavior with their (short-term) session activity and outlines the features and model training. Section 4 describes the experiment, including the data and methodology. We present findings in Section 5, discuss them and their implications in Section 6, and conclude in Section 7.

2. RELATED WORK

There is growing interest in the information retrieval (IR) community in examining how knowledge of a searcher's interests and context can be used to improve various aspects of search such as

SIGIR'12, August 12–16, 2012, Portland, Oregon, USA.

ranking or query suggestion. Here we review prior research that examines the use of implicit user profiles generated using a user's searching and browsing actions (queries, clicks on search results, and subsequent navigation) to personalize Web search.

Recent investigations that employ a user's search and browse actions to influence search personalization include those based on: a user's location [1], a user's history of search activity [25], the ability of a user to read at differing levels of complexity [8] and patterns of re-finding the same search result [31]. Others discuss how different forms of context and search activity may be used to cast search behavior as a prediction problem [22][33].

Several research groups have investigated personalizing search results using user profiles that comprise topical representations of users' search interests. Gauch et al. [14] learned user profiles from browsing history, Speretta and Gauch [26] built profiles using search history, and Chirita et al. [7] and Ma et al. [19] used profiles that users specified explicitly. In all cases, interest profiles were compared with those of search results and used to affect the order in which results were presented to individuals. Bennett et al. [2] demonstrated how category features from the Open Directory Project (ODP, dmoz.org) could be used to improve search ranking in the aggregate for all users but not for individual searchers.

The context of search activities within the current session has been used to build richer models of interests and improve how the search system interprets the user's current query. Cao et al. [4][5] represented search context within a session by modeling the sequence of user queries and clicks. They learned sequential prediction models from large-scale log data, and applied the models to URL recommendation, query suggestion, and query categorization. Mihalkova and Mooney [21] used similar search session features to disambiguate the current search query. White et al. [34] constructed topical models of searcher's interests using the current query and recent search activities such as queries and hyperlink clicks, and used these models to predict future interests. Models of short-term interests based on search queries and result clicks have also been used to improve search quality [9][24][36]. For example, Xiang et al. [36] developed heuristics to promote search results with the same topical category if successive queries in a search session were related by general similarity, and were not specializations, generalizations or reformulations.

Much of the work on search personalization focuses on longer-term models of user interests. Teevan et al. [29] developed rich long-term user models based on desktop search activities to improve ranking. Matthijs and Radlinski [20] developed models of users' interests using browsing behavior and evaluated their ranking improvements using an interleaving methodology (merging original and personalized rankings and observing SERP clicks). Sontag et al. [25] developed generative and discriminative probabilistic models using ODP category models from historical click data. They learned parameters based on the divergence of individual user behaviors from normative behaviors to re-rank search results and found the largest gains for an ensemble of the two models. Tan et al. [28] studied long-term language model-based representations of users' interests based on queries, documents and clicks. They considered different amounts of history and found that for fresh queries recent history was the most important, but for recurring queries longer-term history was more important.

Although long-term models described above include short-term information, few explicitly study it separately. Dou et al. [11] learned both click-based and topic-based user models over a 12-day period. Personalization was most effective for queries with high click entropy (as has also been observed by Teevan et al. [30]). Topic profiles resulted in highly variable performance across queries, and personalization was more effective for users with more historical information. Li et al. [18] modeled short- and long-term user activities but used different representations for each. They built long-term profiles using topics from the Google Directory of previously-clicked results and short-term models using a cache of recently-clicked results. They compared re-ranking based on user models with Google Directory and observed some advantages of personalization for users with varied profiles (so-called ambiguous users). However the study involved only 12 users over a ten-day period, and they were instructed to search for specific topics, hobbies and to repeat queries. Sugiyama et al. [27] modeled users' interests with both ephemeral short-term preferences and persistent long-term preferences, with an exponential decay on the importance of older information. They observed small improvements for personalized methods.

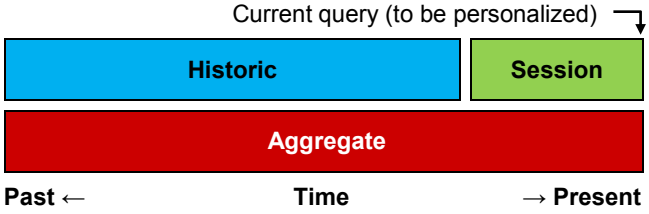

Figure 1. An illustration of how we create profiles from recent (*Session*), past (*Historic*), or a combination (*Aggregate*).

The research presented in this paper differs from previous work in several important ways. First, we present a unified framework over how interests are represented (as topics or URLs) and how short- and long-term temporal dynamics are modeled. Second, we examine the effects of the query position within a session on how these behavior models change in their relative contribution. Third, we show how to learn models that effectively combine short- and long-term behavior to create improved personalization models. Finally, our evaluation is conducted on large-scale logs from a major search engine over a two-month timespan, thus addressing scale and representativeness issues inherent in smaller studies.

3. PERSONALIZATION FRAMEWORK

We can model users' preferences from different temporal *views* of their history of interaction with the search engine. Figure 1 illustrates the relationship between each of the three temporal views that comprise the framework. The figure also shows the position of the current query which we would like to personalize. We can build a model based only on recent interactions in an attempt to capture the user's current focus. One such approach used previously, and which we adopt in this work, is to look at *session* interaction [34]. Additionally, we consider the *historic* interactions that a user had with the search engine prior to the current session, potentially comprising many days or weeks of activity. This long-term information may be useful in disambiguating underspecified queries by preferring results on topics known to be historically of interest to the user [25]. In addition, since information seeking tasks may extend across many sessions [17] the current task (session) may relate to previous sessions. This suggests a model that *aggregates* all history over a shifting time window, effectively combining short- and long-term interests.

To compare the contributions of the three different temporal views, we need a framework that yields comparable features in each of them. Otherwise, the success of one view may simply be an artifact of having modeled a component in one view that is missing from the other views. To do this, we present a novel

framework that incorporates both functions that correspond to the temporal views depicted in Figure 1 and time-weighting functions. We use two representations that are commonly studied in the literature: topics and URLs [14][20][25] and show how features often studied in prior research emerge from our framework.

We now present our framework and explain how it incorporates the following factors that are understood to affect personalization quality [28][31][34]: recency; the similarity of the current query to past user queries; the similarity of a document to be ranked for the current query to the documents previously returned to a user; and how the user interacted with those past results.

3.1 Framework Overview

Within the context of personalizing search based on past behavior, we consider a number of different factors. For example, given the user has issued query q, from the temporal view on the user's past search interactions, $\text{view}(u_I)$, we can consider all related queries, $\text{related}(q, u_I)$ where u_I is the set of the user's past issued queries, the search results, and any behavioral interactions the user had with the results. We focus on clicks but skips (explicitly ignoring a result) and misses (failing to notice a result) can also be considered. The function $\text{related}(q, u_I)$ returns a set of queries that are related to q. For each related query, q_r, we also model the strength of its relationship to the current query for this user, $\text{w}(q_r, q, u_I)$, which we abbreviate as w_{q_r}. For each related query, the search engine returns a set of results, $\text{results}(q_r)$, whose elements we refer to as d_{q_r}, the documents returned for the related query. For each d_{q_r}we consider both its similarity, $\text{sim}(d_{q_r}, \text{d})$, to a document, d, for the current query when determining how we should estimate the relevance of d, and we may also consider what action, denoted as $\text{action}(q_r, d_{q_r})$, the user took with respect to the document (e.g., a satisfied click serving as an indication that the document is relevant [12][13]). We can write a family of features related to personalization parameterized by choices for the related query function, the relationship weight, the similarity and the action. One such simple formulation that considers all of these factors to generate a feature value from a $\langle q, d, u_I \rangle$ triple is:

$$f(q, d, u_I) = \sum_{q_r \in \text{related}(\text{q}, u_I) | q_r \in \text{view}(u_I)} w_{q_r} \sum_{d_{q_r}} \text{sim}(d_{q_r}, \text{d})\ \text{action}(q_r, d_{q_r})$$

Given appropriate choices for the related query function, the weighting, similarity, and action, one can derive many commonly studied features for personalization. For example, when the set of related queries comprises all queries in the user's past interaction, the relationship weight is uniformly 1, the similarity function returns 1 for identical URLs and 0 otherwise, and the action is whether or not the user clicked on the document from the related query, then the resulting feature is simply the number of times the user has clicked on d. With the same choices except where the related queries are only queries from the past interactions that are identical, then one obtains the number of previous times the user clicked on d when issuing q, a quantity highly indicative of refinding in personalization research [31].

This framework is useful for studying different temporal views because it allows us to vary the weight and related query functions to capture temporal effects while ensuring each time view has comparable features. Furthermore, it is amenable to choosing multiple types of similarity functions and yielding comparable feature families. For example, given a representation of a document, d, as a vector of topic probabilities then one reasonable similarity function would be the inner product between the current document and the document from the related query, d_{q_r}. Again assuming that all queries in a user's history are uniformly weighted and the action is a click, then the resulting feature is the inner product between the topics of results that the user has clicked on in the past and the topics of the current document.

Seeing how this amounts to the inner product over all clicked on topics and d's topics is instructive in understanding how these features could be computed efficiently in practice. For any similarity function that can be written as the inner product of two documents by using a representation of the document, $\Phi(d)$, such that $\text{sim}(d_{q_r}, d) = \langle \Phi(d_{q_r}), \Phi(d) \rangle$, we can rewrite the inner product as an inner product of d and an aggregated weight vector: $f(q, d, u_I)$

$$= \langle \Phi(d), \sum_{q_r \in \text{related}(\text{q}, u_I) | q_r \in \text{view}(u_I)} w_{q_r} \sum_{d_{q_r}} \Phi(d_{q_r})\ \text{action}(q_r, d_{q_r}) \rangle$$

$$= \langle \Phi(d), \omega(\, q, d, u_I) \rangle$$

where $\omega(\, q, d, u_I)$ is shorthand for the weight vector derived from the sum. This new weight vector is more convenient to work with because it can be updated simply by adding in the most recent interactions for any case where the relationship weight does not change based on the overall interaction history. Even when the related queries selection is more complex such as including all queries that are a superset of the current query, this can be computed efficiently by constructing an inverted index for each user mapping query words to weight vectors, $\omega(\, q, d, u_I)$.

When the representation $\Phi(\cdot)$ returns a document's topic vector, one obtains quantities commonly used in studying personalization by topic [25]. When the representation is based on identical URLs, we capture that by defining $\Phi(\cdot)$ as a sparse vector over all URLs (i.e., 1 for the argument URL and 0 elsewhere).

Now that we have laid the foundations for the framework and shown that it can be used to represent several previously-proposed personalization features, we address the problems of choosing functions for related queries, relationship weight, similarity, and actions motivated by literature on personalized search. Each combination of these will yield a feature which we use as input to the machine-learned ranker examined in this study.

3.2 Query Selection and Weighting

3.2.1 Temporal Selection

We now return to our depiction of time views of a profile in Figure 1 and define concretely how we represent these views. We represent the *Session*, *Historic*, and *Aggregate* views as choices of the $\text{view}(u_I)$ function. That is, the *Session* view returns the queries from the current session. If the related function is "identical queries", then together with the *Session* view filter the sum is over "identical queries within the session." Likewise, the *Historic* view returns queries before the current session, and the *Aggregate* view returns all queries in the user's past interactions. Thus, given choices for the parameterization of weight, similarity, and action, instead of having a single feature, we now have three features: a session version, a historic version, and an aggregate version. When we turn to modeling later in the paper, we separate the features and build models using only features from each view enabling us to study each view and compare the views with the same number and types of features in each.

Note that we chose to make a session distinction since sesssions have been commonly used as a proxy to identify task boundaries [15][17]. Session has been used in the context of personalization

to predict short-term activity [33][34], and it is currently of broad interest to the IR community as a track in the Text Retrieval Conference [16]. For our purposes a session is demarcated by 30 minutes of user inactivity as described in [32].

3.2.2 Temporal Weighting

In addition to the session boundary, we may also posit that recent interactions of a user matter more than distant ones. This has been captured in personalization by introducing a decay function on past interactions [10][23][27][34]. We introduce a similar decay into our framework by setting the relationship weighting function, w_{q_r}, to a decay value. Let $p(q_r)$ refer to the number of queries in the time view (session, historic, aggregate) by which the related query precedes the current query. Thus $p(q_r)$ =1 is the most recent previous query (in session, history, and aggregate respectively). Then we choose $c^{p(q_r)-1}$ as the decay, where c is a decay factor. Rather than emphasizing absolute time, this emphasizes recent activity. Figure 2 shows the weights for various decay factors over previous queries. For experiments reported in this paper, we chose c=0.95 (shown as a red dotted line in the figure) because it lies between the extremes of massively emphasizing recent activity and uniformly emphasizing all actions. We also used simple uniform weighting (c=1), shown as the blue solid horizontal line at the top of Figure 2.

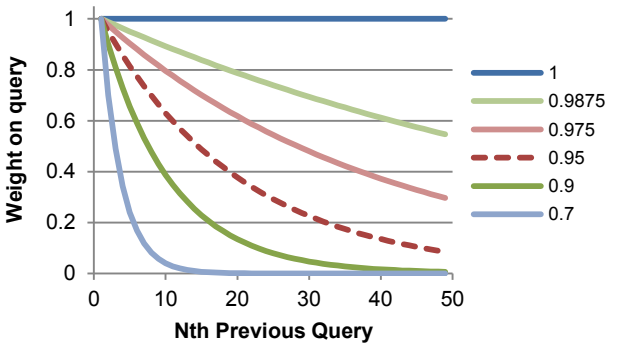

Figure 2. Query decay weights given various decay factors.

3.2.3 Query Generalization and Specialization

Various studies have demonstrated that users interact differently with results if they have recently generalized a query (reformulated by dropping words) or specialized it (reformulated by adding words) [36]. Because of this we introduce four choices for the related(q, u_I) function: (1) all queries in the user's profile (generated using view(u_I)), giving rise to features that capture the user's preference for a document independent of the current query, q; (2) all queries exactly matching q in the user's profile, which captures specific interactions; (3) all queries in the profile that are (improper) subsets of q, which captures behavior on generalizations of q (i.e. a subset has fewer words and is therefore more general); (4) all queries in the profile that are (improper) supersets of q, which captures behavior on all specializations of q. The subsets and supersets were determined after stopwords were removed and had to share at least one non-stopword with q (i.e., empty sets were not permitted).

3.3 URL Representation

As noted in Section 3.1, the user's interaction with a particular document is a useful predictor for personalization as an indicator of re-finding behavior. Therefore, we use one choice of $\Phi(\cdot)$ which returns a sparse vector over URLs (document IDs) that is 1 for the dimension corresponding to the URL of Φ's argument and 0 elsewhere. As further discussed in Section 3.1, this gives rise to a variety of useful features depending on the remaining parameter choices, e.g., the user's preference (measured as number of clicks) for a document: (1) across all queries; (2) for the current query, q; (3) for generalizations of q; and (4) for specializations of q.

3.4 Topical Representation

Topical representations of documents are commonly used in personalization studies (e.g., [14][25][34]). To obtain such a topic representation for this study, we labeled each document with a vector of probabilities of categories from the top two levels of the ODP hierarchy using a text-based classifier. Each document's vector was restricted to the three most probable classes. The classifier has a micro-averaged F1 value of 0.60 and is described more fully in [2]. As mentioned in Section 3.1, given appropriate choices for the remaining parameters, this representation will yield features that capture the user's preference for the topics of d: (1) across all queries; (2) for the current query, q; (3) for generalizations of q; and (4) for specializations of q. To measure the similarity between the document's topical representation and $\omega(q, d, u_I)$ resulting from the topic parameterization we use the cosine similarity, which is commonly used in the literature.

3.5 Search Actions

While a variety of behavioral signals have been studied in the literature, we restrict our choice for the action(q_r, d_{q_r}) function to *satisfied* (SAT) clicks on the search engine result page. A SAT click involves a user dwelling on the result for at least 30 seconds or one which terminates the search session [12][13]. We focus on SAT clicks since research indicates that dwell time is indicative of relevance and that clicks with short dwell times ("quick backs") are unlikely to be relevant [37].

3.6 Profile Information Measures

Personalization studies have also commonly introduced features that measure the amount of information known about the user or the appropriateness of personalizing for a query [1][8][30][34]. Similar notions also emerge from our framework. When the choice of relationship weight, the representation, and action are all non-negative, then $\omega(q, d, u_I)$ can be treated as describing a probability space over the representation space, $\Phi(\cdot)$, by normalizing across the feature dimensions. In fact, if we further summed across all users when the relationship is exact query match, with the URL representation, and click actions, then the entropy of the resulting vector is *QueryClickEntropy*, commonly referred to as click entropy in earlier personalization research [11][30]. To simplify the personalized versions because tracking all URLs for a user can be costly, we make a simplifying assumption and compute this entropy over (clicked) rank positions instead of URLs. If the same query always returns the same search results in the same rank order, then the personalized click entropy for a query (*UserQueryPositionEntropy*) is the same as a personalized computation for standard click entropy.

Likewise, we use the topical representation and compute entropy of the resulting $\omega(q, d, u_I)$. In which case if we sum across users, we obtain *QueryTopicEntropy* (used to identify ambiguous queries for personalization in [25]), conditional on the user across all queries we obtain *UserTopicEntropy*, conditional on user and query we obtain *UserQueryTopicEntropy*, and similarly for subset and superset. Because these entropy-based features derive from the appropriate parameterization of $\omega(q, d, u_I)$, they can depend on each time view and use decay or uniform weighting.

Table 1. Summary of Features. For each parameterized feature, the feature type $\mathbf{action}(q_r, d_{q_r})$ = SAT click. When a row has the "Temporal Selection" box checked then a version is instantiated for each value of $\mathbf{view}(u_I) = \boldsymbol{Session, Historic, All}$. When a row has the weight box checked then a version is instantiated for each value of w_{q_r} = uniform, decay. Thus rows with both temporal selection and temporal decay correspond to 3 views × 2 weight = 6 features.

Temporal Selection	Weight	Feature	Description
Query-Doc-User Features			
×	×	*DocTopicCosineUserTopicProfile* : cosine of $\Phi(d), \omega(q, d, u_I)$ with related(q, u_I)=all, $\Phi(\cdot) = \text{Topic}$	Topic similarity of overall user's history to this URL
×	×	*UserClicksOnUrl* : inner prod. of $\Phi(d), \omega(q, d, u_I)$ with related(q, u_I)=all, $\Phi(\cdot) = \text{URL}$	Click count of overall user's history to this URL
×	×	*DocTopicCosineUserTopicProfileForQuery* : cosine of $\Phi(d), \omega(q, d, u_I)$ with related(q, u_I)=q, $\Phi(\cdot) = \text{Topic}$	Class similarity of selected user history (exact query) to this URL
×	×	*UserClicksOnUrlForQuery* : inner prod. of $\Phi(d), \omega(q, d, u_I)$ with related(q, u_I)=q, $\Phi(\cdot) = \text{URL}$	Click count of selected user history (exact query) to this URL
×	×	*DocTopicCosineUserTopicProfileForSubsetQuery* : cosine of $\Phi(d), \omega(q, d, u_I)$ with related(q, u_I)=subset q, $\Phi(\cdot) = \text{Topic}$	Class similarity of selected user history (subset of query after stopword removal) to this URL
×	×	*UserClicksOnUrlForSubsetQuery* : inner prod. of $\Phi(d), \omega(q, d, u_I)$ with related(q, u_I)=subset q, $\Phi(\cdot) = \text{URL}$	Click count of selected user history (subset of query after stopword removal) to this URL
×	×	*DocTopicCosineUserTopicProfileForSupersetQuery* : cosine of $\Phi(d), \omega(q, d, u_I)$ with related(q, u_I)=superset q, $\Phi(\cdot) = \text{Topic}$	Class similarity of selected user history (superset of query after stopword removal) to this URL
×	×	*UserClicksOnUrlForSupersetQuery* : inner prod. of $\Phi(d), \omega(q, d, u_I)$ with related(q, u_I)=superset q, $\Phi(\cdot) = \text{URL}$	Click count of selected user history (superset of query after stopword removal) to this URL
Query Features			
Query Ambiguity Measures			
		QueryClickEntropy (commonly called click entropy in the literature)	Measures the diversity of clicks across users. Higher entropy indicates queries with more intents
		QueryTopicEntropy	Higher entropy indicates topically ambiguous.
Query Difficulty Measures			
		PositionInSession	Easy queries come early in a session with reformulations later
		QueryLength	Query length has been shown to be predictive of query difficulty
		QueryFrequency	More frequent queries often have more click information
Query-Doc			
		Rank	Rank of base ranker – non-personalized estimate of relevance
Query-History Features			
Query Number Features in Profile (not quite analogs)			
×		*NumberOfQueries*	Number of distinct queries in interaction view
×		*NumberOfSessionsWithQuery*	Number of sessions in view containing this query
×		*NumberOfSubsetQueries*	Number of distinct queries in view matching subset relation
×		*NumberOfSupersetQueries*	Number of distinct queries in view matching superset relation
Focus of User Profile			
×	×	*UserTopicEntropy:* entropy of normalized $\omega(q, d, u_I)$ with related(q, u_I)=all, $\Phi(\cdot) = \text{Topic}$	Measures diversity of the user's observed needs
×	×	*UserQueryTopicEntropy:* entropy of normalized $\omega(q, d, u_I)$ with related(q, u_I)=q, $\Phi(\cdot) = \text{Topic}$	Measure of diversity of topics that have satisfied the user's need for this exact query in the past
×	×	*UserSubsetQueryTopicEntropy:* entropy of normalized $\omega(q, d, u_I)$ with related(q, u_I)=subset q, $\Phi(\cdot) = \text{Topic}$	Measures diversity of topics that have satisfied the user's need for the selected (subset) history
×	×	UserSupersetQueryTopicEntropy: entropy of normalized $\omega(q, d, u_I)$ with related(q, u_I)=superset q, $\Phi(\cdot) = \text{Topic}$	Measure of diversity of topics that have satisfied the user's need for the selected (super) history
×	×	*UserPositionEntropy:* entropy of normalized $\omega(q, d, u_I)$ with related(q, u_I)=all, $\Phi(\cdot) = \text{Position}$	Higher positional entropy means that the user is not always satisfied by results in the top position
×	×	*UserQueryPositionEntropy:* entropy of normalized $\omega(q, d, u_I)$ with related(q, u_I)=q, $\Phi(\cdot) = \text{Position}$	Higher positional entropy means that the user is not always satisfied by results in the top position for this query. If the results for a query are always stable, this is personalized click entropy

We also introduce simple features that measure the number of queries or sessions containing the query in each view. We use these features in our model and describe them in detail in Table 1.

3.7 Additional Features and Summary

In addition to the features listed thus far, we also have a number of other non-personalized features of the query and the search results that we use in our models. These include the query click entropy and query class entropy (whose relationship to the framework is described in Section 3.6), which measures the diversity of clicks across users, and the diversity of topics in the search results for the current query, both derived from historic log data. Other features such as the length of the query and the rank position of each of the result URLs, as determined by the baseline ranker, are also included. A full description with additional motivation of each of these additional features, as well as all features described in text above, is provided in Table 1.

In summary, we consider several features, 3 temporal views and 2 weights. There are 14 rows in the table that depend both on temporal selection *and* weight accounting for 14×3×2=84 features. There are four features that depend *only on* temporal selection accounting for 4×3=12 features, and there are 6 features that have no temporal selection or weight – resulting in 102 features total. Restricting the set of features used to a single temporal view yields 14×2+4+6=38 features.

4. EXPERIMENTAL METHODOLOGY

Using the models described in the previous section, we aim to answer questions about the relative performance of each of the temporal views. In this section we describe the method that we followed for our experiments. We begin by describing the set of research questions that we answer in our study.

4.1 Research Questions

The study investigates the following conditions, each specifying the amount and type of search history used for personalization:

1. **Session:** All previous actions in current search session. The full set of features in Table 1 instantiated for the *Session* view yielding 38 features.
2. **Historic:** All previous actions apart from those in the current session. The full set of features in Table 1 instantiated for the *Historic* view yielding 38 features.
3. **Aggregate:** All previous actions before the current query. The full set of features in Table 1 instantiated for the *Aggregate* view yielding 38 features.
4. **Union** of *Session*, *Historic*, and *Aggregate*: Combines the sets of features associated with each of the three other views. The full set of features in Table 1 instantiated for the each of the *Session*, *Historic*, and *Aggregate* views yielding 102 features (32 temporal selection features × 3 + the 6 additional features).

Adding the *Union* condition allows the ranker to determine how much weight to put on each of the three possible time views.

Using these four conditions, we train models that re-rank the top Web search results provided by a large commercial search engine. The baseline used for our experiments is the original ranking of the top ten results provided by the engine. Note that this baseline is highly competitive and outperforming it is very challenging.

We address the following three research questions:

1. Do short- and long-term models both provide evidence for improved personalization? (Session, Historic vs. baseline)
   - Which one provides more? (Session vs. Historic)
   - Do they provide additive information? (Union vs. the best-performing model of Session and Historic)
2. Does aggregating all activity (with or without a decay factor emphasizing recent activity) capture the full interaction of short- and long-term models? (Aggregate vs. Union)
3. Do some features act differently in short- versus long-term models? (Can the performance of Union be explained by any of Session, Historic, or Aggregate?)

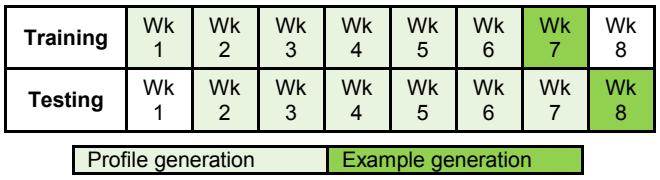

Figure 3. Illustration of our usage of data to profiles relative to the training and testing sets.

Answers to these questions provide valuable insight about the relative utility of the time views and help inform decisions about when and how to use these views for search personalization.

4.2 Data Set and Evaluation Methodology

The primary source of data for this study is a proprietary data set comprising anonymized logs of users of the Microsoft Bing search engine. The logs contained a unique user identifier, a search session identifier, the query, the top-10 URLs returned by the search engine for that query, and clicks on the results. We used eight weeks of log data gathered from July and August 2011 to train and evaluate our different models. Logs were collected from flights where other personalization support was disabled, so as to not bias our results with other personalization signals.

For evaluation, we need a personalized relevance judgment for each result. Obtaining many explicit relevance judgments from real users is impractical, and there is no known approach to train expert judges to provide reliable judgments that reflect real user preferences. Hence we obtained these judgments using a log-based methodology inspired by [12] and similar to that used in [2][8]. Specifically, we assign a positive judgment to one of the top 10 URLs if it is a *satisfied result click* (SAT click). We define a SAT click in a similar way to previous work [12] as either a click followed by no further clicks for 30 seconds or more, or the last result click in the session. We also assign a positive judgment to a URL if it is a SAT click in one of the following two queries in the session -- as long as all queries up to the SAT click share at least one URL in the top 10 with the original query. The remaining top-ranked URLs receive a negative judgment. This gives us a positive or negative judgment for each of the top-10 URLs for each query. The rank positions of the positive judgments are used to evaluate retrieval performance before and after re-ranking. Specifically, we measure our performance using the mean average precision (MAP) of the re-ranked lists. This is the mean of the average precision attained for each of the queries across the top-10 results retrieved before re-ranking (for the baseline) and after re-ranking (for each of the views of interest). Queries for which we cannot assign a positive judgment to any top-10 URL are excluded from the dataset.

We used a modified form of five-fold cross validation by user for training and testing, splitting based on user identifier. This is modified in the sense that during each fold 80% of the users were used for training and 20% of the users were held out for testing, but all of the test data came from the week after the training data. Split-

ting on user meant that there are no overlapping users between training and test. We did this because we wanted to ensure that the predictive patterns we learned also apply to users not seen during model training (as would be the case when deploying such models in practice). Figure 3 demonstrates how we extracted training and test data from the logs. Both training (week 7) and testing (week 8) data use the six weeks immediately prior to each to ensure feature distributions between train and test are based on the same amount of profile information. For each query in week seven, the historic features are computed based on actions from up to six weeks before that query (weeks 1–6). For users in the test fold (week 8), we used data from weeks 2–7 to build the profiles. Because we aim to compare the contributions of both historic and recent activity, if users lack search activity in a timespan then reliably comparing our experimental outcomes becomes challenging. We leave studying how this tradeoff changes with the amount of available history from a user as future work. Therefore, we restricted users to those with at least one SAT click in each of the six weeks before the week of interest (note this only uses weeks 1–7 and does *not* use knowledge of week 8 when selecting users). Over the 8 week period, the selection process resulted in around 155K user profiles over 10.4M sessions with an average of 174.40 (σ=181.49) queries/user and 2.61 queries/session (σ=3.36). All MAP results are means of performance across the five folds.

4.3 Experiments

Using the described dataset, we train a ranking model using the LambdaMART learning algorithm [35] for re-ranking the top ten results of the query. LambdaMART is an extension of LambdaRank [3] based on boosted decision trees. LambdaMART has been shown to be one of the best algorithms for learning to rank. Indeed, an ensemble model in which LambdaMART rankers were the key component won Track 1 of the 2010 Yahoo! *Learning to Rank Challenge* [6]. However, we note the choice of learning algorithm is not central to this work, and any reasonable learning to rank algorithm would likely provide similar results.

We use LambdaMART with 500 decision trees. We did a grid search using cross-validation by user over a 5 percent sample of the *training* set using the Union feature set. We used a range of number of leaves ∈ {35, 70, 140, 280, 560}, minimum instances in a leaf node ∈{200, 400, 800, 1600, 2000}, learning rate ∈ {0.075, 0.15, 0.30, 0.60}, and number of trees in the ensemble ∈{50, 100, 200, 400}. There was relative insensitivity in the area where number of leaves ≤ 100, learning rate ≤ 0.3, minimum instances in a leaf node ≤ 2000, and number of trees ∈ [50,200]. We broke ties arbitrarily and used number of leaves = 70, minimum instances in a leaf node = 2000, learning rate = 0.3, and number of trees = 50. After sweeping to determine parameters, we used the same parameters for all other models. We did this because we wish to understand how each model/feature set behaves in combination with the other parameters. For each fold, 10% of the training set is used as validation for model selection.

5. RESULTS

In this section we present the findings of our analysis. We focus on comparing the performance of the models constructed using the four conditions listed in Section 4.1. We begin by describing the overall performance of the models across all queries, then focus only on queries for which there is a measurable difference in search performance (e.g., where MAP changes) since those more clearly illustrate performance differences (although mask coverage effects, which we also explore). We report the change in performance from the baseline ranking – a highly competitive top Web search engine. We also conducted paired t-tests to compare the performance of the models with each other and the baseline.

5.1 Overall Performance

We begin by analyzing the overall performance of the models over the baseline. We measured the change (difference) in MAP from the baseline non-personalized search engine ranking's MAP across all queries for each of the four experimental conditions. For proprietary reasons, we only report the change from the baseline's MAP, rather than reporting absolute performance.

5.1.1 Performance on All Queries

Figure 4 presents the change in overall MAP for each model from the baseline. Error bars denote standard error of the mean in this chart and all other charts in the remainder of the paper.

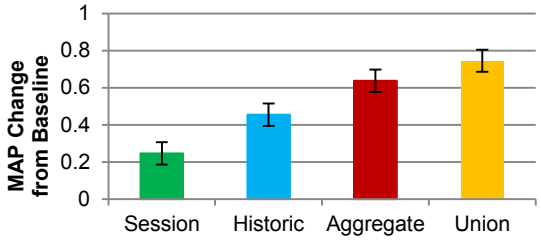

Figure 4. Average change in MAP from baseline ranker MAP

As seen in Figure 4, all methods improve over the baseline (i.e., all reported changes are positive). All gains over the baseline and differences between methods are significant with paired t-tests at $p < .01$. The figure shows increasing amounts of profiling information leads to greater improvements in retrieval performance. Interestingly using all sets of features (*Union*) and allowing the ranker to learn how the time views should be used for each query leads to the largest improvements over the baseline. This suggests that how personalization should be applied may be more nuanced than simple aggregation can capture (e.g., there may be times when we want to ignore historic data for personalization if the task is atypical of the user). We emphasize that while greater profiling information leads to better performance, this is not a foregone conclusion. For example, if every session was a new task unlike anything a user has previously done, then we would expect *Session* to outperform *Historic*. Likewise, if simple combinations of long- and short-term data could fully summarize the information available, then *Union* would not outperform *Aggregate*.

5.1.2 Effect of Query Position in Session

During search sessions, searchers reformulate their queries and adapt their information needs based upon exposure to information. We wanted to study whether this had an effect on the performance of the models. Figure 5 shows the average change in MAP from the baseline, broken out by the query position in the session, from the first to the fifth query and all remaining queries thereafter.

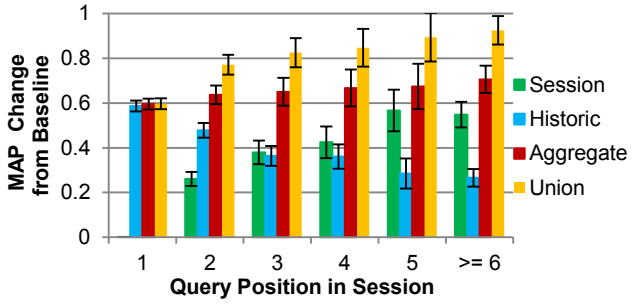

Figure 5. Avg. change in MAP by position of query in session.

From Figure 5, we see the *Session*-based personalization steadily increases its gains as more session information becomes available and seems to stabilize around 0.55 gain in MAP. On the other hand, *Historic* quickly decreases its gains as the information from the current session is not captured and the searcher's immediate interests may not be reflected in their long-term interests. It is interesting to note that by the fifth query, the session information accounts for half the gains in personalization (*Aggregate* vs. *Historic*). Looking at *Union* we can see that allowing the ranker to learn how to combine short- and long-term history leads to the best gains versus simple aggregation of the two profile sources. Note that the gains of *Union* over the other models are significant at $p < .01$ across all queries in the session except the first. For the first query in a session, the three models incorporating historic information have access to the same information, and as expected, their performance is nearly statistically identical. Overall this implies that re-finding and personalization are likely qualitatively different in the short term versus long term – otherwise simple aggregation would likely capture the behavior.

5.2 Performance When Measurably Different

Until now, we have focused on the effect on overall retrieval performance. However, since personalization may only be appropriate for a subset of queries, a more sensitive measure of the relative performance of the temporal views is to consider only performance on queries where personalization resulted in a different MAP score (i.e., non-zero difference) than the baseline – which we call *measurably different* queries for brevity. This provides important insight into the relative precision that each personalization method has independent of the fraction of queries that they affect. We present results from that analysis in this section.

5.2.1 All Measurably Different Queries

We begin by examining the change in MAP for the measurably different queries. Figure 6 shows the average change for each of the four profiling methods.

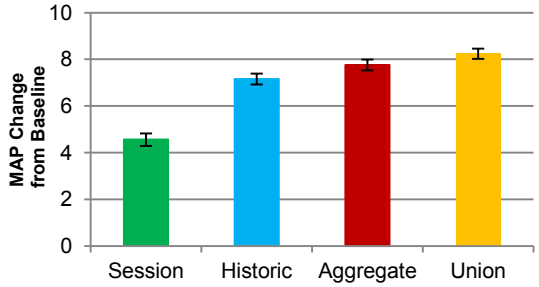

Figure 6. Average change in MAP given measurably different.

The figure shows that the gains are much larger when we focus on the measurably different queries (e.g., for *Session*, the MAP gain is more than 4 vs. 0.2 in Figure 4). The relative ordering of the sources is the same as presented in Figure 4, although the performance of the three models which incorporate historic information are more similar between themselves and more noticeably different from *Session* than when looking at changes in overall MAP. Differences are significant with paired *t*-tests at $p < .01$ for all comparisons apart from the *Aggregate* versus *Union* comparison, which is significant at $p < .10$.

5.2.2 Effect of Query Position in Session

In a similar way to earlier, we also study the effect of query position in session on the change in MAP for the measurably different queries. Figure 7 shows the gain over the baseline as the session proceeds. We see the gains per query are quite large, ranging from 3 points to 13 points of MAP. Several observations can be made from these differences. Again we see that the gain from *Session* increases then plateaus and that the gain from *Historic* decreases as the session progresses. Differences between *Session* and *Historic* are significant at $p < .01$ apart from at queries 3 and 4 where the gains in MAP that they offer cross.

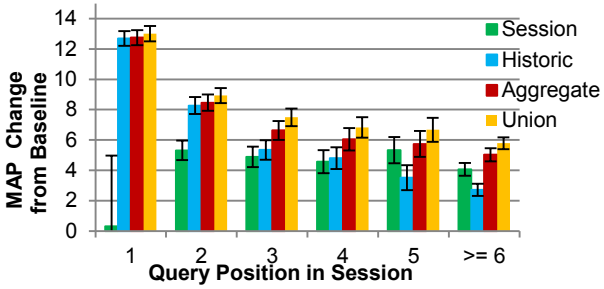

Figure 7. Average change in MAP on measurably different queries for each temporal view vs. position of query in session.

Next, we see that the gains of *Aggregate* and *Union* over measurably different queries are comparable in all positions (*Union* is better but generally not significant, other than at queries 3 and 5 ($p < .05$)). Since *Union* outperforms *Aggregate* overall (see Section 5.1.1) this implies the query volume impacted is the key difference (also see Figure 9) and suggests that being able to differentially weight short-term and long-term behavior can personalize different sets (i.e., it is not simply a gain in precision over the same queries).

Finally, the large increase in MAP in the first position for the methods using long-term information suggests that more ambiguous queries, which personalization most benefits [11][30], may be more likely to happen as the first query in the session. This intuitively makes sense since much session behavior is captured by refinement and reformulation after the first query.[1] In Figure 8 we examine several important properties of the queries in each position in the session to help us interpret such differences. We report Z-scores to highlight how the queries in each position differ from the mean of all positions.

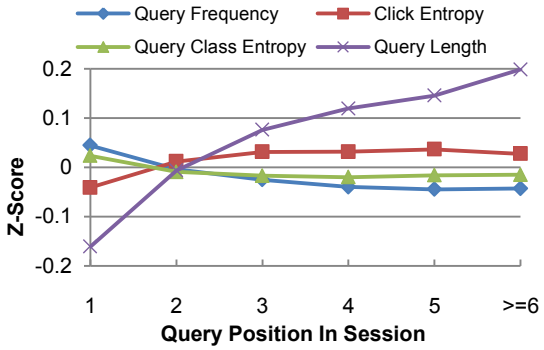

Figure 8. Mean of query properties for queries in a position normalized by their overall mean and standard deviation. Below/above zero is below/above average. Y-axis units are standard deviations (represented as the Z-score).

[1] The small gains with large variance of *Session* in position one results from a small number of queries that improve from the non-personalization features (e.g., query frequency, length) that are always available. It is clear from the error bars, that this is non-significant.

From Figure 8, we see the first position consists of very short queries that are frequent and have lower than average click entropy with higher than average topic ambiguity. Perhaps this is the reason why long-term profiles are so successful in the first position since they can frequently provide disambiguation. The most clear trend observable in the graph is for query length. Typically short queries start a session but the query length increases dramatically later in the session. While the low click entropy at the first position may seem contradictory to having ambiguous queries at the start of a session, this is an effect of the frequency of short sessions containing primarily navigational intents. For example, if we focus simply on sessions of length five or more, the z-score of the click entropy in the first position of such long sessions is 0.04 – well above the average. The decrease in ambiguity and increase in complexity (length) later in the session may explain why observed gains in MAP across the impacted queries are lower as we extend further into the session.

5.2.3 Query Volume Impacted by Personalization

Since personalization is not applied to all queries, it is important to understand the fraction of queries that can be affected by each of the time views. Overall the query volume impacted (percentage of measurably different queries) for the methods ranges from 5.42% (*Session*) to 9.05% (*Union*). However, there are some interesting effects when we consider the position of the query in the session. Figure 9 presents the fraction of queries with impacted performance at different query positions in the session.

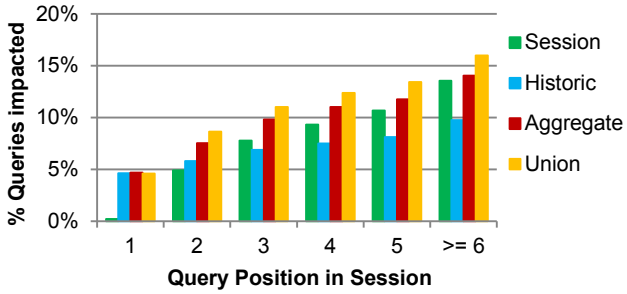

Figure 9. Percentage of queries with performance different than baseline vs. position of query in session.

In examining the impact percentages reported in Figure 9, we see that all of the methods personalize more often later in the session. This rate of impact rises most rapidly for the *Session* personalization as more information becomes available to it. Interestingly, impact rate also rises for the *Historic* method even though that method does not incorporate information from the current session. This suggests that long sessions may be *a priori* more likely to come from a user's prototypical interests although this observation requires further study. Changes in impact rate as the session proceeds may indeed be related to changes in query properties such as query length and popularity as the session proceeds and searchers' needs become more narrowly focused. As we saw in Figure 8, properties of the queries change over the course of the search session, and these changes may affect the percentage of queries that are amenable to personalization.

5.3 Effects of Other Conditions

As well as measuring the effect of time view on the performance of the model, we also explored the effects of other conditions mentioned in Section 3. Given the representation, one possibility is that the uniform weight or the decay weighting is crucial. For all of the features that use a weight we ran an ablation study that only used uniform weighted features, decay-weighted, or both and we observed no significant differences (not shown for space).

It is also interesting to consider whether these temporal aspects that happen using all features (*All*) are due only to topical effects (*Topic*) or to URL effects (*URL*). We summarize these conditions in Figure 10, normalizing each of them by the maximum performance within condition to focus on the temporal trend within condition.

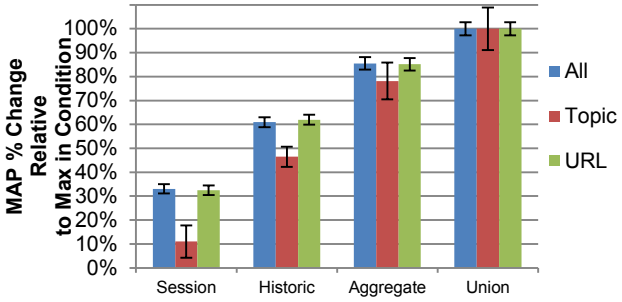

Figure 10. Changes in performance for other model conditions relative to the best performing time view and condition pair.

We see the same trends observed overall also hold within each of these conditions. This implies that the tradeoffs between short- and long-term aspects are not simply due to effects seen under one representation but are more general across representation choices. Both for space and because our emphasis here is on temporal aspects, we do not focus on comparing performance across conditions and address that in future work.

6. DISCUSSION AND IMPLICATIONS

We have studied how short-term (session) behavior and long-term (historic) behavior interact, and how each may be used in isolation or in combination to optimally contribute to gains in relevance through search personalization. Through a large-scale analysis of search logs, we have shown: that historic behavior provides substantial benefits at the start of a search session; that short-term session behavior contributes the majority of gains in an extended search session; and that the combination of session and historic behavior outperforms either using just session behavior alone or using simple aggregations. Importantly, we also showed that by learning a combination of the three views, the model can determine which should be weighted most highly given the current query and prior behavior. These results have important implications for search personalization, which has typically only studied short- or long-term behaviors independently. We show there is value from carefully considering interactions between them.

We observe that over the course of a search session there are variations in query properties such as query length and query ambiguity. These query changes may also affect the potential for personalization. As the session proceeds, we observe that more queries can be impacted by personalization since there may be more evidence of searcher interests resulting in increasing overall utility, but smaller gains in retrieval effectiveness for queries that do change, perhaps because the queries are more difficult.

Our research improves the understanding of how short- and long-term behaviors can be used by search engine designers to improve the performance of search personalization. For example, based on the position of the current query in a session, the search engine could use a particular source (e.g., historic data early in the session, session data as the session proceeds). However, we have also shown we can learn a model that can outperform any source in isolation and appropriately choose behaviors from each time view to attain better retrieval performance. Future work will extend this

research, including actions beyond clicks (e.g., skips) and other query similarity measures. We will also quantify the impact of short-term modeling for the cold-start problem of providing personalization to users with no previously observed interactions.

7. CONCLUSIONS

Previous work in search personalization has leveraged short-term or long-term behaviors to construct models of searcher interests. However, little is known about how these behaviors interact and when we should be leveraging them separately or in combination. In this paper we investigated the interaction between short- and long-term interests, and how this information can be combined to learn performant search personalization models. We demonstrated the benefits of historic behavior: at the outset of a session, short-term models yield benefit as the session proceeds; and allowing the ranker to learn weights for short-term features, long-term features, and their combination models searcher interests more effectively. This work makes an important step toward unifying prior work on personalization. Future work will explore further aspects of the interaction between different feature durations as well as other conditions from the framework that we only touched on briefly here (e.g., topic vs. URL), all focusing on how best to improve search performance through personalization.

8. ACKNOWLEDGMENTS

We would like to thank Dan Schwartz, Sebastian de la Chica, and Yi Mao for contributing to early prototypes of the research here. We would also like to thank Filip Radlinski for discussions around the implications for session-based personalization.

REFERENCES

[1] Bennett, P.N., Radlinski, F., White, R.W., and Yilmaz, E. (2011). Inferring and using location metadata to personalize web search. *SIGIR*, 135–144.

[2] Bennett, P., Svore, K., and Dumais, S. (2010). Classification-enhanced ranking. *WWW*, 111–120.

[3] Burges, C.J.C., Ragno, R., and Le, Q.V. (2006). Learning to rank with non-smooth cost functions. *NIPS*, 193–200.

[4] Cao, H., Jiang, D., Pei, J., He, Q., Liao, Z., Chen, E., and Li, H. (2008). Context-aware query suggestion by mining click-through and session data. *KDD*, 875–883.

[5] Cao, H., Hu, D.H., Shen, D., Jiang, D., Sun, J.-T., Chen, E., and Yang, Q. (2009). Context-aware query classification. *SIGIR*, 3–10.

[6] Chapelle, O., Chang, Y., and Liu, T.-Y. (2010). The Yahoo! learning to rank challenge. *http://learningtorankchallenge.yahoo.com.*

[7] Chirita, P., Nejdl, W., Paiu, R., and Kohlschutter, C. (2005). Using ODP metadata to personalize search. *SIGIR*, 178–185.

[8] Collins-Thompson, K., Bennett, P.N., White, R.W., de la Chica, S., and Sontag, D. (2011). Personalizing web search results by reading level. *CIKM*, 403–412.

[9] Daoud, L. Tamine-Lechani, L., Boughanem, M., and Chebaro, B. (2009). A session based personalized search using an ontological user profile. *SOC*, 1732–1736.

[10] Downey, D., Dumais, S., Liebling, D. and Horvitz, E. (2008). Understanding the relationship between searchers' queries and information goals. *CIKM,* 449–458.

[11] Dou, Z., Song, R. and Wen, J-R. (2007). A large-scale evaluation and analysis of personalized search strategies. *WWW*, 581–590.

[12] Fox, S., Kuldeep, K., Mydland, M., Dumais, S., and White, T. (2005). Evaluating implicit measures to improve Web search. *ACM TOIS, 23(2)*: 147-168.

[13] Gao, J., Yuan, W., Li, X., Deng, K., and Nie, J-Y. (2009). Smoothing clickthrough data for web search ranking. *SIGIR*, 355–362.

[14] Gauch, S., Chaffee, J., and Pretschner, A. (2003). Ontology-based user profiles for search and browsing. *WIAS*, 219–234.

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

[16] Kanoulas, E., Carterette, B., Clough, P., and Sanderson, M. (2010). Session track overview. *TREC 2010*.

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

[18] Li, L., Yang, Z., Wang, B. and Kitsuregawa, M. (2007). Dynamic adaptation strategies for long-term and short-term user profile to personalize search. *APWeb/WAIM*, 228–240.

[19] Ma, Z., Pant, G., and Sheng, O. (2007). Interest-based personalized search. *ACM TOIS*, *25(1)*: Article 5.

[20] Matthijs, N. and Radlinski, F. (2011). Personalizing web search using long term browsing history. *WSDM*, 25–34

[21] Mihalkova, L. and Mooney, R. (2009). Learning to disambiguate search queries from short sessions. *ECML*, 111–127.

[22] Piwowarski, B. and Zaragoza, H. (2007). Predictive user click models based on click-through history. *CIKM*, 175–182.

[23] Shen, X., Dumais, S.T., and Horvitz, E. (2005). Analysis of topic dynamics in Web search. *WWW*, 1102–1103.

[24] Shen, X., Tan, B., and Zhai, C. (2005). Context-sensitive information retrieval using implicit feedback. *SIGIR*, 43–50.

[25] Sontag, D., Collins-Thompson, K., Bennett, P.N., White, R.W., Dumais, S.T., and von Billerbeck, B. (2012). Probabilistic models for personalizing web search. *WSDM*, 433-442.

[26] Speretta, M. and Gauch, S. (2005). Personalizing search based on user search histories. *WI*, 622-628.

[27] Sugiyama, K., Hatano, K., and Yoshikawa, M. (2004). Adaptive web search based on user profile constructed without any effort from users. *WWW*, 675–684.

[28] Tan, B., Shen, X. and Zhai, C. (2006). Mining long-term search history to improve search accuracy. *KDD*, 718–723.

[29] Teevan, J., Dumais, S.T., and Horvitz, E. (2005). Personalizing search via automated analysis of interests and activities. *SIGIR*, 449–456.

[30] Teevan, J., Dumais, S.T., and Liebling, D.J. (2008). To personalize or not to personalize: modeling queries with variation in user intent. *SIGIR*, 163–170.

[31] Teevan, J., Liebling, D., and Geetha, G.R. (2011). Understanding and prediction personal navigation. *WSDM*, 85–94.

[32] White, R.W. and Drucker, S.M. (2007). Investigating behavioral variability in Web search. *WWW*, 21–30.

[33] White, R.W., Bailey, P., and Chen, L. (2009). Predicting user interests from contextual information. *SIGIR*, 363–370.

[34] White, R.W., Bennett, P.N. and Dumais. S.T. (2010). Predicting short-term interests using activity-based search context. *CIKM*, 1009–1018.

[35] Wu, Q., Burges, C. J.C. Svore, K. M., and Gao, J. (2008). Ranking, boosting, and model adaptation. *Microsoft Research Technical Report MSR-TR-2008-10.*

[36] Xiang, B., Jiang, D., Pei, J., Sun, X., Chen, E., and Li, H. (2010). Context-aware ranking in Web search. *SIGIR*, 451–458.

[37] Zhong, F., Wang, D., Wang, G., Chen, W., Zhang, Y., Chen, Z., and Wang, H. (2010). Incorporating post-click behaviors into a click model. *SIGIR*, 355–362.