Predicting Short-Term Interests Using Activity-Based Search Context

Ryen W. White
Microsoft Research
One Microsoft Way
Redmond, WA 98052 USA
ryenw@microsoft.com

Paul N. Bennett
Microsoft Research
One Microsoft Way
Redmond, WA 98052 USA
pauben@microsoft.com

Susan T. Dumais
Microsoft Research
One Microsoft Way
Redmond, WA 98052 USA
sdumais@microsoft.com

ABSTRACT

A query considered in isolation offers limited information about a searcher's intent. Query context that considers pre-query activity (e.g., previous queries and page visits), can provide richer information about search intentions. In this paper, we describe a study in which we developed and evaluated user interest models for the current query, its context (from pre-query session activity), and their combination, which we refer to as *intent*. Using large-scale logs, we evaluate how accurately each model predicts the user's short-term interests under various experimental conditions. In our study we: (i) determine the extent of opportunity for using context to model intent; (ii) compare the utility of different sources of behavioral evidence (queries, search result clicks, and Web page visits) for building predictive interest models, and; (iii) investigate optimally combining the query and its context by learning a model that predicts the context weight for each query. Our findings demonstrate significant opportunity in leveraging contextual information, show that context and source influence predictive accuracy, and show that we can learn a near-optimal combination of the query and context for each query. The findings can inform the design of search systems that leverage contextual information to better understand, model, and serve searchers' information needs.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval – *search process, information filtering.*

General Terms

Algorithms, Experimentation, Human Factors, Measurement.

Keywords

Search context, short-term interests, interest models.

1. INTRODUCTION

Search behavior resides within an external *context* that motivates the problem situation and influences interaction behavior for the duration of the search session and beyond [14]. Satisfying searchers' information needs involves a thorough understanding of their interests expressed explicitly through search queries, or implicitly through search engine result page (SERP) clicks or post-SERP browsing behavior. The information retrieval (IR) community has theorized about context [14], developed context-sensitive search models (e.g., [24][26]), and performed user studies investigating the role of context in the search process (e.g., [18]).

Most IR systems assume that queries are context-independent. This abstraction is necessary in Cranfield-style evaluations where relevance judgments are gathered independent of any user or interaction context [31]. In larger operational systems such as Web search engines, scale constraints have often favored simple context-independent approaches. Recent research suggests that this may be changing as log data and machine learning techniques are applied to model *activity-based context* (i.e., context gleaned from prior user interactions) in applications such as query suggestion [7], query classification [8], Web page recommendation [30], and Web search result ranking [32]. However, this research is often specific to particular applications, and an assessment of the value of modeling activity-based context that is applicable to a broad range of search and recommendation settings is required.

In this paper we describe a systematic study of the value of contextual information during Web search activity. We construct interest models of the current query, its context comprising preceding session activity such as previous queries or previous clicks on search results, the combination of the query and its context (called *intent*), and evaluate the predictive effectiveness of these models using future actions. Figure 1 illustrates each of the models and their role in representing users' interests. Queries are depicted as circles and pages as rectangles. The current query is $q3$.

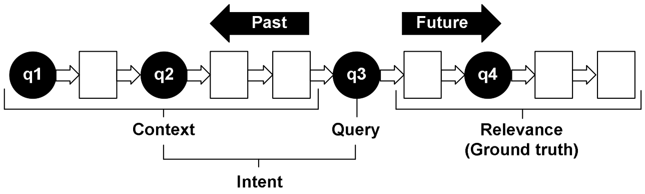

Figure 1. Modeling search context and short-term future interests within a single search session. User is at $q3$.

Accurate understanding of current interests and prediction of future interests are core tasks for user modeling, with a range of possible applications. For example, a query such as *[ACL]* could be interpreted differently depending on whether they previous query was *[knee injury]* vs. *[syntactic parsing]* vs. *[country music]*. This contextual knowledge could be used to re-rank search results, classify the query, or suggest alternative query formulations. Similarly, an accurate understanding of current and future interests could be used to dynamically adapt search interfaces to support different tasks. In our study we: (i) determine the fraction of search engine queries for which context could be leveraged, (ii) measure the value of different models and sources for predicting future interests, and (iii) investigate learning the optimal combination of query and context on a per-query basis, and use the learned models to improve the accuracy of our predictions. We use a log-based methodology as logs contain behavioral evidence at scale and cover many classes of information needs. This is important since performance differences may not hold for all search tasks.

CIKM'10, October 26–30, 2010, Toronto, Canada.

The remainder of this paper is structured as follows. In Section 2 we present related work on implicit profile generation from user activity, on representing interests with topical categories, on query analysis, and on the development of predictive interest models. Section 3 describes how we define and construct the models developed for this study. We describe the study and the findings from our analysis in Section 4. We discuss findings and their implications in Section 5 and conclude in Section 6.

2. RELATED WORK

Although most search systems match user queries to documents, independent of the interests and activities of the searcher, there is a growing body of work examining how knowledge of a searcher's interests and search context can be used to improve various aspects of search (e.g., ranking, query suggestion, query classification). User interests can be modeled using different sources of profile information (e.g., explicit demographic or interest profiles, or implicit profiles based on previous queries, search result clicks, general browsing activity, or even richer desktop indices). Profiles can be based on long-term patterns of interaction, or on short-term session-level patterns. Here we review prior research that examines the use of short-term implicit profiles generated using user's searching and browsing actions (queries, clicks on search results, and subsequent navigation to other pages). Further, we focus primarily on research that has used topical categories, such as the human-generated Web ontology provided by the Open Directory Project (ODP, dmoz.org), to model user interests since this provides a consistent representation for queries and Web page visits.

Several research groups have investigated personalizing search results using user profiles that consist of ODP-like topic categories. In an early personalized Web search system, *Outride*, Pitkow et al. [21] used the ODP classes associated with browser favorites and the last 1000 unique pages to represent a user's interests. They used this representation to both modify queries and re-rank search results. Gauch et al. [13] learned a user's profile from their browsing history, Speretta and Gauch [25] built profiles using just the search history, and Chirita et al. [9] and Ma et al. [19] used profiles that user's specified explicitly. In all cases, interest profiles were compared with those of search results and used to alter the order in which results were presented to individuals. Personal profiles have also been used to create personalized versions of the *PageRank* ranking algorithm for setting query-independent priors on Web pages [16]. Such personalized document priors can be combined with content match features between queries and Web pages for the improved ranking of Web search results. More recently, Bennett et al. [5] demonstrated how category-level features can be used to improve search ranking in the aggregate. In that work categories were not used to represent user profiles, but rather to propagate user behavior for a small number of specific URLs that are clicked on for a query to a much larger set of URLs that belong to the same topical category. Similarly, Xiang et al. [32] developed heuristics to promote URLs with the same topical category if successive queries in a search session were related by general similarity, and were not specializations, generalizations, or reformulations of the previous query.

Using the context of user activities within a search session has also been used to improve query analysis. Short queries are often ambiguous, so researchers have used previous queries and clicks in the same session to build a richer models of interests and improve how the search system interprets users' information needs. Cao et al. [7][8] represented search context by modeling sessions as sequences of user queries and clicks. They learned sequential prediction models such as hidden Markov models and conditional random fields from large-scale log data, and applied the models to query suggestion, query categorization, and URL recommendation. Mihalkova and Mooney [20] used similar search session features to disambiguate the current query. Although neither of these studies uses ODP categories, they are session-based and illustrate non-ranking applications of search session information.

Finally, several research groups have developed predictive models of user interactions within search sessions. Early Web recommendation systems such *WebWatcher* suggested new Web pages for individuals based on their recently-viewed pages [3]. Shen et al. [23] learned probabilistic models to predict the class (from top-level ODP categories)of the next URL a searcher would click on. They compared models for individuals, groups, and aggregate search behavior patterns using long-term interaction patterns. The best predictive accuracy was obtained with individual or group models, but they did not explore richer combinations. Piwowarski and Zaragoza [22] explored three different predictive click models based on personalized and aggregate click information in a Web search setting, trying to predict relationships between queries and clicked documents. They built a probabilistic user-centric model, a group model, and a global model, and a model that combined all three. The best of their models was able to achieve either moderate prediction accuracy (50% of the clicks) with high recall (75% of the time), or very high accuracy of 98% but low recall (5% of the time). White et al. [30] compared the value of a variety of sources of contextual information for predicting future interests at different time scales (one hour, one day, or one week). The best accuracy was obtained when recent actions (the so-called *interaction context*) were used to predict interests in the following hour. Bailey et al. [4] focused on assigning ODP labels to long and rare (previously-unseen) Web search engine queries. They used labels assigned to pages on browse trails extracted from toolbar logs following seen queries and matched the unseen queries to them.

The research presented in this paper differs from the previous work reviewed above in several important ways. First, we examine contexts from several sources: queries, URLs visited (from search engine results pages and in subsequent Web browsing), and learn optimal source combinations. Second, we focus on developing models capable of accurately predicting the future interests in a search session, and explore precision/coverage tradeoffs. Such predictive models can be used for a wide variety of applications, including supporting pro-active changes to the interface to emphasize results of likely interest or to suggest contextually-relevant query alternatives, as well as more traditional applications to ranking and filtering. Finally, we base our analyses on a large set of user searching and browsing sessions obtained from log data, thus addressing scale and representativeness issues.

3. MODELING SEARCH ACTIVITY

The scenario that we model in this investigation is that of a Web searcher who has just issued a query to a search engine. It is at this point that the engine may leverage the recent search activity of the user to augment the current search query with a more sophisticated representation of their search intent. Important questions around which types of search activity sources can be used to build contextual models—as well as how and when to combine these sources—are answered in the study presented in this paper. We begin by describing the data used to both model search activity and evaluate the predictive performance of the models.

3.1 Data

The primary source of data for this study was a proprietary data set containing the anonymized logs of URLs visited by users who consented to provide interaction data through a widely-distributed browser plugin. The data set contained browser-based logs with both searching and browsing episodes from which we extract search-related data. These data provide us with examples of real-world searching behavior that may be useful in understanding and modeling search context. Log entries include a timestamp for each page view, and the URL of the Web page visited. To remove variability caused by geographic and linguistic variation in search behavior, we only include log entries generated in the English-speaking United States locale. The results described in this paper are based on URL visits during the last week of February 2010 representing billions of Web page visits from hundreds of thousands of unique users. From these data we extracted *search sessions* on the Bing Web search engine, using a session extraction methodology similar to [29]. Search sessions begin with a query, occur within the same browser and tab instance (to lessen the effect of any multi-tasking that users may perform), and terminate following 30 minutes of user inactivity. We use these browser-based logs rather than traditional search-engine logs since they provide access to all pages visited in the search session preceding and succeeding the search query, information that is important for our later analyses. The median session length was 19 actions (queries and Web page views) (mean=29 actions). The median duration of a search session was 10 minutes 9 seconds (mean=8 minutes 32 seconds). To augment the browser-based logs, we also mined Bing search engine logs to obtain the URLs of the top-ten search results returned for each query (to build query models).

3.2 ODP Labeling

We represented context as a distribution across categories in the ODP topical hierarchy. This provides us with a consistent topical representation of queries and page visits from which to build our models. ODP categories can also be effective for reflecting topical differences in the search results for a query [5] or a user's interests [30]. Given the large number of pages present in our log data, we used automatic classification techniques to assign an ODP category label to each page. Our classifier assigned one or more labels to the pages based on the ODP using a similar approach to Shen et al. [23]. In this approach, classification begins with URLs present in the ODP and incrementally prunes non-present URLs until a *match* is found or *miss* declared. Similar to [23], we excluded pages labeled with the "Regional" and "World" top-level ODP categories, since they are location-based and are typically uninformative for constructing models of user interests. To lessen the impact of small differences in the labels assigned, we also filtered to only use 219 categories at the top two levels of the ODP hierarchy, referred to as L hereafter. The coverage of the resulting ODP classifier with URL back-off was approximately 60%. To improve the coverage of the classifier we combined it with a text-based classifier, described in [5], that uses logistic regression to predict the ODP category for a given Web page. When optimized for the $F1$ score in each ODP category, the text-based classifier has a micro-average $F1$ of 0.60. Predicted ODP category labels from this classifier were available for many pages in the Bing search engine index. For URLs where only one classifier had labels, the most frequent label (for ODP lookup) or the most probable label (for the text-based classifier) was used. For URLs where both classifiers had a label, the label was determined by first looking for an exact match in the ODP, then in the classified index pages, and then incrementally pruning the URL and checking for a category label in the ODP or in the classified index pages. This classifier combination led to coverage exceeding 80% across all URLs in our set. We did not attain 100% coverage since some URLs were in the hidden Web and not in the Bing index (e.g., because logins are required, the pages are dynamically generated, etc.).

3.3 Sources and Source Combinations

We use three sources to build models from search sessions:

1. *Query*: ODP labels automatically assigned to the top-ten search results for the query returned by the engine used in our study. Label assignment is described in more detail in Section 3.4.
2. *SERPClick*: ODP labels automatically assigned to the search results clicked by the user during the current search session.
3. *NavTrail*: ODP labels automatically assigned to Web pages that the user visits following a SERP click.

By examining the effectiveness of interest models built with these sources, we can help determine their relative value and provide insight into which sources are the most important for search engines to represent to attain good prediction performance. For example, queries and SERP clicks are easy for search engines to capture, but post-SERP capturing browsing behaviors involves the deployment of client-side software such as a Web browser plugin.

In this study, we experiment with building models of the search context for the current query using: (i) previous queries only; (ii) previous search engine activity only (previous queries and SERP clicks), and; (iii) all previous activity (previous queries, SERP clicks, and post-SERP Web page visits). We also compare each of these source combinations against the current query alone.

3.4 Model Definitions

Three models were constructed to represent users' short term interests: *query* (the current query), *context* (queries and Web pages viewed prior to the current query), and *intent* (a weighted combination of current query and context). The sequence of actions following the current query in the session is used to develop the relevance model used as ground truth. Note that the "models" are different from the "sources" described in the previous section. The sources determine the information used in building the models. The decision about which sources are used in constructing the models can be made based on availability (e.g., search engines may only have access to queries and SERP clicks) and/or desired predictive performance (more sources may lead to more accurate models, but may also contain more noise if searchers deviate from a single task). All models represent user interests as a probability distribution across the ODP labels in L. In the remainder of this subsection we provide more details on each of the models.

3.4.1 Query Model (Q)

Given the method for assigning ODP category labels to URLs, we assigned labels to a query as follows. For each query, we obtain the category labels for the top-ten search results returned by the Bing Web search engine at query time. Probabilities are assigned to the categories in L by using information about which URLs are clicked for each query. We first obtain the normalized click frequencies for each of the top-ten results from search-engine click log data gathered during all of 2009, and computed the distribution across all ODP category labels. Search results without click information are ignored in this procedure. ODP categories in L that are not used to label top-ranked results are assigned the prior probabilities for query models, as described in Section 3.5.

3.4.2 Context Model (X)

The context model is constructed based on actions that occur *prior to the current query* in the search session. Actions comprise queries, Web pages visited through a SERP click, or Web pages visited on the navigational trail following a SERP click. A query model is created for each previous query in the context using the method described in the previous subsubsection. A model for each Web page is created using the ODP category label assigned via the strategy described earlier (i.e., first check for exact match in ODP, then check for exact match with logistic regression classifier, etc.). The weight attributed to the category label assigned to the page is based on the amount of time that the user dwells on the page. Dwell time has been used previously as a measure of user satisfaction with Web pages [1][11]. In a similar way, we assume that if a user dwells on a page for longer than 30 seconds, then the page contains useful content. However, instead of using a binary relevant/non-relevant threshold of 30 seconds we used a sigmoid function to smoothly assign weights to the categories. Function values ranged from just above zero initially to one at 30 seconds.

In addition to varying the probability assigned to the class based on page dwell time, we also assigned an exponentially-decreasing weight to each action as we move deeper into the context. That is, pre-query actions were weighted according to $e^{-(n-1)}$, where n represents the number of actions before the current query. A similar discount has been applied in previous work on ostensive relevance [6]. Using this function, we could assign the action immediately preceding the current query a weight of one and downweight the importance of all preceding session actions, such that more distant events received lower weights. This is supported by previous work which suggests that the most recent action is most relevant for predicting the next action (e.g., [10][23]). All page and query models in the context had their contribution toward the overall context model weighted based on this discount function. All of these models were merged and their probabilities normalized so that they summed to one (after priors were assigned to unobserved categories). The resultant distribution over the ODP category labels in L represents the user's context at query time.

3.4.3 Intent Model (I)

The intent model is a weighted linear combination of the query model (for the current query) and the context model (for the previous actions in the search session). Since this model includes information from the current query and from the previous actions, it can potentially provide a more accurate representation of user interests than the query model or the context model alone. The intent model is defined as:

$$I(w) = wX + (1 - w)Q, \text{ where } w \in [0,1] \quad (1)$$

where I, X, and Q represent the intent, context, and query models respectively, and w represents the weight assigned to the context model. When combining the query and context models to form the intent model, by default w = 0.5. However, as we will show, the optimal value of w varies per query and can be accurately predicted using features of the query and its activity-based context.

3.4.4 Relevance Model or Ground Truth (R)

The relevance model contains actions that occur *following the current query* in the session. This captures the "future" as shown in Figure 1 and represents the ground truth for our predictions. The relevance model comprises a probability distribution over L and is constructed in a similar way to the context model. The only difference between how the two models are built is that the relevance model considers future actions rather than past actions. In the relevance model, we weight the action immediately *following* the query—typically another query or a SERP click—most highly, and decrease the weight rapidly for each succeeding action in the session (using the same exponential decay function as the context model). This regards the next action as more important to the user than the other actions in the remainder of the session. This seems reasonable as the next action may be most closely related to their interests for the search query. We use this relevance model as the ground truth for measuring the accuracy of our predictions of short-term user interests and for learning the optimal combination of query and context for a query. User behavior has been shown to be a useful measure of search success in previous work [2][14]. Since the relevance model is automatically generated, it can be used to evaluate performance on a large and diverse set of queries, but may contain noise associated with variance in search behavior.

3.5 Assigning Model Priors

To handle missing values, each of the interest models was assigned a prior distribution across L based on ODP categories assigned to URLs in a held out set of 100,000 randomly-selected search sessions from our data set, hereafter referred to as S_P. In this set the number of sessions from any single user was restricted to ten. Limiting the number of sessions per user lessened the likelihood that highly-active users would bias our sample. Priors were tailored to the sources being used to construct each interest model. For example, the query models were always initialized with the query prior (generated from the ODP categories labels assigned across *all URLs appearing in search result lists* in S_P), whereas a context model's priors are based on all sources used to build the model (e.g., *all search engine activity* in S_P).

4. STUDY

We now describe our study, beginning with research questions.

4.1 Research Questions

Three research questions drove our investigation:

1. What fraction of search engine queries could be impacted by the use of context information?
2. What is the predictive accuracy of the user interest models generated from the current query, context, and intent? What is the effect of varying the source of context information on predictive accuracy?
3. Can we learn how best to combine query and context models?

In the remainder of this section we answer each question in turn, beginning with the extent of the opportunity offered by context.

4.2 Extent of Context Opportunity

We first investigate the potential opportunity of using short-term session context information for interpreting the current query. To do so we selected a random sample of one hundred thousand search sessions, hereafter referred to as S_T, from the data set described in Section 3.1 with the same ten-session-per-user limit as in S_P. In S_T, there were a total of 325,271 queries. Of these queries, 224,634 (69%) were not reached through session-level revisitation (e.g., through the browser "back" button). This is important in estimating the opportunity for using contextual information, since it focuses attention on those queries that users actually submitted. For queries that are repeated using the back button, search engines would probably not want to update search results when users returned to them. Indeed, previous work has shown that care should be taken when adapting the content or the ordering of

Table 1. Prediction accuracy of the models and the percentage of queries ***for which each model or source performed best.***

Context source	Accuracy ($F1$)			Percentage of queries best			
	Models			*Between models*			*Between sources (intent only)*
	Q	X	I	Q	X	I	
None (i.e., current query only)	0.39	–	0.39	100%	–	100%	15%
Query (i.e., all previous queries)	0.39	0.42	0.43	25%	18%	22%	19%
Query + *SERPClick* (i.e., all previous queries and result clicks)	0.39	0.46	0.46	30%	16%	25%	22%
Query + *SERPClick* + *NavTrail* (i.e., all previous actions)	0.39	0.50	0.49	34%	11%	30%	26%

search results for repeat queries as this could lead to user confusion and frustration [27]. We therefore focus on intentionally-issued search queries in the remainder of our analysis.

Around 40% of the sessions in S_T contained multiple queries and around 60% of queries had at least one preceding query in the search session, providing an opportunity to generate a context model. Given an ideal interest model construction method that could build query/context/intent models for all encountered contexts, we have opportunity to improve the search experience for a significant fraction (60%) of the search traffic that search engines receive.[1] This is important, since implementing support for context modeling *at scale* is a potentially costly undertaking for search engine companies, who must also consider the investment in infrastructure required to serve such context rapidly for many millions of search queries daily. The simple label assignment methods described in Section 3.2 cover almost 80% of all contexts, so we cover roughly 50% of all queries in our experiments.

4.3 Prediction Accuracy of Models & Sources

In the previous section we showed that context can cover a significant portion of search engine queries. In this section, we focus on how well each of our models – current query only (Q), context only (X), and intent (I) – predicts the future short-term interests of the current user, represented by the relevance model or ground truth (R). The advantage of using logs for this study is that we can observe future actions and construct the relevance model automatically from them. As part of the analysis, we also vary the source of information used to construct the context model: either previous queries, previous queries and SERP clicks, or all previous actions. In building the intent models in this section we set the context weight (w) to 0.5. For this analysis we used the same set of sessions used in the previous section (S_T). Parametric statistical testing is performed where appropriate with alpha set to .05.

To evaluate the predictive accuracy of the model sources we used the $F1$ measure. This measure computes the harmonic mean of precision and recall, and has been used successfully in a number of search scenarios including prior work on evaluating context sources [30]. We prefer $F1$ to alternatives such as Jensen-Shannon divergence since $F1$ is easily interpretable and makes model comparisons simpler. $F1$ is defined in the standard way as:

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{2}$$

where we define precision and recall as:

$$Precision = \frac{|L_M \cap L_R|}{|L_M|}, Recall = \frac{|L_R \cap L_M|}{|L_R|} \tag{3,4}$$

where L_M is the set of instantiated labels in the model being tested and L_R is the set of instantiated labels in the relevance model. Label instantiation occurs when category labels are assigned to a query or a Web page as described in Section 3.4. When either L_M or L_R are empty, precision and recall equal zero. We use $F1$ to measure the predictive accuracy of each of the query, context, and intent models. For each model, the recall depth is computed based on the number of categories which were instantiated in building the relevance model comprising all future session activity.

In order to compare the different models on the same queries, we focus on queries for which we could construct a model for the current query, its context, and its future (for relevance), for all context sources. This equated to 30% of the queries in S_T. We iterated over each query, and computed the models per the approach described earlier and calculated how accurately the models predicted the future (using the $F1$ score). The findings of this analysis across all queries in our set are summarized in Table 1. Note that the contribution of the query model (column Q) is the same for all sources because no context is used. The findings under "Accuracy" in the table suggest that: (i) leveraging any context (columns X and I) yields significant prediction accuracy gains over using no context (column Q), in terms of predicting the relevance model; (ii) leveraging more sources yields more accurate predictions (improved predictions for both X and I as more context sources are included), and; (iii) the context (X) and intent (I) models do not differ in terms of overall accuracy. All observed differences in the $F1$ scores between the three models were statistically significant using one-way analyses of variance (ANOVA) (all F(2,122154)=6.91, all p<.001).

In addition to comparing the overall $F1$ scores, we also computed the fraction of queries for which each model outperformed the other models and the fraction of queries for which each source outperformed the other sources. This analysis allows us to identify queries for which each of the models provides best performance. The percentage of queries for which each of the models and context sources wins is shown in Table 1 under "Percentage of queries best." Only queries where one model / context source was a clear winner are included under this heading in the table. Queries with ties, where all models failed to predict (around 10% of the sample used) or two of more models / context sources had the same $F1$ score (around 30% of queries) are not included. The columns Q, X, and I show the percent of queries for which that model for each context source – e.g., for the *Query* source, Q is best on 18% of queries, X on 25%, and I on 22%. The final column shows differences for sources for the intent model. Here we see that the richer the context the better: the *Query* source is best

[1] Note that it may also be possible to create contexts from preceding actions beyond search sessions (e.g., previous Web page visits alone). See the work of White and colleagues [30] for details.

on 19% of the queries, the *Query + SERPClick* source is best for 22% of queries, and all sources are best for 26% of queries.

At least two observations can be made from the findings. First, as expected, there are sets of queries for which each of the models performs best, and second, there are queries for which each of the model sources performs best. To understand more about which queries each of models performed best on we visually inspected a sample of queries and their sessions to understand why the model or source performed so well. The findings of our analysis revealed that in cases where the current query model (Q) wins, the query either has very specific search intent (e.g., *[espn]*, *[webmd]*, *[call of duty 4 demo]*), or is in situations where the query represents the first action after a noticeable shift in topical interests within the search session. In cases where the context model (X) wins, the query is a continuation of constant intent, the query is ambiguous (e.g., *[amazon]*) or is detached from the search session (e.g., query for *[facebook]* during a search session seemingly about Amazonian rainforests). In cases where the intent model (I) wins, there is typically a consistent intent throughout the session.

To detect shifts in interests, we computed the cross entropy (CE) of the query model versus the context model. Cross entropy is an information theoretic measure of the average number of bits needed to identify an event from a set of possibilities given two distributions. It has been used in previous work to compare distributions in order to make ranking decisions [5]. It is defined as:

$$CE(Q,X) = -\sum_{c} Q_c \log_2(X_c) \tag{5}$$

where Q and X represent the current query model and the context model respectively, and Q_c and X_c represent the probability assigned to each of the category labels (c) in the current query and context models. In the case of comparing query and context, the interpretation would be the number of bits on average to encode the class of query that is actually issued next given predictions regarding what class would be issued using the previous context.

In situations where the entropy between the query model and the context model is low, we would expect the query to be a continuation of the same intent (and perhaps more weight should be placed on the context). In cases where it is high, we would expect there to be a shift in intents (and perhaps more weight should be placed on the current query). The Pearson's correlation coefficient (r) between the cross entropy and the $F1$ scores for the context and the query models were 0.63 and 0.58 respectively, suggesting that difference between the query and the context may be important in determining when contextual information should be leveraged.

To gain insight into the breadth of interests associated with a query, we computed the click entropy of the query. Click entropy has been used in previous work [28] to gauge the variability in intents associated with a query, manifested as clicks on search results. Low click entropy for a query suggests that searchers generally target a single result URL. In contrast, a query with high click entropy indicates that a range of URLs are targeted. Interestingly, for queries in which the context and intent models outperformed the query models, the click entropy of the query was significantly higher (2.48 versus 2.12). This is consistent with the intuition that for ambiguous queries, knowing more about previous session context can be helpful. It also seems that query click entropy provides insight into when more weight should be placed on context, perhaps to help disambiguate search results of ambiguous queries. We will now explore varying the context weight for each query.

In our analysis thus far we have assumed that the weight assigned to the context in the intent model is always 0.5. However, as our findings in this section, and the findings of previous work in this area [32], have suggested, it is unlikely that the same context weights should be used for all queries. In the next section we present an investigation of whether we can automatically learn the optimal context weight on a per-query basis.

4.4 Learning Optimal Context Weights

To learn the optimal weight to assign to context when combining the context model and the query model we identified the optimal context weight (w) for each query on a held out training set, creating features for the query and the context that could be useful in predicting w, and then learning w using those features. In this section we cover all three of these steps, beginning with the optimization task. We also present findings on improvements in prediction accuracy obtainable by learning w, and results of prediction experiments using different model sources.

4.4.1 Determining the Optimal Context Weight

The goal of the optimization is to determine the context weight that minimizes the difference in distributions between the intent and the relevance models. To construct a set for learning, we assume therefore that we are given a set of queries with their context, query, and relevance models collected from observed session behavior. We first need to convert the knowledge of the future represented in the relevance model to an optimal context weight that we then use for training a prediction model. The function that we wish to minimize in this scenario is the cross-entropy, as defined in Equation 5, between the intent model and the relevance model. In this case, the reference distribution is the relevance model, and the cross-entropy takes its minimal value (the entropy of the relevance distribution) when the intent model distribution is equal to the relevance distribution. The objective function used is:

$$\begin{gathered}\min_{w}\left[-\sum_{c} R_c \log_2[I_c(w)]\right] + a\left(\log_2 \frac{w}{1-w}\right)^2 \\ \text{where } I_c(w) = wX_c + (1-w)Q_c \\ \text{s.t. } w \in (0,1)\end{gathered} \tag{6}$$

Here R_c , X_c, and Q_c represent the probability assigned to the cth category by relevance, context, and current query models, respectively. Similarly, $I_c(w)$ is the corresponding intent probability using w as the context weight. The first term in this equation is simply the cross-entropy between the relevance and intent distributions. The second term is a regularizer that penalizes deviations from w=0.5. It is essentially a Gaussian regularization applied after a logit transform (which is monotone in w and symmetric around w=0.5). The regularizer also has the negligible effect of constraining the optimum to lie in the open interval (0,1) instead of the closed interval [0,1]. After squaring then, the regularization term is convex. Since cross-entropy minimization is also known to be convex, for $a > 0$, the resulting problem is convex and can be minimized efficiently to find an optimal value of w. Besides keeping w closer to 0.5, the regularizer is helpful in that without it, small deviations in the distributions (e.g., due to floating point imprecision) can force the optimal weight to 0 or 1 although the value of the objective is essentially (near) flat. This adds a source of unnecessary noise to learning and is easily handled through regularization. For our experiments, we set $a = 0.01$, and further exploration of this parameter remains as future work.

Table 2. Features used in predicting optimal context weight. Log-based features for the query are *italicized.*

Feature	**Feature description**
Query class	
QueryLength	Number of characters in query
QueryWordLength	Number of words in query
AvgQueryWordLength	Average length of query words
AvgClickPos	Average SERP click position for query
AvgNumClicks	Average number of SERP clicks for query
AvgNumAds	Average number of advertisements shown on the SERP for query
AvgNumQuerySuggestions	Average number of query suggestions shown on the SERP for query
AvgNumResults	Average number of total search results returned for the query
AbandonmentRate	Fraction of times query issued and has no SERP click
PaginationRate	Fraction of times query issued and next page of results requested
QueryCount	Number of query occurrences
HasDefinitive	True if a single best result for the query is in the result set (usually for navigational queries)
HasSpellCorrection	True if search engine spelling correction is offered for query
HasAlteration	True if query is automatically modified by engine (e.g., stemming)
FracQueryModelNotPrior	Fraction of all categories in the query model that are instantiated
QueryEntropy	Entropy of the query model
ClickEntropy	Click entropy of query based on distribution of result clicks
QueryJensenShannon	Jensen-Shannon divergence between the query model and the previous query model in session
Context class	
NumActions	Number of queries and page visits (excludes current query)
NumQueries	Number of queries (excludes current query)
Time	Time spent in session so far
NumSERPClicks	Number of search results clicked
NumPages	Number of non-SERP pages visited
NumUniqueDomains	Number of unique domains visited
NumBacks	Number of session page revisits
NumSATDwells	Number of page dwells exceeding a 30-second dwell time threshold
AvgQueryOverlap	Average percentage query overlap between all successive queries
FracContextModelNotPrior	Fraction of all categories in the context model that are instantiated
LastContextWeight	Previous estimate of optimal context weight in the session. Note: Uses previous query model, previous context model, and actions between previous query and current query as relevance model (ground truth)
ContextEntropy	Entropy of the context model
ContextEntropyByNumAct	Entropy of the context model divided by the number of actions in session so far
ContextJensenShannon	Jensen-Shannon divergence between the context model and the previous context model in session
QueryContext class	
QueryContextCrossEntropy	Cross entropy between the query model and the context model
ContextQueryCrossEntropy	Cross entropy between the context model and the query model
JensenShannonDivergence	Jensen-Shannon divergence between the query model and the context model

To create a training set, we use the query, context, and relevance models to compute the optimal context weight per query by minimizing the regularized cross-entropy for each query independently. Note that the relevance model is implicitly the labeled signal which optimization converts to a "gold-standard" weight to be used in learning and prediction.

4.4.2 Generating Features of Query and Context

At query time, a search engine has access to a large number of features about the query and the activity-based search context that could be useful for learning the optimal context weight. Table 2 lists the features that were used in our predictions. Features were divided into three classes: *Query*, capturing characteristics of the current query and the query model, including log-based features based on search logs of the Bing search engine for the last week in January 2010, italicized in Table 2; *Context*, capturing aspects of the pre-query interaction behavior as well as features of the context models themselves, and *QueryContext*, capturing aspects of how the query model and context model compare.

The broad range of features used enabled us to capture many aspects of search activity. These features were generated for each session in our set and used to train a predictive model to estimate the optimal weight to be placed on the context when building the intent model. We now describe the experiments performed to evaluate our predictions of the optimal context weight.

4.4.3 Predicting the Optimal Context Weight

We used Multiple Additive Regression Trees (MART) [12] to train a regression model to predict the optimal context weight. MART uses gradient tree boosting methods for regression and classification. MART has several strengths, including model interpretability (e.g., a ranked list of important features is generated), facility for rapid training and testing, and robustness against noisy labels and missing values, that make it attractive for this task. We selected a new set of one hundred thousand search sessions with no overlap with the sessions used in the analysis presented previously in this paper (referred to as S_T). From these sessions we selected around 45,000 queries for which we could construct a query, context, intent, and relevance model. We used all search session activity (queries, SERP clicks, and post-SERP Web page visits) since models constructed from those sources yielded the best predictive performance in our earlier analysis and we wanted to see how well we could do given this rich source of information. We used 60% of those queries for training, 20% for validation, and 20% for testing, and performed ten-fold cross validation to improve result reliability. Note that in constructing the folds, we split by session so that all queries in a session are used for either training, validation, or testing. We do not allow queries from the same session to be used in different phases as this may bias our experiments. Pearson's correlation (r) and root mean squared error ($RMSE$) were used to measure our performance at predicting optimal w. Correlation measures the strength of association between predicted and actual on a scale from -1 to 1, with one indicating a perfect correlation and zero indicating no correlation. $RMSE$ is the square root of the mean of the squared deviations between the actual and predicted values and resides between zero and one. The ideal value of $RMSE$ is zero, with larger values showing more errors. Following our analysis, the average obtained r across all experimental runs was 0.85 and the average $RMSE$ across all runs was 0.15.[2] The weights assigned by the model to the top-15 features are shown in Table 3, normalized relative to the most predictive, *QueryContextCrossEntropy*.

Table 3. Feature importance.

Feature	Class	Importance
QueryContextCrossEntropy	QueryContext	1.00
JensenShannonDivergence	QueryContext	0.86
ContextEntropy	Context	0.69
ContextQueryCrossEntropy	QueryContext	0.46
ContextJensenShannon	Context	0.19
QueryEntropy	Query	0.18
LastContextWeight	Context	0.17
QueryCount	Query	0.14
NumActions	Context	0.12
FracQueryModelNonPrior	Query	0.12
FracContextModelNonPrior	Context	0.12
NumQueries	Context	0.11
NumSERPClicks	Context	0.05
QueryJensenShannon	Query	0.04
ClickEntropy	Query	0.04

From Table 3 it appears that the most performant features relate to the information divergence of the query models and the context models. This suggests that the strength of the relationship between the current query and the context is an important indicator of how much weight to assign to the context. Also important are features of the current query only and its context only.

In Figure 2 we plot the predicted and optimal context weight and show the line of best fit between them for a representative cross-validation run. Each point on the plot is a search query. From the figure it appears that we perform well when predictions place a large amount of weight on the context. Indeed, although the average true optimal w across all queries is 0.69, the average predicted w is 0.75, suggesting that our predictions may overweight context.

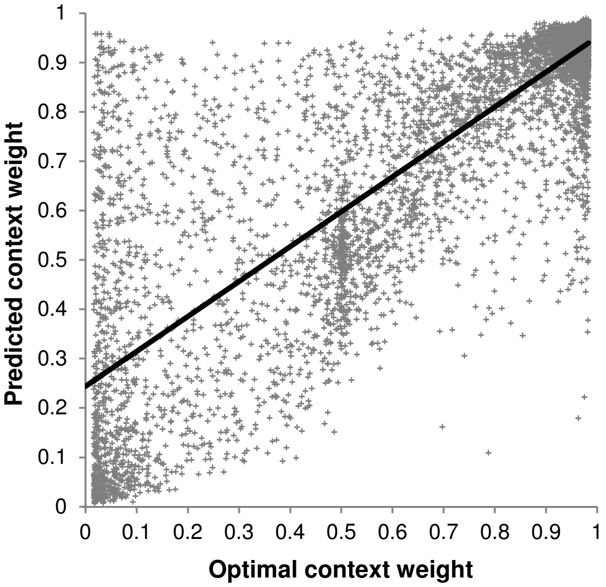

Figure 2. Predicted context weight vs. optimal context weight.

Prediction error was highest for those queries in the top-left corner of the figure. Inspection of the queries revealed that they were rare queries or contained typographical errors, which meant that we could not generate some of the log-based query features such as *QueryCount*, which from Table 3, appear to be important in our predictions. Prediction errors were also larger for queries at the beginning of search sessions, meaning that there was limited contextual information was available to build interest models.

In the next section we apply the estimates of the optimal context weight to the prediction of future interests tackled in Section 4.3.

4.4.4 Applying Optimal Context Weight Predictions

Given that we are able to predict the optimal context weight with good accuracy, we now investigate the impact on predictive accuracy of utilizing the predicted optimal context weight in the intent model when combining the query model and the context model. To do this, we used a model trained on all sessions in S_T, created

[2] For reference, the average performance of the model trained only on search engine interactions was r=0.75, $RMSE$=0.19.

a test set of 100,000 randomly-chosen search sessions (not in S_T or S_P and limited to ten sessions per user), and chose the optimal context weight for queries where could construct the models.

For each query in the test set, we then constructed the intent model with the predicted context weight (w) and computed the $F1$ score between the intent model and the relevance model for each query. In Table 4 we present the average $F1$ score using the per-query estimates computed from our predictive model and the average score assuming that the context and query models each get a weight of 0.5, the optimal combination across all queries (computed by averaging w over all sessions in S_T). To provide an upper bound for $F1$, we computed the score that could be obtained if we used an oracle, the optimal w for each query in the test set. This is shown in the last row of the table. The oracle's $F1$ score is not one because it is based on the optimization described earlier, and may contain noise. These scores give us a sense for the gain to be obtained from estimating the context weight per query and how much lift we would get if we just applied a global optimum (derived by averaging the optimal values over all queries), removing the need for engines to deploy a run-time classifier.

Table 4. Predictive accuracy for heuristic, learned, and oracle context weights.

Context weight source	$F1$	**Percentage of oracle**
Default (w = 0.5)	0.49	75.3
Global optimum (w = 0.75)	0.52	80.0
Per-query optimum	0.56	86.1
Oracle	0.65	–

The findings suggest that the global optimum helps to obtain a performance that is close to the oracle (80%), but the per-query optimum, based on features of the query, the context, and their combination, achieves over 85% of the predictive accuracy of the oracle. Both the global optimum and per-query optimum led to significant improvements in predictive performance over the default, using paired t-tests (global: t(44873)=2.58, p<.01, per-query: t(44873)=2.87, p<.01). These findings suggest significant benefit from optimizing context weights, even if search engines can only use the global optimum due to infrastructure constraints.

4.4.5 Varying Context and Relevance Information

To evaluate how the amount of context and relevance information available to build the predictive model influences its prediction accuracy, we built models using different amounts of context and relevance. In building the context model we used either all previous actions or the most recent previous action. In building the relevance model we use all future actions, the next action, or the last action in the session. Table 5 shows the average performance.

Table 5. Average predictive performance by model source.

Relevance model source	**Context model source**	
	All actions	*Previous action*
All actions	r=0.85, $RMSE$=0.15	r=0.78, $RMSE$=0.19
Next action	r=0.83, $RMSE$=0.16	r=0.77, $RMSE$=0.19
Last action	r=0.83, $RMSE$=0.16	r=0.76, $RMSE$=0.19

We analyzed the models using the same methodology described in Section 4.4.3, ran a 3 × 2 ANOVA with relevance model source and context model source as the factors, and examined the significance of pairwise differences using Tukey *post-hoc* testing. We found that when context models were built from only the previous action, performance was significantly lower than when all previous actions were used (all p<.01). This suggests that additional preceding actions adds predictive signal. We found no significant difference for any source when varying the amount of information used to generate the ground truth (first subsequent action, last action in session, or all actions), all p≥.12. It appears that the source of relevance information has only a marginal impact on how well the models estimate the weight to assign to the context.

In this section we demonstrated predictive value in context, variation by source and by model, and have shown that we can learn to predict optimal context weights with good accuracy.

5. DISCUSSION AND IMPLICATIONS

Through a log-based analysis, we quantified the opportunity for using activity-based search context, compared the accuracy of interest models of the current query, context, and their combination, and learned optimal weights to combine the query and context models on a per-query basis. These findings can inform the design of search systems to leverage contextual information to better understand, model, and serve searchers' information needs.

Context models based on recent search activity present significant opportunity to improve search performance. We showed that over 60% of queries had at least one preceding query. The simple method we used for generating category labels covers almost 80% of all contexts, meaning that we can potentially improve engine responses for roughly 50% of the queries in our experiments.

We found that using context led improved the accuracy of predictions of future interests over the current query alone. This is in line with previous work, which has also demonstrated the benefits of contextualized search [24][26]. Further, leveraging increasingly-richer sources of contextual information (queries, SERP clicks, and post-SERP Web page visits) improved predictive accuracy. We showed that there are distinct query sets for which different interest models and sources perform most effectively, suggesting that query information is likely important in selecting sources and/or model weights. Finally, we showed improvements in predictive accuracy by learning per-query context weights.

By representing short-term session-contexts we are able to significantly improve our ability to model user intent. The richer and more accurate predictive models we developed can be used to interpret the query for a variety of search-related applications, including interface changes to emphasize results of likely interest, to suggest contextually-relevant query alternatives, or for ranking and filtering. Category-level information has already been shown to improve result relevance for just the current query [5]. A direct extension of our work would be to use the context model (assigned a weight based on features of the query and context) to improve the quality of search engine result rankings, by promoting results that are consistent with the inferred user intent.

There are several directions for improving model development. The gains in the optimal context weight prediction performance when moving from previous action only to all preceding actions suggests value in using multi-action context. To reduce noise from the context, we need to experiment with ways to select only relevant actions and explore other context decay functions. The priors used in the models for each of the sources were based on the behavior of many users across many different queries. Personalized model priors based on a user's search history could also be used so that predictions can be tailored to topics that interested them.

Further work is needed to verify the accuracy of the relevance models based on future actions. Manual labeling of a random sample of sessions (going beyond the visual inspection performed in this study) may be necessary to create a reliable ground truth and ensure that our session demarcation (which was temporally-based not topic-based) is accurate. These labels could provide an additional source of ground truth information for learning the optimal combination of query and context and in evaluating predictions or result rankings generated using such a combination. More generally, there are important opportunities to develop new context-aware evaluation methodologies. Evaluation of search results is typically based on the query alone, and new judgment protocols and evaluation metrics (e.g., [16]) are necessary to also consider search context. The framework we have developed to represent and predict user interests could also be generalized by using other context features and alternative outcome measures.

6. CONCLUSIONS

In this paper we have described a study investigating the effectiveness of activity-based context in predicting users' search interests. We demonstrated that context can be captured and modeled for a significant portion of search queries, suggesting that there lies significant opportunity in leveraging contextual information. We explored the value of modeling the current query, its context, and their combination (which we refer to as *intent*), and different sources of context (search queries, SERP clicks, and post-SERP navigation). Our findings showed that intent models developed from many sources perform best overall. In addition, we found that all models and context sources have some set of queries for which they provide the best performance. Thus, we also developed techniques to learn the optimal combinations of query and context models per query. Our findings demonstrate significant opportunity in leveraging short-term contextual information to improve search systems. Future work involves constructing more sophisticated user interest models, and the development and deployment of search engine enhancements to ranking and result presentation that leverage context information effectively.

REFERENCES

[1] Agichtein, E., Brill, E. and Dumais, S. (2006). Improving web search ranking by incorporating user behavior information. *Proc. SIGIR*, 19-26.

[2] Agichtein, E., Brill, E., Dumais, S. and Ragno, R. (2006). Learning user interaction models for predicting web search result preferences. *Proc. SIGIR*, 3-10.

[3] Armstrong, R., Freitag, D., Joachims, T. and Mitchell, T. (1997). WebWatcher: a learning apprentice for the world wide web. *Proc. IJCAI*, 770-775.

[4] Bailey, P., White, R.W., Liu, H. and Kumaran, G. (2010). Mining past query trails to label long and rare search engine queries. *ACM TWEB*, in press.

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

[6] Campbell, I. and Van Rijsbergen, C.J. (1996). The ostensive model of developing information needs. *Proc. CoLIS*, 251-268.

[7] 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. *Proc. KDD*, 875-883.

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

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

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

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

[12] Friedman, J.H., Hastie, T. and Tibshirani, R. (1998). *Additive Logistic Regression: A Statistical View of Boosting*. Tech. Report, Department of Statistics, Stanford University.

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

[14] Hassan, A., Jones, R. and Klinkner, K.L. (2010). Beyond DCG: user behavior as a predictor of a successful search. *Proc. WSDM*, 221-230.

[15] Ingwersen, P. and Järvelin, K. (2005). *The Turn: Integration of Information Seeking and Retrieval in Context*. Springer.

[16] Järvelin, K., Price, S., Delcambre, L. and Nielsen, M. (2008). Discounted cumulated gain based evaluation of multiple-query IR sessions. *Proc. ECIR*, 138-149.

[17] Jeh, G. and Widom, J. (2003). Scaling personalized web search. *Proc. WWW*, 217-279.

[18] Kelly, D., Dollu, V.D. and Fu, X. (2005). The loquacious user: a document-independent source of terms for query expansion. *Proc. SIGIR*, 457-464.

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

[20] Muhalkova, L. and Mooney, R. (2009). Learning to disambiguate search queries from short sessions. *Proc. ECML*, 111-127.

[21] Pitkow, J., Schütze, H., Cass, T., Cooley, R., Turnbull, D., Edmonds, A., Adar, E. and Breuel, T. (2002). Personalized search. *CACM,* 45(9): 50-55.

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

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

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

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

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

[27] Teevan, J. (2008). How people recall, recognize, and reuse search results. *ACM TOIS*, 26(4).

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

[29] White, R.W. and Drucker, S.M. (2007). Investigating behavioral variability in web search. *Proc. WWW*, 21-30.

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

[31] Voorhees, E. and Harman, D. (2005). *TREC: Experiment and Evaluation in Information Retrieval*, MIT Press.

[32] Xiang, B., Jiang, D., Pei, J., Sun, X., Chen, E. and Li, H. (2010). Context-aware ranking in web search. *Proc. SIGIR*.