Modeling and Analysis of Cross-Session Search Tasks

Alexander Kotov[1], Paul N. Bennett[2], Ryen W. White[2], Susan T. Dumais[2], and Jaime Teevan[2]
[1] Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA
[2] Microsoft Research, One Microsoft Way, Redmond, WA 98052, USA
akotov2@illinois.edu, {pauben, ryenw, sdumais, teevan}@microsoft.com

ABSTRACT
The information needs of search engine users vary in complexity, depending on the task they are trying to accomplish. Some simple needs can be satisfied with a single query, whereas others require a series of queries over a longer period of time. While search engines effectively satisfy many simple needs, searchers receive little support when their information needs span sessions. In this work, we propose methods for modeling and analyzing user search behavior that extends over multiple search sessions. We focus on two problems: (i) given a user query, identify all related queries from previous sessions that the user has issued, and (ii) given a multi-query task for a user, predict whether the user will return to this task in the future. We model both problems within a classification framework that uses features of individual queries and long-term user search behavior at different granularity. Experimental evaluation of the proposed models for both tasks indicates that it is possible to effectively model and analyze cross-session search behavior. Our findings have implications for improving search for complex information needs and designing search engine features to support cross-session search tasks.

Categories and Subject Descriptors
H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval – *search process, selection process*

General Terms
Algorithms, Experimentation, Human Factors

Keywords
Cross-session search tasks, machine learning, user behavior.

1. INTRODUCTION
Web searchers perform a broad range of information seeking tasks, from figuring out how to spell a word to researching cancer treatment options. Correspondingly, the information needs of search engine users vary in complexity. Some simple information needs, like finding a person's home page or navigating to a social networking site, can be unambiguously expressed as keyword queries and have distinct answers. Other, more complicated needs, like planning a wedding or vacation, have multiple aspects and cannot be satisfied by the results shown on a single search result page. Addressing complex information needs requires a user to issue a series of queries, potentially spanning a long period of time and multiple search sessions. In doing so, searchers may collect, filter, and summarize information from many Web pages. In previous work [9], it has been estimated through manual analysis of query logs that approximately 10% of search sessions include queries on such longitudinal tasks and 25% of the overall query volume corresponds to this type of search task.

SIGIR'11, July 24–28, 2011, Beijing, China.

While modern search engines effectively serve many of the individual queries that correspond to simple information needs, users get little or no help when their information needs transcend the boundary of a single search session. A *search session*, as defined by Boldi *et al.* [5], is a *sequence of queries issued by a single user within a specific time limit*. In this work, we model and analyze complex, multi-session information needs, which we call *cross-session search tasks*. Cross-session tasks are related to research missions and goals [9][16], as we describe in more detail below, but we explicitly focus on tasks that extend across sessions. We assume that an individual cross-session task consists of a series of queries that corresponds to a distinct high-level information need. The queries related to the task are not necessarily consecutive, and a single search session may contain interleaved queries from multiple cross-session tasks, as well as shorter, within-session tasks. Cross-session tasks may evolve over time, with users starting with only a general idea of what they are searching for and progressively refine their need over time. During a single session, a user may find some results of interest, disregard others and continue exploring, try related queries, or interleave one task with other tasks. They may then drop the task before returning to it at a later time [17][19][20].

Since Web search is currently stateless, the cognitive burden of keeping track of complex search tasks is placed on the searcher. Incorporating the analysis and prediction of long-term user search behavior into search engine infrastructure could improve the search experience for searchers in many ways. For example, Web search traditionally only considers a user's current query when identifying relevant search results. If a search engine were able to identify past queries and interactions related to the searcher's current long-term intent, this information could be used to improve search quality. Past information could also be retained and displayed to help the user re-establish the context of a long-term search task, relieving the user from the burden of recalling past queries and pages visited (e.g., [9]). Similarly, if a search engine could predict that a user was going to return to a task that has only been temporarily suspended, the search engine could help support future related searches by, for example, pre-identifying and pre-caching relevant documents, or soliciting user assistance to archive the current search session for future use.

In this paper, we explore how effectively cross-session search tasks can be modeled. Specifically, we focus on two related problems: (i) identify all previous queries by a user on the same search task, and (ii) given a search task for a user, predict whether the user will return to this task in the future. After a discussion of recent work in modeling user search behavior, we formally define the specific aspects of the challenge of modeling and analyzing cross-session search tasks that we address. We next discuss the formulation of the two research problems as classification tasks, and describe the classification models used. We then present the

experimental setup and results for both problems. Finally, we summarize our findings and discuss future work.

2. RELATED WORK

User search behavior has been actively studied in recent years, primarily using search logs as a valuable resource to understand people's interactions with Web search engines. Earlier work on understanding search behavior focused on methods for classifying queries into high-level search goals, such as informational, navigational and transactional [8][18][29]. Downey *et al*. [10] studied the relationship between information needs and their formulation as search queries. Recent work indicates that, in addition to queries themselves, long-term and short-term search contexts can be effectively leveraged to predict user interests [34], search success [14], and improve search result ranking [1].

Search behavior can be analyzed over time to identify queries that express the same underlying information need. Most previous work has focused on search behavior analysis and prediction within a single search session. Related queries within a session have been referred to as being part of a *query chain* [26] or *search goal* [14][16][24]. He *et al*. [15] proposed an algorithm to segment a query stream into sessions by detecting topical shifts between the queries. Hassan *et al*. [14] modeled session-level search goals using hidden Markov models. They experimentally demonstrated that models that took into account users' search behavior were more predictive of session success than those that relied on document relevance. Piwowarski *et al*. [24] used a layered Bayesian network to model a hierarchy of user search actions, with the goal of identifying distinct patterns of user search behavior that correspond to the latent states of the Bayesian network. They used a classifier to learn the mapping from the distribution of latent states for a clicked document to the relevance assessment of that document in the absence of document content models. Mei *et al*. [21] proposed a general framework to study sequences of search activities and focused on simple prediction and classification tasks, ranging from predicting if the next click will be on an algorithmic result to segmenting the query stream into goals and missions. Single-session analyses have also been used for various search-related tasks such as query suggestion [6][7], interactive feeback [30], and query disambiguation [22].

In this paper we focus on tasks that extend across sessions. There has been some work on characterizing such tasks using log or survey data, and on automatically identifying queries on the same task. Teevan *et al*. [32] showed, via query log analysis, that nearly 40% of queries were attempts to re-find previously encountered results. Using a survey methodology, Aula *et al*. [2] studied the search and information re-access strategies of experienced Web users. They found that people often have difficulty remembering the queries they used originally to discover information of interest. In a field study of 21 people, MacKay and Watters [20] explored a variety of Web-based information seeking tasks. They found that information gathering tasks accounted for 13.4% of tasks, and that 58.8% of these tasks continued across sessions. Information gathering tasks were complex and people used a variety of browser tools and actions to help complete these tasks. Liu and Belkin [19] examined the structure (parallel or dependent) of tasks that extend across different sessions. Jones and Klinker [16] proposed methods to partition a query stream into research missions and goals, where each mission corresponds to a set of related information needs and may include multiple search goals. This work is closely related to our problem of identifying previous queries on the same search task. However, we do not decompose tasks into a hierarchical structure of missions and goals, we examine tasks that extend over a longer time period (up to a week) and we study an order of magnitude more users. We also propose many new features and experiment with different classification models.

Several algorithms and tools have been developed to support the resumption of search tasks. Morris *et al*. [23] developed *SearchBar*, a system that proactively and persistently stores query histories, browsing histories, and users' notes and ratings. SearchBar supports multi-session investigations by assisting with task context resumption and information re-finding. However, instead of determining the search tasks automatically, SearchBar requires users to explicitly identify them, thus creating additional user overhead. Donato *et al*. [9] developed *SearchPad*, a system which automatically identifies research missions and presents a search workspace comprising previous queries and results related to the mission. SearchPad uses measures of topic coherence between pairs of consequtive queries and user engagement to identify such research missions. Although Donato and colleagues describe a method and architecture for detecting research missions, the system evaluation presented in [9] focused primarily on systems issues (triggering without influencing latency and using online evaluation to set application-specific parameters). They did not compare alternative classification algorithms or consider the problem of predicting whether users will return to the same task in the future.

The research presented here addresses two important problems in modeling cross-session information needs: (i) identifying all previous queries in a user's search history on the same task as the current query, and (ii) predicting whether a user will return to the task in future sessions. We formalize these problems as classification tasks within a supervised learning framework. Our work differs from prior work in several ways:

- We formalize and evaluate the problem of predicting search task continuation. Predicting whether a search task will be resumed in the future is new, as far as we know.
- We extend previous work on same-task detection (notably that on research missions) by examining tasks that extend over longer time periods, studying many more searchers, and evaluating new features and classification models.
- We describe a new method for automatically and semi-automatically creating labeled data sets that can be used for both problems addressed in this work.
- We experiment with different feature sets and classifiers to identify the most informative features and the best-performing classifiers for the two problems addressed here.

In the following section, we formally define the two problems that we address and the context from which they arise.

3. PROBLEM DEFINITION

User search behavior has been modeled at different levels of granularity, ranging from eye fixations on search results [11] to information needs underlying a set of queries [16][26]. Different sources of data can be used depending on the particular modeling task. An important data source for many high-level modeling tasks is search logs, traditionally defined as follows:

DEFINITION 1 SEARCH LOG *$\mathcal{L}$ is a temporally-ordered set of quintuples $(u_i, t_i, q_i, \mathcal{R}_i, \mathcal{C}_i)$, associated with user queries, where u_i is the identifier of the user, t_i is the time of user action, q_i is the query submitted by the user, $\mathcal{R}_i$ is the set of results returned for q_i and $\mathcal{C}_i$ is the set of clicks on results $\mathcal{R}_i$.*

Time	Query	Automatic Label	Labeled Dominant	
			Automatically	By a Human
1/22/2011 1:10pm	peanut butter recipes	peanut butter recipes	×	×
1/22/2011 1:13pm	peanut butter cookies	peanut butter recipes	×	×
1/22/2011 1:25pm	calories peanut butter cookies	peanut butter recipes	×	
1/22/2011 3:10pm	weather nyc	weather nyc		
1/22/2011 3:11pm	nyc 10-day weather forecast	weather nyc		
1/22/2011 3:15pm	peanut butter sandwiches	peanut butter recipes	×	×
1/22/2011 3:16pm	pb&j			×
1/22/2011 3:18pm	fluffanutter			×
1/22/2011 3:19pm	fluffernutter			×
1/22/2011 6:15pm	sigir 2011	sigir 2011		
1/22/2011 6:17pm	sigir 2010 schedule	sigir 2011		
1/23/2011 3:17pm	nytimes			
1/24/2011 3:00pm	flight status united 123			
1/24/2011 3:31pm	peanut butter cookies low calorie	peanut butter recipes	×	×
1/24/2011 3:33pm	peanut butter cookies foodtv	peanut butter recipes	×	×
1/24/2011 4:10pm	weather forecast nyc	weather nyc		
1/25/2011 3:05pm	nytimes			
1/25/2011 3:29pm	foodtv			×
1/25/2011 3:31pm	famous pb&j drop recipe			×
1/25/2011 3:31pm	famous pb&j drop cookie recipe			×
1/25/2011 3:33pm	pb&j drop cookie recipe foodtv			×
1/25/2011 3:43pm	peanut butter cookies foodtv	peanut butter recipes	×	×

Table 1. Example of automatically and manually assigned cross-session and early-dominant tasks labels.

Queries from the search logs can be aggregated for each user to create a user search history:

DEFINITION 2 SEARCH HISTORY $\mathcal{H}_u = \{(a_1, t_1), \dots, (a_n, t_n)\}$ *for a particular user* u *is a temporally-ordered sequence* ($t_1 < \cdots < t_n$) *of pairs of user search actions (such as issuing a query, clicking on a search result URL and navigating back to search results) and time stamps associated with each action.*

The search history provides rich sequences of observations for making inferences about search behavior, including what intent motivated the user to search and whether that intent was satisfied. In order to simplify the analysis of user search history, it is typically partitioned into units, called search sessions:

DEFINITION 3 SEARCH SESSION $\mathcal{S} = \{(a_i, t_i), \dots, (a_j, t_j)\}$ *is a maximal subset of user search history* $\mathcal{S} \subset \mathcal{H}_u$, *such that* $\forall k: k = i, \dots, j\ t_{k+1} - t_k < \Theta$, *where* Θ *is a threshold for a period of user inactivity.*

In query log analysis, sesson timeouts are often used as boundaries to demarcate the sessions. The session threshold is typically set to 30 minutes [25][26][32]. Since search sessions include user actions other than queries, the time interval between two successive queries in one session can be more than 30 minutes. A sequence of queries forms a *user query stream*:

DEFINITION 4 QUERY STREAM $\mathcal{Q}_u = \{(q_1, t_1), \dots, (q_n, t_n)\}$ *is a temporally-ordered sequence of queries, submitted by a particular user* u *during the course of user search history.*

The first two columns in Table 1 show an example of a query stream for a fictitious user. In this example, more than 20 queries (related to several intents) are issued over the course of four days.

DEFINITION 5 CROSS-SESSION SEARCH TASK

$$\mathcal{T} = \{(q_i, t_i, l_i), \dots, (q_j, t_j, l_j)\}$$

is a subset of a query stream $\mathcal{T} \subset \mathcal{Q}_u$, *corresponding to a certain high-level intent that motivated search. Each query* q_i *is automatically or semi-automatically assigned a label* l_i. *When a labeled task extends over multiple sessions, we designate this as a cross-session search task.*

Cross-session tasks typically correspond to high level information needs, which may not be directly reflected in the queries. For example, queries corresponding to the same task may not have any terms in common. Such tasks represent a level of abstraction above the stream of queries. For the purpose of modeling, all queries about the same cross-session task can be assigned a label, representing such a task. The third column of Table 1 provides examples of automatic labels, assigned to the queries in the second column, according to the method presented in Section 4.2. As can be seen from this example, automatic labeling is effective when there is significant term overlap between the queries or when the inference step needed to relate one query to another query is fairly simple. However, our automatic labeling method was unable to infer the meaning of the acronym "pb&j" and connect it to the task labeled as "peanut butter recipes."

In order to identify previous queries on the same task or predict if a task is going to continue in the future, we must first define the task of interest. Because our search log data covered a limited period of time, we chose to focus our labeling efforts on the tasks that occurred early in the observation period and thus have the potential to be continued in the future. We designated those tasks as *early-dominant tasks*, and formally define them as follows:

DEFINITION 6 EARLY-DOMINANT TASK*: given a query stream* $\hat{Q}_u = \{(q_1, t_{11}, l_1), \ldots, (q_n, t_{nd}, l_n)\}$*, in which the queries are labeled with the tasks they correspond to, for a particular user* u *over* d *days, an early-dominant task is the first task that spans at least* k *distinct queries with the same label that occur within the first* m *days of the user search history. When multiple task labels meet the threshold criterion, the first such task is taken as the early-dominant task.*

The goal of identifying the early-dominant task is to create a data set for the problem of predicting whether the user will return to the task, given search log data covering a limited period of time. Parameters k and m can be set depending on the length of the available search history. The fourth column in Table 1 shows the queries that were automatically labeled as corresponding to the early-dominant task, based on the criteria that there should be at least two distinct queries automatically labeled with the same task (peanut butter recipes) in the first two days of the observed user search history. By requiring two distinct queries on the same task we omit most repeat navigational tasks, which are easy to identify and have been studied by others [32].

Query streams containing automatically-identified early-dominant tasks can be post-processed by humans to add additional queries that were missed by the labeling method or remove incorrectly labeled queries. As can be seen from the fifth column of Table 1, human annotators added queries on "pb&j drop cookie recipe," "fluffernutter," and "foodtv" to the automatically-identified early-dominant task.

In this work, we focus on two specific practical problems arising in the context of *cross-session search tasks*:

1. **SameTask:** Given a user query, *identify all previous queries on the same early-dominant search task* in the query stream of the user;
2. **TaskContinuation:** Given an early-dominant task for a user and the last query of the user on the early-dominant search task, *predict whether the user will return to this task in a future session*.

Methods that effectively solve the above two problems can be applied in a range of search scenarios. We describe some specific application scenarios for cross-session tasks later in the paper.

Having introduced the two problems in the context of analysis of *cross-session search tasks* addressed in the present work, we now discuss our approach to solving them. For both tasks we adopt a machine learning methodology by learning a classifier on a catalog of features. In the following section we describe the experimental setup for both problems.

4. EXPERIMENTAL METHODOLOGY

The most important and challenging aspect of an experimental evaluation of models for the analysis of cross-session tasks is generating a training set, in which the queries are labeled with the corresponding long-term tasks. In this section, we discuss the data that we used for evaluating the models proposed in this work as well as the semi-supervised process for generating training labels.

4.1 Dataset

We used a dataset containing the anonymized logs of URLs visited by users who consented to provide interaction data through a widely-distributed browser plug-in. 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 are useful in understanding and modeling natural search behavior. Log entries include a timestamp for each Web 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 that started with a query to Bing using a session extraction methodology similar to [35]. After the initial query, search sessions include subsequent clicked results and queries occurring in the same browser/tab instance, and ending after 30 minutes of inactivity.

Number of users	**3k**	**10k**	**Human**
Total	3,376	10,852	1,218
Return to dominant task	1,688	1,694	701
Number of queries	**3k**	**10k**	**Human**
Total	66,219	119,814	28,474
Query pairs	866,860	1,486,492	660,120

Table 2. Statistics of data set used in experiments.

4.2 Labeling

Labeling queries with long-term search tasks is challenging, since it requires an abstraction from the query stream to the level of information needs. The mapping from queries to information needs is perhaps best performed by human annotators, who are better able than machines to understand the relationship between the queries and visited URLs. However, manual labeling of a large data set is cognitively demanding and time consuming. To address this, we examine both fully automatic initial labeling (for more than ten thousand users) and additional human annotation for a subset of these data (more than one thousand users).

4.2.1 Automatic labeling

The automatic labeling process comprises four stages:

In the **first stage**, a subset of users, who are likely to be involved in some long-term task, were selected using a simple heuristic: a user must have at least five search sessions with at least 10 queries in their search history during the week covered by our log. This resulted in a set of 270,470 users.

In the **second stage**, queries in the original query stream were expanded using two query association resources:

1. A list of queries, in which each query is associated with a set of clusters computed using query reformulation and click data via the method described by Radlinski *et al*. [27]. Each cluster is assigned a score, reflecting the frequency of a particular intent in the data set;
2. A list of pairs of related queries, with the strength of association between the queries derived using query-click data in a manner similar to what has been described in previous work [4][33]. A query may be part of many pairs.

The representation of a query was expanded with queries from the top-scoring query cluster, and with related queries from the query graph. Each expanded query was then divided into terms to create a bag-of-words representation for the query.

In the **third stage**, in order to automatically identify and label queries that are on the same task, all pairs of queries in the user's query steam were enumerated and two similarity measures were

calculated for each query pair, using the bag-of-words query representations. We used the size of the intersection and the Jaccard coefficient between the term sets of two queries as similarity measures and experimented with different thresholds for both measures. If a similarity measure exceeded a threshold, the queries were labeled as belonging to the same long-term task. The labels were assigned to the pair of queries exceeding the threshold according to the following rules: (i) if one of the queries has already been labeled before, the other query is assigned the same label; (ii) if neither query has been labeled before, the first query in the pair is used as the label for both queries.

In addition to automatically assigning labels corresponding to search tasks, we determined whether there was an *early-dominant task* in each user's search history. An early-dominant task was previously defined as the first task during the first two days of the search history that was associated with at least two unique queries. The early-dominant task labels were used to evaluate predictions of whether a user will return to a task in a future session. Focusing labeling efforts on a single task per user (namely the early-dominant task) simplifies the human editorial task labeling process and, as we show in the next section, leads to high inter-judge agreement.

We experimented with all combinations of query expansion strategies and similarity measures and evaluated each combination in terms of the proportion of queries with automatically assigned cross-session task labels, the total number of users who have early-dominant task labels, and the proportion of users who return to the early-dominant task. The best performing automatic labeling method expanded queries using both the top scoring query cluster and a one-step walk on the query graph, and used the Jaccard coefficient with threshold 0.5.

In the **fourth stage**, from the set of 270,470 users selected in the first stage, we selected a subset of 10,852 users who had an early-dominant task and issued at least one query after the first two days. Of these people, 1,694 (15.6%) issued at least one query on the dominant task after the first two days and the remaining 9,158 (84.4%) did not. We will refer to this data set as *10k* and it represents an 85-15 split between the (automatically labeled) negative and positive examples of returning to the dominant task. In order to create a training set with a balanced number of positive and negative examples, we included all users who returned to the early-dominant task and randomly subsampled 1,688 of the 9,164 users who did not return to the early-dominant task. We will refer to the resulting data set with a 50-50 split between the positive and negative examples as *3k*.

4.2.2 Human labeling

From the *3k* data set we randomly selected 1,250 users and three annotators manually modified the automatic labels. Annotators were instructed to start with the automatically-identified early-dominant task and find other queries by the same user that were on the same task. This typically involved identifying additional queries that were missed by the automatic algorithm, but sometimes also involved removing queries that were not on the same task. As a result, 1,218 users had an early-dominant task and 701 (57.6%) of these users returned to the task in a subsequent session. Table 2 summarizes the statistics of the data sets used for experiments. The numbers presented for the 3k and 10k data sets are for the fully automatic labels, and the Human set represents the human augmentation.

Automatic labeling identified 7,038 queries corresponding to the early-dominant task (for the 1,250 users). The human labeling process identified more than 7,500 additional queries on the early-dominant task, for a total of 14,549 such queries. Examples of additions include: adding *angelcare deluxe* to a task about *baby monitors*, and adding *princesspeach* to a task about *mariotoys*. The labelers also removed 232 of the queries that the automatic method had identified as dominant. Examples include: removing *map of south carolina* from a task about visiting haiti which included the query *map of haiti*, and correcting some labeling errors. Overall the agreement was 96.7% for automatic postive labels and 79.2% for automatic negative labels.

All annotators also labeled queries from 100 additional users which we used to measure inter-labeler agreement. Cohen's kappa showed high inter-annotator agreement, ranging from 0.86 to 0.92 for the three pairs of annotators.

4.2.3 Task characteristics

Recall that we required two distinct queries on the early-dominant task in an effort to omit simple repeat navigational tasks. In order to verify this, the query sessions used to assess inter-labeler agreement were also annotated as to whether the early-dominant task was navigational or informational. The majority (88%) of the tasks identified by the method described above were indeed informational. The remaining 12% of the tasks were navigational and were included because of spelling errors, word boundary differences (*mc gilvery oil* vs. *mcgillvery oil*), or different query formulations to find items of interest *(drudgereport com, matt drudge report, drudge report*). As intended, the vast majority of tasks identified were informational needs such as research, school work, shopping, travel planning, and general topic search [20].

4.3 Classifiers

We used two different classifiers to address the two problems of multi-session search tasks outlined earlier: (i) identifying all previous queries on the same early-dominant task (*SameTask*), and (ii) given an early-dominant task for a user, predicting whether the task will be continued in subsequent sessions (*TaskContinuation*). The two classifiers that we used were Logistic Regression (LR) and Multiple Additive Regression Trees (MART) [13]. MART is a boosted tree algorithm that uses gradient descent for regression and classification. Logistic regression has previoulsy been used for similar task-modeling problems [16]. MART allows us to model conditional interactions so that we can evaluate the importance of richer feature combinations.

For all experiments, we used *z*-score normalization for feature values and performed 10-fold cross validation.

4.4 Performance measures

To compare the performance of the classification methods we look at the standard performance measures of accuracy and F1 [28]. We also display precision, recall, and the contingency tables for each method. By analogy to topic classification, we look at both the *macro* average of F1 which weights the F1 for each query equally and *micro* average which weights each binary prediction (for the current query to all previous queries) equally 0. We look at only the micro averages for the problem of predicting whether the user will return to the task, as there is only one example per user and thus macro and micro are the same. Significance between approaches is calculated using two-tailed independent samples *t*-tests.

In the following sections, we discuss in detail our approaches to solve the *SameTask* and *TaskContinuation* problems in the context of modeling cross-session tasks.

Query-Based Features	
NUMQUERYCHARS	The number of characters in the query.
NUMQUERYTERMS	The number of terms in the query.
NUMTOP10CLICKS	The number of clicks on the top 10 search results for the query.
MINCLICKPOS	The minimum position of a clicked result for the query.
MAXCLICKPOS	The maximum position of a clicked result for the query.
SPELLSUGGEST	Whether a spelling suggestion was shown for the query.
Session-Based Features	
NUMQUERIESSESS	The number of queries since the beginning of the session.
NUMCLICKSSESS	The number of clicks on search results since the beginning of the session.
TIMESINCESESS	The amount of time from the beginning of the session to the query.
SAMEQUERYSESS	Whether the same query appeared earlier in the same session.
SUBQUERYSESS	Whether a query with a subset of the query's terms appeared in the same session.
SUPQUERYSESS	Whether a query with a superset of the query's terms appeared in the same session.
History-Based Features	
NUMSESSHIST	The number of sessions in the user's search history.
NUMQUERIESHIST	The number of queries in the user's search history.
NUMCLICKSHIST	The number of clicks on search results in the user's search history.
SAMEQUERYHIST	Whether the same query appeared in the user's search history.
SUBQUERYHIST	Whether a query with a subset of the query's terms appeared in the user's history.
SUPQUERYHIST	Whether a query with a superset of the query's terms appeared in the user's history.
Pair-Wise Features	
NUMTERMSOVER	The number of overlapping terms between the two queries.
QUERYTERMSJAC	Jaccard coefficient between the term sets of two queries.
LEVENDIST	Levenshtein edit distance between the two queries.
TIMEBETWEEN	The time between the two queries.
SAMESESS	Whether the two queries occurred in the same session.
QUERIESSAME	Whether the two queries are identical.
QUERYSUBSET	Whether one query's terms are a subset of the other's.
HAVECOCLICKURLS	Whether the two queries have co-clicked URLs.
HAVECOCLICKDOM	Whether the two queries have co-clicked URLs with the same domain.

Table 3. Features for identifying whether two queries relate to the same long-term task. Except for the pair-wise features, every feature is calculated individually for each query in a pair and thus occur twice. Session and history features are computed over all actions *before* the current query.

5. SAME-TASK QUERY IDENTIFICATION

We begin by addressing the first problem of detecting queries on the same long-term task. Specifically: *for a given query, find all previous queries in the user's search history that are on the same long-term task*. We formulated this as a classification problem. Given a set of users, whose queries have been manually or automatically labeled with the early-dominant task, train a classification model that will classify each pair of user queries as either on the same task or not.

5.1 Formal definition

Formally, given a history of automatically or manually labeled queries $\mathcal{H} = \{(q_1, t_{11}, l_1), \ldots, (q_n, t_{nd}, l_n)\}$, where $l_i = \{d, \neg d\}$ is a label that indicates whether the query is part of the early-dominant task, we create a set of all possible pairs of queries:

$$\{(q_1, q_2, l_{12}'), \ldots, (q_1, q_n, l_{1n}'), \ldots, (q_{n-1}, q_n, l_{n-1,n}')\}$$

in which each pair is labeled as either a positive or negative example as follows:

$$l_{i,j}' = \begin{cases} 1, & \text{if } l_i = d \text{ and } l_j = d \\ 0, & \text{if } l_i = \neg d \text{ or } l_j = \neg d \end{cases} \quad (1)$$

A query pair is labeled as a positive example if both queries are related to the early-dominant task, and as a negative example if one of the queries is not related to the early-dominant task. We do not consider instances where neither query is labeled as being related to the early-dominant, since the queries might be on the same long-term task (but not labeled as the early-dominant task). The numbers of query pairs given in Table 2 were calculated *after* dropping such pairs. Because queries are only compared with queries that occur earlier in the user's history, the later in time that a query occurs, the more pairs it will be involved in.

5.2 Features

Since the *SameTask* problem involves predicting the similarity relationship between pairs of queries, we extracted pair-wise features as well as features for the individual queries. Individual query features are computed at different levels of granularity for historical information of a user: ranging from the session in which a query occurred, to the entire search history of a user. The 18 single-query and nine pair-wise features extracted to identify queries on the same long-term task are summarized in Table 3.

As a baseline (BASE), we use logistic regression to learn a model using only Levenshtein edit distance between the current (given) query and all previous queries. This is a reasonable baseline under the assumption that an intelligently-chosen threshold applied to the dissimilarity between two queries could provide an accurate prediction of whether two queries are on the same task.

	3k			10k			Human		
	BASE	**LR**	**MART**	**BASE**	**LR**	**MART**	**BASE**	**LR**	**MART**
Micro statistics									
TP	57,273	86,673	97,870	66,175	112,081	126,369	78,084	194,126	166,438
FP	14,262	20,913	23,260	16,582	25,293	28,971	52,322	52,032	65,310
TN	717,120	710,469	708,122	1,298,996	1,290,285	1,286,607	309,972	310,262	296,984
FN	78,205	48,805	37,608	104,739	58,833	44,545	219,742	103,700	131,388
Recall	0.4157	0.6431	0.7258	0.4300	0.6693	0.7556	0.3267	0.6067	0.5617
Precision	0.8382	0.8135	0.8183	0.8240	0.8268	0.8233	0.6138	0.7857	0.7325
Accuracy	0.8970	0.9227	**0.9331**	0.9245	**0.9646**	0.9534	0.6104	**0.7629**	0.7118
F1	0.5495	0.7160	**0.7681**	0.5520	0.7383	**0.7876**	0.3957	**0.6670**	0.6156
Macro statistics									
Recall	0.8520	0.9243	0.9440	0.8205	0.9288	0.9507	0.7329	0.8440	0.8277
Precision	0.8746	0.9134	0.9218	0.8406	0.9183	0.9284	0.6661	0.7781	0.7944
Accuracy	0.9540	0.9569	**0.9597**	0.9612	0.9646	**0.9661**	0.7139	**0.8196**	0.8133
F1	0.8419	0.8904	**0.9063**	0.8131	0.8982	**0.9146**	0.5676	0.7164	**0.7265**

Table 4. Experimental results for identifying queries on the same cross-session task. The best result for F1 and accuracy in each dataset is in bold. Significant differences relative to BASE are indicated by underline ($p < .01$).
TP = Num. true positives, FP = Num. false positives, TN = Num. true negatives, FN = Num. false negatives.

5.3 Results

The results of classification experiments with features specified in Table 3 on three experimental data sets are reported in Table 4. Several conclusions can be made from the findings presented in Table 4. First, both classifiers (LR and MART) consistently outperform the baseline (BASE), which is not surprising. The two classifiers show similar levels of accuracy especially for the automatically-labeled data. Second, for the automatically-labeled data, classication results improve as more data is observed, but this is difficult to interpret since the ratio of positive to negative examples changed from 50-50 to 15-85. Third, classification performance decreases on the task labels that have been assigned by human annotators. This suggests that the human-labeled data provides a more challenging learning problem. This is expected since the human labels were intended to capture task structure that was not already captured automatically, and this often involved identifying related queries that were not lexically similar.

The micro precision-recall curves for the two classifiers for the task of identifying queries on the same cross-session task on different data sets are shown in Figure 1 (aggregated over all test splits). Performance at default thresholds is indicated by markers.

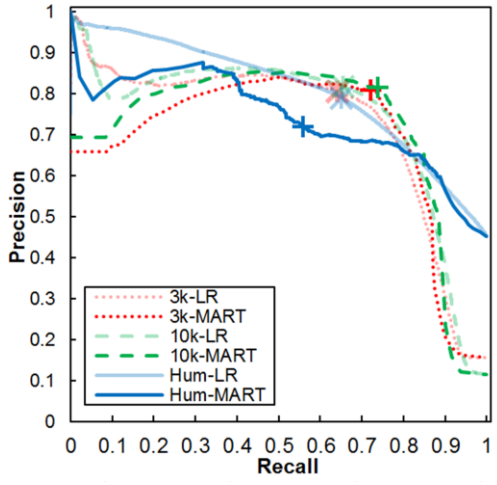

Figure 1. Micro precision/recall curves for LR and MART for identifying queries on the same task.

Both classifiers show good performance. However, for all three dataets, LR dominates at the low recall/high precision end of the curves. This suggests that the LR model might be more applicable for high-precision tasks such as suggesting related queries. As in Table 4, LR has notably better micro performance over the human-labeled data in the area of optimal F1 (upper right) although this does not translate to higher macro performance. Understanding the relationship between micro and macro performance for this domain is an area of future work.

5.3.1 Feature Importance Analysis

Feature weights from the logistic regression model for the identification of queries on the same long-term task, trained on the manually annotated data, are summarized in Table 5. Both single and pair-wise features are important in identifying queries on the same task. But, pair-wise features are more prevalent, meaning that understanding the relationship between the queries is more valuable than understanding either query in isolation. This is not surprising given that the goal is to understand if both queries are on the same task. Similarity features, such as whether the queries are identical and term overlap measures between the bag-of-words representations for a pair of queries (NUMTERMSOVER and

Feature	**Feature Type**	**Weight**
QUERYTERMSJAC	Pair-wise	1.44
NUMQUERYCHARS$_1$	Query-based	1.05
NUMTERMSOVER	Pair-wise	0.93
QUERYSUBSET	Pair-wise	0.88
NUMCLICKSHIST$_2$	History-based	0.81
NUMQUERYCHARS$_2$	Query-based	0.79
SAMESESS	Pair-wise	0.52
HAVECOCLICKDOM	Pair-wise	0.40
NUMCLICKSHIST$_1$	History-based	0.39
NUMQUERIESSESS$_1$	Session-based	0.31
SUBQUERYSESS$_2$	Session-based	-0.30
NUMQUERYTERMS$_2$	Query-based	-0.47
NUMQUERIESHIST$_1$	History-based	-0.52
NUMQUERYTERMS$_1$	Query-based	-0.68
LEVENDIST	Pair-wise	-0.84

Table 5. The top 15 (absolute magnitude) feature weights for the logistic regression model to identify queries on the same cross-session task. Features related to the first query are marked with a 1 and to the second query with a 2.

Session-Based Features	
AVGINTERQTIMESESS	The average time between all sequential pairs of queries in the session.
AVGINTERDOMQTIMESESS	The average time between sequential pairs of dominant queries in the session.
NUMDWELL30SESS	The number of queries in the session with a dwell time more than 30 seconds.
AVGNUMQTERMSSESS	The average number of unique terms per query within the session.
NUMDOMQUERIESSESS	The number of queries on the dominant task in the session.
PCTDOMQUERIESSESS	The number of queries on the dominant task divided by the number of queries in the session.
PCTCLICKQUERIESSESS	The number of the queries with clicked results divided by the number queries in the session.
NUMCOCLICKDOMSESS	The number of co-clicked URLs with the same domain in the session.
History-based features	
AVGINTERQTIMEHIST	The average time between all sequential pairs of queries in the user's search history.
AVGINTERDOMQTIMEHIST	The average time between sequential pairs of dominant queries in the user's search history.
NUMDWELL30HIST	The number of queries with dwell time more than 30 seconds in the user's history.
AVGNUMQTERMSHIST	The average number of unique terms per query in the user's search history.
NUMDOMQUERIESHIST	The number of queries on the dominant task in the user's search history.
PCTDOMQUERIESHIST	The number of queries on the dominant task divided by the number of queries in the user's history.
PCTCLICKQUERIESHIST	The number of queries with clicked results divided by the number of queries in the user's history.
NUMCOCLICKDOMHIST	The number of co-clicked URLs with the same domain in the user's search history.

Table 6. Additional features to those shown in Table 3 that are used for predicting whether a user will return to the early-dominant long-term task. The pair-wise features from Table 3 are not relevant for this task.

QUERYTERMJAC) are among the strongest signals. The high negative weight of the LEVENDIST feature shows that most queries on the same task are morphologically similar. Features related to the length of each individual query in characters (NUMQUERYCHARS) receive high positive weights, while features related to the length in terms (NUMQUERYTERMS) receive negative weights. This suggests that long, descriptive query terms are particularly indicative of cross-session tasks. This seems reasonable since longer terms may be associated with complex information needs spanning multiple search sessions.

6. TASK CONTINUATION PREDICTION

We next consider the second problem of predicting whether a user will return to a task. Specifically: *given an early-dominant task for a user and the user's last query on the early-dominant task, predict whether the user will return to this task in a future session*.

6.1 Formal definition

Given a stream of user queries and a target date, features are computed up to the end of the session containing the last early-dominant query on the target date. The feature vector is assigned a positive label, if there are queries on the early-dominant task after the target date, and a negative label if there are no such queries.

6.2 Features

The nature of the *TaskContinuation* problem suggests that the most predictive features should reflect two aspects of a cross-session task: (i) user satisfaction with the presented search results, and (ii) the difficulty of the task itself. The most frequently-used feature to capture user satisfaction is click-through rate on search results (e.g., [1]). The intuition behind this is that if a user issued a series of queries and clicked on at least one search result for most or all of these queries, she is likely to have obtained useful information that allowed her to make progress on the task and, hence, is less likely to return to it. Similarly, the dwell time on results has been shown to reflect satisfcation (e.g., Fox et al. [12]). The difficulty of the task can be reflected by several patterns of user behavior including the number of queries issued, the time between successive queries, etc. [3][17]. Individual query features, reflecting these interaction patterns, are computed at different levels of granularity (ranging from the session where a query occurred to the entire search history of a user). Features summarizing the user's history with the early-dominant task are also included. These additional features are shown in Table 6. The features in Tables 3 and 6 are used together to predict whether the user will return to the early-dominant task.

As a baseline (BASE), we use logistic regression to learn a model using the number of queries in the user's history before the cutoff date (NUMQUERIESHIST). This is a reasonable baseline under the assumption that an intelligently-chosen threshold applied to the level of user activity in a short window of time around the task could provide an accurate prediction of task continuation.

	3k			10k			Human		
	BASE	**LR**	**MART**	**BASE**	**LR**	**MART**	**BASE**	**LR**	**MART**
TP	680	1,112	1,106	14	470	523	569	514	489
FP	477	360	327	12	226	211	391	169	141
TN	1,211	1,328	1,361	9,146	8,932	8,947	126	348	376
FN	1,008	576	582	1,680	1,224	1,171	132	187	212
Recall	0.4038	0.6593	0.6546	0.0084	0.2777	0.3096	0.8154	0.7342	0.6971
Precision	0.5906	0.7555	0.7708	0.675	0.6784	0.7219	0.5961	0.7538	0.7766
Accuracy	0.5601	0.7228	**0.7308**	0.8441	0.8664	**0.8726**	0.5706	0.7074	**0.7100**
F1	0.4775	0.7029	**0.7072**	0.0166	0.3933	**0.4305**	0.6844	**0.7428**	0.7326

Table 7. Experimental results for predicting whether the user will return to the early-dominant task. The best result for F1 and accuracy in each dataset is in bold. Significance relative to BASE indicated by underline ($p < .01$).

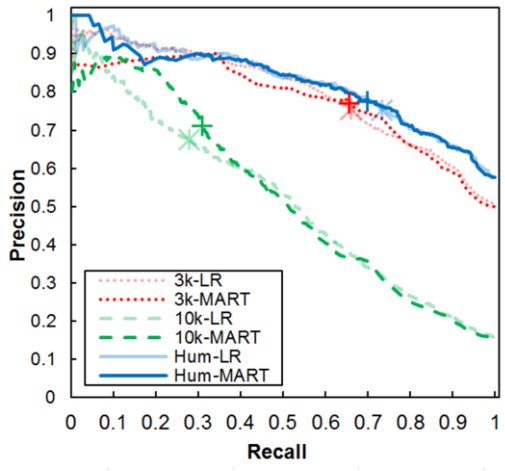

Figure 2. Precision/recall curves for LR and MART for predicting return to the early-dominant task.

Feature	Feature Type	Weight
SAMEQUERYHIST	History-based	1.11
NUMSESSHIST	History-based	0.60
NUMDOMQUERIESHIST	History-based (Table 6)	0.39
AVGINTERQTIMEHIST	History-based (Table 6)	0.24
FREQDOMQUERIESHIST	History-based (Table 6)	0.24
NUMDWELL30HIST	History-based (Table 6)	0.22
NUMQUERYHIST	History-based	0.21
NUMTOP10CLICKS	Query-based	-0.16
AVGINTERQTIMESESS	Session-based (Table 6)	-0.17
NUMCLICKSHIST	History-based	-0.18
NUMQUERYCHARS	Query-based	-0.21
SUBQUERYHIST	History-based	-0.23
SUPQUERYSESS	Session-based	-0.40
SUPQUERYHIST	History-based	-0.40
SUBQUERYSESS	Session-based	-0.49

Table 8. Top 15 (absolute magnitude) feature weights for the logistic regression model which predicts return to the early-dominant task.

6.3 Results

Experimental results for predicting whether the user will return to the early-dominant task with features specified in Tables 3 and 6 on three data sets are reported in Table 7. Several major conclusions can be drawn from Table 7. First, again we see that both classifiers improve over the baseline in all datasets and perform similarly to each other overall. Additionally, recall and precision substantially decrease when moving from a smaller balanced dataset (*3k*) to a larger unbalanced one with more negative examples (*10k*). However, both LR and MART still classify a large number of negative examples correctly, as evidenced by the fact that the accuracy increases for both classifiers. Next, recall significantly improves for both classifiers on manually corrected labels, which can be attributed to the fact that this data set is less sparse. There are more positive examples, because human annotators assign the missing early-dominant task labels to some queries.

Precision-recall curves for the classifiers for *TaskContinuation* are shown in Figure 2 (aggregated over all test splits). Performance at default thresholds is indicated by markers. Both models perform very similar for each dataset. There are some differences across datasets, with overall performance being worse on the *10k* set. MART has a slight advantage over LR for the low recall/high precision region of the curves for the human dataset.

6.3.1 Feature Importance Analysis

Weights of the LR model for predicting whether the user will return to the early-dominant task, trained on the editorially-corrected data set, are summarized in Table 8. From the table, it appears that the most important feature is whether the query has ever occurred in the user's history (SAMEQUERYHIST). All identical queries were labeled as being from the same task, both automatically and by human labelers. The importance of this feature is consistent with previous research that suggests re-finding is very common [32]. Although most of the early-dominant tasks are informational (as described in 4.2.3), queries are sometimes repeated as part of such tasks.

Features of the user's history with the dominant task were also important. If many of their past queries (NUMDOMQUERIESHIST) or a high proportion of them (PCTDOMQUERIESHIST) were related to the dominant task, they appeared particularly likely to return to the task again at a later date. This suggests that intense interest in a topic at one point in time is likely to lead to returning to the task at a later point (during the week).

In the previous section we observed that past queries on the dominant task could be identified in part by complex queries. Here again we see that the complexity of the user's dominant information need, as indicated by the high weight on longer queries (NUMQUERYCHARS) and deeper examination in result lists (NUMTOP10CLICKS), suggests that it is more likely that the user will return to the task. The predictive value of NUMTOP10CLICKS provides evidence to support similar claims by Donato *et al*. [9].

Features related to the user's intensity of search engine use (e.g., AVGINTERQTIMEHIST, NUMSESSHIST) are among the most important. This is not surprising, since people who search more are more likely to search again on all tasks, including both the early-dominant task and other tasks. However, not only the absolute frequency of searches, but also the deeper engagement with past search results (NUMDWELL30HIST) appear to be important, suggesting that people who use search deeply may also use search for more extended tasks.

7. CONCLUSIONS AND FUTURE WORK

As the importance of a deeper understanding of user search behavior continues to grow, it becomes necessary to develop models that consider complex long-term information needs and effectively incorporate them into existing search engine infrastructures. In this work, we introduced and addressed the two problems in the context of analysis of cross-session search tasks: (i) identifying queries from earlier sessions on the same task, and (ii) predicting whether a user will return to the same task during a later session, formulating both problems as supervised machine learning tasks. We proposed a method for creating a semi-automatically labeled data set that can be used for both problems and developed feature sets, tailored for each of the individual problems. Experimental results using two classifiers (logistic regression and MART) for both problems indicate that we can effectively model and analyze cross-session information needs.

Our research is an important first step in helping searchers more effectively manage long-term information needs. Knowledge of previous user queries on the same long-term task enables a search engine to provide support for task resumption. For example, if it is known that a user has previously been undertaking a vacation-planning task and has issued queries about airline tickets,

whenever the user comes back to this task, a search engine can show pertinent updates in the time since the last query (e.g., ticket price drops) or suggest queries to re-find useful past results. By using a model that can accurately predict continuation of a cross-session task, a search engine can determine whether it is necessary to retain the task context, start monitoring Web content (e.g., during Web crawls or others' queries) for information pertaining to the task, and use the task model for query suggestions or search result suggestions.

There are several directions for future work, including using richer prediction models and alternative feature sets, exploring new prediction and classification problems in the context of cross-session information needs, and incorporating our models into commerical search engines.

8. REFERENCES

[1] E. Agichtein, E. Brill and S. Dumais. Improving Web search ranking by incorporating user behavior information. *SIGIR '06*, 19–26, 2006.

[2] A. Aula, N. Jhaveri and M. Käki. Information search and re-access strategies of experienced Web users. *WWW '05*, 583–592, 2005.

[3] A. Aula, R. M. Kahn and Z. Guan. How does search behavior change as search behavior becomes more difficult. *CHI '10*, 35–44, 2010.

[4] D. Beeferman and A. Berger. Agglomerative clustering of a search engine query log. *KDD '00*, 407–416, 2000.

[5] P. Boldi, F. Bonchi, C. Castillo, D. Donato, A. Gionis and S. Vigna. The query-flow graph: Model and applications. *CIKM '08*, 609–618, 2008.

[6] H. Cao, D.H. Hu, D. Shen, D. Jiang, J.-T. Sun, E. Chen and Q. Yang. Context-aware query classification. *SIGIR '09*, 3–10, 2009.

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

[8] Y.-S. Chang, K.-Y. He, S. Yu and W.-H. Lu. Identifying user goals from Web search results. *WWW '06*, 1038–1041, 2006.

[9] D. Donato, F. Bonchi, T. Chi and Y. Maarek. Do you want to take notes? Identifying research missions in Yahoo! Search Pad. *WWW '10*, 321-330, 2010.

[10] D. Downey, S. Dumais, D. Liebling and E. Horvitz. Understanding the relationship between searchers' queries and information goals. *CIKM '08*, 449–458, 2008.

[11] S. Dumais, G. Buscher and E. Cutrell. Individual differences in gaze patterns for web search. *IIiX '10*, 185–194.

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

[13] J. Friedman, T. Hastie and T. Tibshirani. Additive logistic regression: A statistical view of boosting. *Annals of Statistics, 28(2),* 337–407, 2000.

[14] A. Hassan, R. Jones and K. Klinkner. Beyond DCG: User behavior as a predictor of a successful search.*WSDM '09*, 221–230, 2010.

[15] D. He, A. Göker, and D.J. Harper. Combining evidence for automatic Web session identification. *Information Processing & Management*, 38(5):727–742, 2002.

[16] R. Jones and K. Klinkner. Beyond the session timeout: Automatic hierarchical segmentation of search topics in query logs. *CIKM '08*, 699–708, 2008.

[17] M. Kellar, C. Watters, and M. Shepherd. A field study characterizing Web-based information-seeking tasks. *JASIST, 58(7),* 999–1018, 2007.

[18] U. Lee, Z. Liu and J. Cho. Automatic indetification of user goals in Web search. *WWW '05*, 391–400, 2005.

[19] J. Liu and N.J. Belkin. Personalizing information retrieval for multi-session tasks: The roles of task stage and task type. *SIGIR '10*, 26–33, 2010.

[20] B. MacKay and C. Watters. Exploring multi-session Web tasks. *CHI '08*, 1187–1196, 2008.

[21] Q. Mei, K. Klinkner, R. Kumar and A. Tomkins. An analysis framework for search sequences. *CIKM '09*, 1991–1994, 2009.

[22] L. Mihalkova and R. Mooney. Learning to disambiguate search queries from short sessions. *ECML '09*, 111–127, 2009.

[23] D. Morris, M. Ringel Morris and G. Venolia. SearchBar: A search-centric Web history for task resumption and information re-finding. *CHI '08*, 1207–1216, 2008.

[24] B. Piwowarski, G. Dupret and R. Jones. Mining user Web search activity with layered Bayesian networks or how to capture a click in its context. *WSDM '09*, 162–171, 2009.

[25] B. Piwowarski and H. Zaragoza. Predictive user click models based on click-through history. *CIKM '07*, 175–182, 2007.

[26] F. Radlinski and T. Joachims. Query chains: Learning to rank from implicit feedback. *KDD '05*, 239–248, 2005.

[27] F. Radlinski, M. Szummer and N. Craswell. Inferring query intent from reformulations and clicks. *WWW '10*, 1171–1172, 2010.

[28] C. J. van Rijsbergen. *Information Retrieval*. Butterworths, London, 1979.

[29] D.E. Rose and D. Levinson. Understanding user goals in Web search. *WWW '04*, 13–19, 2004.

[30] X. Shen, B. Tan and C. Zhai. Context-sensitive information retrieval using implicit feedback. *SIGIR '05*, 43–50, 2005.

[31] B. Tan, X. Shen and C. Zhai. Mining long-term search history to improve search accuracy. *KDD '06*, 718–723, 2006.

[32] J. Teevan, E. Adar, R. Jones and M.A.S. Potts. Information re-retrieval: Repeat queries in Yahoo's logs. *SIGIR '07*, 151–158, 2007.

[33] J.-R. Wen, J.-Y. Nie and H.-J. Zhang. Clustering user queries of a search engine. *WWW '01*, 162–168, 2001.

[34] R.W. White, P. Bailey and L. Chen. Predicting user interests from contextual information. *SIGIR '09*, 363–370, 2009.

[35] R.W. White and S.M. Drucker. Investigating behavioral variability in Web search. *WWW '07*, 21–30, 2007.\

[36] Y.Yang and Z. Liu. A re-examination of text categorization methods. *SIGIR '99*, 42-49, 1999.