Mining the Search Trails of Surfing Crowds: Identifying Relevant Websites From User Activity

Mikhail Bilenko
Microsoft Research
One Microsoft Way
Redmond, WA 98052, USA
mbilenko@microsoft.com

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

ABSTRACT

The paper proposes identifying relevant information sources from the history of combined searching and browsing behavior of many Web users. While it has been previously shown that user interactions with search engines can be employed to improve document ranking, browsing behavior that occurs beyond search result pages has been largely overlooked in prior work. The paper demonstrates that users' post-search browsing activity strongly reflects implicit endorsement of visited pages, which allows estimating topical relevance of Web resources by mining large-scale datasets of search trails. We present heuristic and probabilistic algorithms that rely on such datasets for suggesting authoritative websites for search queries. Experimental evaluation shows that exploiting complete post-search browsing trails outperforms alternatives in isolation (e.g., clickthrough logs), and yields accuracy improvements when employed as a feature in learning to rank for Web search.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Data Mining; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

Keywords

Learning from user behavior, Mining search and browsing logs, Implicit feedback, Web search

1. INTRODUCTION

Traditional information retrieval (IR) techniques identify documents relevant to a given query by computing similarity between the query and the documents' contents [36]. Challenges posed by IR on Web scale motivated a number of approaches that exploit data sources beyond document contents, such as the structure of the hyperlink graph [10, 23, 26], or users' interactions with search engines [17, 2, 4, 42], as well as machine learning methods that combine multiple features for estimating resource relevance [5, 7, 33, 19].

A common theme unifying many of these recent IR algorithms is the use of evidence stemming from unorganized behavior of many individuals for estimating document authority. Hyperlink structure created by millions of individual Web page authors is one example of phenomena arising from local activity of many users, which is exploited by such algorithms as HITS [23] and PageRank [26], as well as many others. Another group of IR algorithms that leverage user behavior on a large scale includes methods that utilize search engine clickthrough logs, where users' clicks on search results provide implicit affirmation of the corresponding pages' authority and/or relevance to the original query [17, 48, 2, 4]. Additionally, search engine query logs can be used to incorporate query context derived from users' search histories, leading to better query language models that improve search accuracy [42].

While query and clickthrough logs from search engines have been shown to be a valuable source of implicit supervision for training retrieval methods, the vast majority of users' browsing behavior takes place beyond search engine interactions. It has been reported in previous studies that users' information seeking behavior often involves *orienteering*: navigating to desired resources via a sequence of steps, instead of attempting to reach the target document directly via a search query [43]. Therefore, post-search browsing behavior provides valuable evidence for identifying documents relevant to users' information goals expressed in preceding search queries.

This paper proposes exploiting a combination of searching and browsing activity of many users to identify relevant resources for future queries. To the best of our knowledge, previous approaches have not considered mining the history of user activity beyond search results, and our experimental results show that comprehensive logs of post-search behavior are an informative source of implicit feedback for inferring resource relevance. We also depart from prior work in that we propose term-based models that generalize to previously unseen queries, which comprise a significant proportion of real-world search engine submissions.

To demonstrate the utility of exploiting collective search and browsing behavior for estimating document authority, we describe several methods that rely on such data to identify relevant websites for new queries. Our initial approach is motivated by heuristic methods used in traditional vector-space information retrieval. Next, we improve on it by employing a probabilistic generative model for documents, queries and query terms, and obtain our best results using a variant of the model that incorporates a simple random-walk modification. Intuitively, all of these algorithms leverage user behavior logs to suggest websites for a new query that were heavily browsed by users after entering similar (or same) queries.

We evaluate the proposed algorithms using an independent dataset of Internet search engine queries for which human judges identified relevant Web pages, as well as an automatically constructed dataset consisting of previously unseen queries. Results demonstrate that relevant websites are identified most accurately when complete post-search browsing trails with associated dwell times are used, compared with using just users' search result clicks, or ignoring dwell times. We also show that a query-term model is

WWW 2008, April 21–25, 2008, Beijing, China.
ACM 978-1-60558-085-2/08/04.

preferable to lookup based on previously entered queries due to a large fraction of queries that are unique. Finally, we demonstrate that post-search browsing behavior can be used to improve learning-based Web search ranking methods by augmenting the output of our algorithms to standard content-, link-, and behavior-based features used in Web search.

The rest of the paper is organized as follows. In Section 2, we review related work in information retrieval and data mining. Section 3 describes the user behavior data and the pre-processing that yields search and browsing paths for users. Section 4 presents our methods for obtaining relevance scoring functions from the data. Experimental evaluation and analysis of the results are provided in Section 5, followed by discussion of future work and conclusions in Sections 6 and 7.

2. RELATED WORK

IR research has a legacy of using term frequencies and term distribution information as the basis for retrieval operations [36]. There are good reason for this: ranking documents based on statistical models of their contents allows developing probabilistic ranking methods (e.g., [24, 35]) that quantify relevance to information needs, formalized as a search query. However, in Web search, sources of evidence beyond contents have also proven to be useful for ranking documents. Reciprocal hyperlinks between Web pages allow authors to link their pages, sites, and repositories to other relevant sources. Link-analysis algorithms leverage this democratic feature of Web page authorship for the implicit endorsement of Web pages. Link-analysis algorithms are generally either: *query-independent*, e.g., PageRank [26], where relative importance of Web pages and Web domains is computed offline prior to query submission, or *query-dependent*, e.g., HITS [23], whereby scores are assigned to documents at retrieval time given their algorithmic matching to the user's query. The key feature of link-analysis algorithms is that they compute the authority value based on the links created by page authors and assume that users traverse this graph in a random or pseudo-intelligent way. However, given the rapid growth in Web usage, it is now possible to leverage the collective browsing behavior of many users as an improvement over random or directed traversals of the Web graph. In this paper we describe the use of collective post-search browsing behavior of many users for this purpose.

Implicit relevance feedback methods [22] use observable aspects of users' search interactions (e.g., query logs, search result clicks, page display times, page scrolling activity) to support more effective search. Given that the users' expression of their true interests and intentions is very noisy, some studies have addressed the reliability of implicit feedback. Kelly and Belkin [21] report that reading time is not indicative of document relevance, and that it varies significantly between subjects and tasks, making it difficult to interpret. In contrast, Fox et al. [12] show in their study that the overall time a user interacts with a search engine, as well as the number of clicks, are indicative of user satisfaction with the search engine. Joachims et al. [18] found that result clickthrough is influenced by the relevance of the results, and that users are biased by the trust they have in the retrieval function, and by the overall quality of the result set. They propose strategies for generating relative feedback signals from clicks. Shen et al. and Tan et al. [39, 42] have employed query logs to enrich query language models by incorporating context obtained from long-term user behavior. White et al. [47] developed a series of custom interfaces designed to elicit more accurate implicit feedback for the current user based on the information items they interact with; this helps the individual user attain their goals. An alternative is to use implicit feedback to capture the dominant intent of many searchers to inform algorithmic design decisions that potentially benefit most search engine users.

Click records from search engines provide weak indications of relevance based on the metadata presented to the user in the result list. These records can be useful as training data for document-ranking algorithms [17, 3], to rank documents when used in isolation [2] or when combined with querying information [30], for docment annotation [48], and query suggestion [6, 20]. However, in recent studies of Web-search behavior [43, 46] it has been shown that a significant proportion of interaction during search sessions is with pages beyond the search engine result list. Implicit feedback algorithms that focus solely on search engine interactions miss out on this potentially valuable information source, reducing their potential effectiveness. Therefore, we hope that leveraging such interactions allows building better ranking algorithms than those based on search engine interactions in isolation. Agichtein et al. [2] used browsing-based features to train a ranking algorithm and showed that search effectiveness improved. However, our present work is significantly different in three core aspects: (i) through applying language modeling approaches, we can generalize to unseen queries rather than memorize behavior for previously seen queries; (ii) we use the entire post-query navigation trail that includes documents that are many clicks from the result page, rather than only the first three documents visited, and (iii) we conduct a more rigorous comparison between using just result clickthrough and post-search browsing behavior.

Several approaches based on machine learning have been proposed for creating adaptive ranking functions that combine many sources of evidence, including those provided by other rankers. Recent examples of such approaches include work by: Burges et al. [7], who developed a ranking algorithm based on neural networks; Richardson et al. [33], who utilize many users' interaction with Web domains to improve their static rank in a way that is independent of hyperlink structure; Agarwal et al. [1], who propose the combination of random walks over the link graph with relevance feedback information, and; Agichtein and Zhang [4], who used classification techniques and machine learning algorithms to incorporate clickthrough-based evidence into the selection of the top-ranked search result. The approach we describe in this paper can be used in conjunction with these and similar methods, providing additional features for a combined ranking that lead to accuracy improvements.

Log-based analysis of browsing patterns within particular Web sites can help understand user needs and intentions, and consequently inform the redesign of site structure to support them [28, 29]. Browse paths followed by human "trail blazers" [8] through information spaces can implicitly represent similarities and associations between visited items that can be incorporated in trail recommendation systems [11]. The approach we describe in this paper is similar in that it uses trails to infer interests, but on a much larger scale and for a different purpose – relevance estimation, rather than trail recommendation for supporting browsing. Wexelblat and Maes [44] describe a system to support within-domain navigation based on the browse trails of other users. In recent work, Pandit and Olston [27] present an information-scent motivated model of navigation-aided retrieval based on a stochastic simulation of browsing behavior. In contrast, our approach is data-driven: we propose learning relevance models from large datasets of user behavior, directly leveraging search and browsing history of real users.

Research in *collaborative filtering* has leveraged explicit and implicit user preference information to recommend items within restricted domains such as newswire [31], music albums and artists [38], or e-commerce [37]. However, to our best knowledge, these techniques have not been directly applied to Web search. In previous work, we leveraged many users' browsing interactions to make page recommendations based on where many other Web searchers with similar needs end up [45]. These *popular destinations* consistently lay at the end of many users' post-query navigation trails and were recommended to users separately from the search results (i.e., in a list of recommendations positioned adjacent to the ranked results). In this paper we describe the use of the trails for domain ranking rather than interactive domain recommendation, and use all pages on the trails rather than just the end points, which we demonstrate to improve results significantly. We propose algorithms for mining the trails, investigate whether the use of interactions beyond the result page adds significant value over search-result clickthrough alone, and determine the utility of adding trail traversal as a learned ranking feature.

3. USER ACTIVITY LOGS

Web browser toolbars have become increasingly popular in recent years, providing users with quick access to extra functionality such as the ability to search the Web without the need to visit a search engine homepage, or the option to search within visited pages for items of interest. Examples of popular toolbars include those affiliated with search engines (e.g., Google Toolbar, Yahoo! Toolbar, and Windows Live Toolbar), as well as those targeted at users with specific interests (e.g., StumbleUpon and eBay Toolbar). To provide the value-added browser features, most popular toolbars log the history of users' browsing behavior on a central server for users who consented to such logging. Each log entry includes an anonymous session identifier, a timestamp, and the URL of the visited Web page.

From these and similar interaction logs, user trails can be reconstructed using the methodology defined by White and Drucker [46]. We extracted such trails from the logs of users of the Windows Live Toolbar. For each user, interaction logs were grouped based on browser identifier information. Within each browser instance, user navigation was summarized as a path known as a *browser trail*, from the first to the last Web page visited in that browser. Located within some of these trails are *search trails* that originated with a query submission to a commercial search engine; it is these search trails that we use to train the algorithms described in the following sections.

After originating with a query submission to a search engine, search trails proceed until a point of termination where it is assumed that the user has completed their information-seeking activity or has addressed a particular aspect of their information need. Trails must contain pages that are either search result pages, or pages connected to a search result page via a sequence of clicked hyperlinks. Extracting search trails using this methodology also goes some way toward handling multi-tasking, where users run multiple searches concurrently. Since users may open a new browser window (or tab) for each task, each task has its own browser trail, and a corresponding distinct search trail.

To reduce the amount of "noise" from pages unrelated to the active search task that may pollute our data, search trails are terminated when one of the following events occurs: (1) user submits a new search query; (2) user navigates to their homepage, initiates a Web-based email session, or visits a page that requires authentication, types a URL or visits a bookmarked page; (3) a page is viewed for more than 30 minutes with no activity; (4) the user closes the active browser window. On average, there are around 5 steps per search trail. To illustrate the concept, we express the search trail as a Web behavior graph [9], an example of which is shown in Figure 1. This graph represents user activity within a search trail, from the originating query to the point at which one of the four termination criteria is met. The nodes of the graph represent Web pages that the user has visited: rectangles represent page views and rounded rectangles represent search engine result pages. Vertical lines represent backtracking to an earlier state. A "back" arrow, such as that below node p_2, implies that the user revisited a page seen earlier in the search trail. Temporal sequence of events continues from left to right, and then from top to bottom.

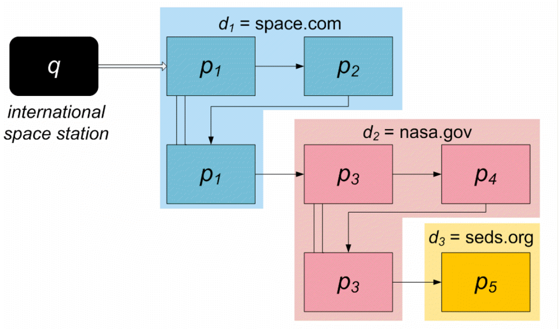

Figure 1: Search trail example

In Figure 1, the trail begins with the the query *[international space station]* submitted to a commercial search engine. From the search engine result page, the user browses to page p_1 in the *space.com* website (d_1), jumps to another page p_2 in the same website, and then returns to the original page p_1. From there, the user follows a link to page p_3 in *nasa.gov* (d_2), then again views a page (p_4) before jumping back to entry point (p_3), from where a link is followed to the homepage of Students for the Development and Exploration of Space (d_3=*seds.org*), where the search trail terminates. This example demonstrates the richness of post-search browsing behavior, which involves navigation across a number of pages in multiple domains over an extended time period.

4. ALGORITHMS

Logs of user behavior data can be transformed into a dataset of search trails: $D = \{q \rightarrow (d_1, \ldots, d_m)\}$ as described in the previous section, where for each search query q an ordered sequence of m documents comprises the trail.[1] Additionally, dwell time $\tau(q \rightsquigarrow d_i)$ can be extracted for every document d_i in the trail using timestamp information. Given that our aim is to exploit this corpus D for identifying relevant resources for future queries, a straightforward approach is to store actual queries along with associated documents, ranking those with highest visitation counts or longest cumulative dwell times the highest. However, we have found that over 60% of queries are unique to a given user session over the twelve-month dataset at our disposal, impeding this approach from working for the majority of incoming queries (comparable proportions of unique queries have been reported in earlier studies of query logs [40, 15]).

Thus, generalizing from past user behavior to new queries requires developing term-based models similar to those that have traditionally been used in standard IR. From here onwards, we assume that every query q can be represented as an unordered set of k terms or phrases, $q = \{t_1, \ldots, t_k\}$, obtained via tokenization and/or additional processing steps that may include token normalization, query expansion, named entity recognition, and construction of n-grams. In the following subsections, we describe several term-based retrieval models that rely on user behavior datasets.

[1] While in this section we refer to each d_i as a "document", it can be a website, a page, a domain, or any other web resource abstraction.

4.1 Heuristic Retrieval Model

First, we consider an ad-hoc model motivated by the empirical success of the *term frequency* × *inverse document frequency* (TF.IDF) heuristic and its variants for traditional content-based retrieval. Based on the search trail corpus D, we construct a vector-space representation for documents, where each document d_i is represented via the agglomeration of *queries* following which the page was visited in the search trails. Every document is thus described as a sparse vector, every non-zero element of which encodes the relative weight of the corresponding term.

Weights in this model must capture the frequency with which users have visited the document following queries containing each term, scaled proportionally to the term's relative specificity across the query corpus. Then, given the search trail corpus D, the component corresponding to term t_j in the vector representing document d_i can be computed as a product of *query-based* term frequency $QTF_{i,j}$ and the term's inverse query frequency IQF_j:

$$w_{d_i,t_j} = QTF_{i,j} \cdot IQF_j =$$
$$= \frac{(\lambda+1)n(d_i,t_j)}{\lambda((1-\beta)+\beta\frac{n(d_i)}{\bar{n}(d_i)})+n(d_i,t_j)} \cdot \log\frac{N_d - n(t_j)+0.5}{n(t_j)+0.5}$$

where:

- λ and β are smoothing parameters; while in this work we use $\lambda = 0.5$ and $\beta = 0.75$, we found that results are relatively robust to the choice of specific values;
- $n(d_i,t_j) = \sum_{q \rightsquigarrow d_i, t_j \in q} f(q \rightsquigarrow d_i)$ is the term frequency aggregated over all trails that begin with queries containing term t_j and include document d_i, where the aggregation is performed via the feature function $f(q \rightsquigarrow d_i)$ computed for the document from each trail;
- $n(d_i)$ is the total number of terms in all queries followed by search trails that include document d_i;
- $\bar{n}(d_i)$ is the average value of $n(d_i)$ over all documents in D;
- $n(t_j)$ is the number of documents for which queries leading to them include the term t_j;
- N_d is total number of documents.

This formula is effectively an adaptation of the BM25 scoring function, which is a variant of the traditional TF.IDF heuristic that has provided good performance on a number of retrieval benchmarks [34]. To instantiate the term frequencies computed from all trails leading from queries containing the term to the document, $n(d_i, q_j)$, different instantiations of the feature function $f(q \rightsquigarrow d_i)$ are possible that weigh the contribution of each particular trail. In this work, we consider three variants of this feature function:

- Raw visitation count: $f(q \rightsquigarrow d_i) = 1$;
- Dwell time: $f(q \rightsquigarrow d_i) = \tau(q \rightsquigarrow d_i)$, where $\tau(q \rightsquigarrow d_i)$ is the total dwell time for document d_i in this particular trail;
- Log of dwell time: $f(q \rightsquigarrow d_i) = \log \tau(q \rightsquigarrow d_i)$.

Given the large size of typical search trail datasets D, computation of the document vectors can be performed efficiently in several passes over the data for term and document index construction, term-document frequency computation, and final estimation of document-term scores.

Given a new query $\hat{q} = \{\hat{t}_1, \ldots, \hat{t}_k\}$, candidate documents are retrieved from the inverted index and their relevance is computed via the dot product between the document and query vectors:

$$Rel_H(d_i, \hat{q}) = \sum_{\hat{t}_j \in \hat{q}} w_{d_i,\hat{t}_j} \cdot w_{\hat{t}_j} \qquad (1)$$

where $w_{\hat{t}_j}$ is the relative weight of each term in the query, computed using *inverse query frequency* over the set of all queries in the dataset: $w_{\hat{t}_j} = \log\frac{N_q - n(\hat{t}_j)+0.5}{n(\hat{t}_j)+0.5}$, with N_q and $n(\hat{t}_j)$ being the total number of queries and the number of queries containing term $\hat{t}_j$, respectively.

4.2 Probabilistic Retrieval Model

Statistical approaches to content-based information retrieval have been considered alongside heuristic methods for several decades, and have attracted increasing attention recently. Besides theoretical elegance, probabilistic retrieval models provide competitive performance and can be used to explain the empirical success of the heuristics (e.g., a number of papers have proposed generative interpretations of the IDF heuristic). Hence, we employ a statistical framework to formulate an alternative approach for retrieving documents most relevant to a given query, provided a large dataset of users' past searching and browsing behavior.

We consider a generative model for queries, terms, and documents, where every query q instantiates a multinomial distribution over its terms. Every term in the vocabulary is in turn associated with a multinomial distribution over the documents, which can be viewed as the likelihood of a user browsing to the document after submitting a query that contains the term (or the likelihood of the user viewing the document per unit time, depending on the particular instantiation of the distribution). In effect, this probability encodes the topical relevance of the document for the particular term. Then, the probability of selecting document d_i given a new query $\hat{q}$ can be used to estimate the document's relevance:

$$Rel_P(d_i, \hat{q}) = p(d_i|\hat{q}) = \sum_{\hat{t}_j \in q} p(\hat{t}_j|\hat{q})p(d_i|\hat{t}_j) \qquad (2)$$

Selecting particular parameterizations for the query-term distribution $p(\hat{t}_j|\hat{q})$ and the distribution over documents for a given term $p(d_i|\hat{t}_j)$ allows instantiating different retrieval functions. In this work, we estimate the instantaneous multinomial query-term likelihood $p(\hat{t}_j|\hat{q})$ via a negative exponentiated term prior estimated from the query corpus, which has an effect similar to IDF weighting: less frequent terms have higher influence on selection of documents:

$$p(\hat{t}_j|\hat{q}) = \frac{\exp(-p(\hat{t}_j))}{\sum_{\hat{t}_l \in \hat{q}} \exp(-p(\hat{t}_l))} = \frac{\exp(-\frac{n(\hat{t}_j)+\mu}{\sum_{\hat{t}_s \in D} n(\hat{t}_s)+\mu})}{\sum_{\hat{t}_l \in \hat{q}} \exp(-\frac{n(\hat{t}_l)+\mu}{\sum_{\hat{t}_s \in D} n(\hat{t}_s)+\mu})} \qquad (3)$$

where $n(\hat{t}_j)$ is the number of queries containing term $\hat{t}_j$, as in previuos subsection, and μ is a smoothing constant (we set $\mu = 10$ in all experiments; alternative smoothing approaches are possible, e.g., those used in language modeling [49]).

Document probabilities for every query term can be estimated as maximum-likelihood estimates over the training data for all paths in D that originate with queries containing $\hat{t}_j$. Using Laplace smoothing for more robust estimation, term-document probabilities become:

$$p(d_i|\hat{t}_j) = \frac{n(d_i,\hat{t}_j)}{\sum_{d_l \in D} n(d_l,\hat{t}_j)} \qquad (4)$$

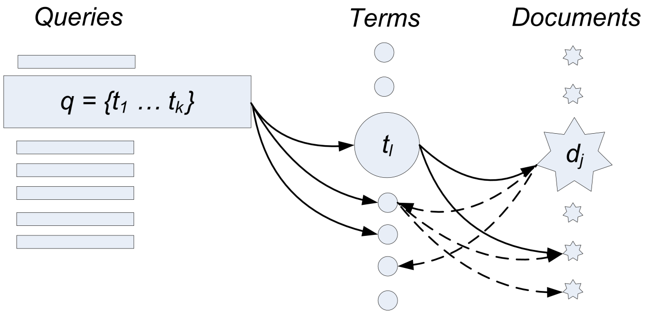

Figure 2: Random walks for search trails

where $n(d_i, \hat{t}_j) = \sum_{q \rightsquigarrow d_i s.t. \hat{t}_j \in q} f(q \rightsquigarrow d_i)$ is again the aggregated count for document d_i reached from queries containing term $\hat{t}_j$. As above, different instantiations of this count (raw visitation counts, dwell time, etc.) can be used, corresponding to different semantics of the probability distribution (likelihood of reaching the document, likelihood of viewing the document per unit time, etc.).

The overall probabilistic formulation is somewhat analogous to the heuristic model described above, in that it aggregates the lists of most relevant documents for each query term, weighing them proportionally to the term's relative prominence.

4.3 A Random-Walk Extension

The process of selecting a document relevant to a query in the probabilistic model described in the previous section can be viewed as a two-step random walk in a tri-partite graph formed by queries, query terms, and documents. Figure 2 illustrates this view with solid lines representing the transitions corresponding to the query-term and term-document probability distributions. Then, the probability of reaching a document starting from a given query is the likelihood of hitting the document node via the two-step random walk that originates at the query node and proceeds via the term nodes, with transition probabilities from the query node to term nodes materializing instantaneously at query time as described by Eq.(3).

This view suggests that the basic model described in the previous section can be enhanced by considering random walks beyond two steps. For computational efficiency, we only consider a simple enhancement that adds four-step walks alongside the two-step walks in the basic probabilistic model above: in Figure 2, these are represented by dotted lines that go back to term nodes from document nodes and then return to document nodes. After reaching a document in the second step of the random walk from the standard model, the walk is either absorbed with probability α, or proceeds to sample from all terms via which the document was reached, and continues to other documents reached from these terms. Then, relevance of a document d_i for a given query $\hat{q}$ is computed via the likelihood of the random walk ending in node d_i:

$$Rel_{P+RW}(d_i, \hat{q}) = \tag{5}$$

$$\sum_{\hat{t}_j \in q} p(\hat{t}_j|\hat{q}) \Bigg(\alpha p(d_i|\hat{t}_j) + (1-\alpha) \sum_{\hat{t}_l \in \hat{q}, d_j} p(d_j|\hat{t}_j) p(\hat{t}_l|d_j) p(d_i|\hat{t}_l) \Bigg)$$

While the computational cost of this formulation appears formidable, the factor in parentheses does not depend on the query and hence can be pre-computed in advance. Intuitively, this model produces higher scores for documents that are reached from multiple terms in the query, as well as increases the scores of documents that are reached via related terms (where the measure of term similarity is effectively the likeness of their document distributions; exploring information-theoretic implications and extensions of this approach is an interesting direction for future work).

5. EXPERIMENTAL EVALUATION

The primary contribution of this paper is demonstrating that post-search browsing behavior logs provide strong signal for inferring document relevance for future queries. To validate this hypothesis empirically, we employ methods described in the previous section to identify relevant websites for two sets of real-world queries (identifying relevant websites vs. relevant pages is discussed in Section 5.2). For the first set, truly relevant webpages are labeled by human judges, while for the second set, relevant websites are computed for novel queries based on search trails from a time period that is separated from training data by several months. Finally, we demonstrate that the proposed methods can aid a powerful supervised algorithm for learning to rank by providing a feature that augments traditional attributes such as those extracted from document contents, link structure, and search engine interactions.

5.1 Methodology

Evaluation of retrieval accuracy is known to be a difficult task on Web scale. A number of alternative evaluation methodologies and metrics have been proposed in the IR community, e.g. [41, 16]. We perform two groups of experiments to validate the feasibility of identifying relevant resources from user activity logs:

- Ranking accuracy: given a dataset of queries along with a list of websites that have relevance ratings, the ranking produced by our methods is compared to the target ranking inferred from relevance ratings.
- Feature for learning to rank: scores produced by our methods are used as a feature for RankNet [7], a supervised learning algorithm that utilizes hundreds of features to learn ranking functions for Web search.

When evaluating ranking accuracy, experiments are conducted at the website level, leaving aside the issue of relevance of individual pages. While investigating the utility of the proposed methods for estimating page-specific relevance is an open challenge, site-level estimates are an important component in a number of retrieval tasks, e.g., dynamic re-ranking of search results [25] and static ranking [33]; we also note that popular search engines currently collapse results from the same website when they are presented to the user on a search result page. However, we employ page-level rankings in learning-to-rank evaluation presented in Section 5.4, providing a measure of the utility of our site-level relevance features.

To compare the agreement between the rankings, we employ Normalized Discounted Cumulative Gain (NDCG) [16]. NDCG is defined as follows for every rank position i in the target ranking:

$$NDCG(i) = N_i \sum_i \frac{2^{r(d_i)} - 1}{\log(1+i)}$$

where $r(d_i)$ is the relevance score of document d_i assigned to position i in the ranking, and N_i is a normalization factor. If a result d_i is not present in the target ranking, the corresponding $r(d_i) = 0$. NDCG has been increasingly popular for evaluating web retrieval because it is highly sensitive to the accuracy of top-ranked results, and can be aggregated across queries with a varying number of results and/or target items, while still producing a value between 0 and 1. For all ranking accuracy experiments, we randomly separate the test sets of queries and their target rankings into ten folds, providing for statistical significance testing.

5.2 Datasets

A training dataset of approximately 140 million search trails covering the 12-month period from January to December 2006 was extracted as described in Section 3 from a random sample of several hundred thousand consenting toolbar users. For testing, we employed two query datasets constructed as follows. The first dataset, *HumanRanking*, contains 33,150 queries randomly sampled based on their relative frequency from query logs of Windows Live Search. For each query, a number of webpages were evaluated by human judges on a five-point relevance scale with grades ranging from *Bad* to *Perfect*. The corresponding websites are assigned the highest relevance score of any page within them.

The second dataset, *UsageRanking*, was constructed automatically by sampling 10,000 queries not seen in training data from search trails observed in May 2007, where each query must have been entered into at least two different search engines. For each query, all sites that were visited in trails following the query were aggregated, and ranked by the total number of page views from distinct users over all search trails. Position in the resulting ranking reflects each site's popularity with users who entered the query, thus representing empirically-relevant resources. Although the resulting ranking is biased as it is obtained from the the same source as the training data, using novel queries and results from multiple search engines provides a good-faith effort to limit the advantages that simple memorization via recovering popular sites from previously seen queries would provide to the algorithms. Relevant results for every query in *HumanRanking* and *UsageRanking* datasets are sorted, resulting in the target ranking, with respect to which predictions of the algorithms are evaluated as described above. Actual relevance scores are used as $r(i)$ for *HumanRanking*, while for *UsageRanking* rank-based scores are used that incrementally decrease for every position in the target ranking (e.g., in a five-item target ranking the top-most site has the relevance score $r(i) = 5$, while the bottom-most site has $r(i) = 1$).

While the methods we described in Section 4 can be applied to page-level scoring, the trails dataset at our disposal does not have sufficient coverage of all pages in the web index, while the coverage of websites is sufficient. However, site-level scoring is a core component of many applications, e.g., learning to rank, spam filtering, and crawl prioritization. As results in Section 5.4 demonstrate, our methods applied at website-level improve the accuracy of learning to rank at page level.

5.3 Ranking Accuracy

Ranking accuracy experiments evaluate the quality of relevance predictions produced by our methods in isolation, which allows studying the effects of such factors as the utility of a term-based model versus query-level memorization, importance of the amount of available training data, and the utility of incorporating post-search browsing data beyond search result selections.

5.3.1 Utilty of the Query-Term Model

Since all proposed methods rely on term-based models, we first compare their performance with a baseline that does not split queries into terms, but rather associates complete queries with the websites that were browsed in subsequent trails. This baseline effectively treats each query as a single term, for which websites are aggregated. This strategy was employed in previous work of Agichtein et al. [2], who considered using the history of search and browsing behavior to extract usage-based features for individual queries.

Table 5.3.1 presents the comparison of the three different methods proposed in Section 4 with the query-based baseline implemented via the probabilistic method augmented with random walks; *HumanRanking* dataset is used since query lookup is not feasible for novel queries in the *UsageRanking* dataset. The lookup-based approach performs much worse than the three term-based methods, which is explained by the fact that a significant proportion of queries are novel, and behavior-based retrieval is impossible for them without splitting into terms. These results also demonstrate the relative performance of the three proposed methods trained on the complete training set. Heuristic model performs the worst of the three for all NDCG levels, while the enhanced probabilistic method that incorporates random walks provides small but consistent improvements over the basic probabilistic model. Differences in NDCG scores between the top-performing method (Rel_{P+RW}) and the other approaches are statistically significant at the 0.05 level.

Method	NDCG@1	NDCG@3	NDCG@10
Query Lookup	0.220	0.200	0.212
Heuristic	0.311	0.279	0.278
Probabilistic	0.313	0.288	0.288
Probabilistic+RW	0.317	0.292	0.293

Table 1: Query lookup vs. Term-based methods

5.3.2 Utility of Full Trails

Previous research has considered using either the starting points of search trails – logs of search engine queries and subsequent clicks on results [17, 2, 4], or just the end points of search trails, also known as search destinations [45]. To investigate the usefulness of exploiting full search trails, we compared NDCG scores obtained on the *HumanRanking* dataset with full browsing trails versus those obtained using either just the starting points (search result clicks), or just the end points of the trails (search destinations). Table 2 summarizes the results of these experiments for the three methods, which again were trained on the entire available set of search trails.

These results show that for all methods, using the full navigational data contained in search trails leads to better performance: taking into account *all* sites visited by users provides more data to the models, yielding more accurate relevance predictions. It is also important to note that higher scores are obtained when the end-points of search trails are used for training versus the starting points, which shows that search destinations capture the resources relevant to users' information needs more accurately that initial clicks on search results, which are biased by suggestions of search engines that may be suboptimal.

5.3.3 Dwell time vs. Visitation Counts

The validity of considering dwell times as an indicator of page relevance or user satisfaction during search engine interactions was debated in previous work [21, 12]. Since all methods proposed in Section 4 rely on function $f(q \rightsquigarrow d_i)$ that can capture either dwell times or raw pageview counts, we compared the NDCG scores obtained using the following three variants of f:

- Raw visitation count: $f(q \rightsquigarrow d_i) = 1$;
- Dwell time: $f(q \rightsquigarrow d_i) = \tau(q \rightsquigarrow d_i)$, where $\tau(q \rightsquigarrow d_i)$ is the total dwell time for document d_i for the particular trail;
- Log of dwell time: $f(q \rightsquigarrow d_i) = \log \tau(q \rightsquigarrow d_i)$.

Results in Table 3 demonstrate that using logs of dwell times provides best performance among the three options above: while dwell times carry some information, smoothing it by taking logarithms

	Heuristic			Probabilistic			Probabilistic-RW		
	NDCG@1	NDCG@3	NDCG@10	NDCG@1	NDCG@3	NDCG@10	NDCG@1	NDCG@3	NDCG@10
Full Trails	0.311	0.279	0.278	0.313	0.288	0.288	0.317	0.292	0.293
Result Clicks	0.297	0.267	0.268	0.295	0.273	0.276	0.296	0.274	0.277
Destinations	0.301	0.271	0.273	0.305	0.279	0.283	0.310	0.287	0.289

Table 2: Full Search Trails vs. Start and End Points

	Heuristic			Probabilistic			Probabilistic-RW		
	NDCG@1	NDCG@3	NDCG@10	NDCG@1	NDCG@3	NDCG@10	NDCG@1	NDCG@3	NDCG@10
Log(Dwell Time)	0.311	0.279	0.278	0.313	0.288	0.288	0.317	0.292	0.293
Dwell Time	0.297	0.267	0.262	0.292	0.271	0.271	0.302	0.278	0.281
Count	0.297	0.267	0.266	0.287	0.268	0.273	0.296	0.275	0.277

Table 3: Dwell Times vs. Visitation Counts

yields higher accuracy since the effect of outliers is reduced, while some differentiation is maintained, effectively providing middle ground between raw times and visitation counts.

5.3.4 Impact of Training Data Availability

To assess the influence that the amount of training data has on relevance predictions, we evaluate the performance of the proposed methods based on varying amounts of training data. Learning curves in Figure 3 illustrate that access to large datasets of user behavior information is essential for obtaining good performance. While this is expected given that the distribution of query frequencies follows a power law and is therefore very sparse, these experiments demonstrate that datasets of at least 100 million trails are preferable for maximizing the accuracy of ranking predictions based on logs of user behavior. The results on the *UsageRanking* dataset demonstrate the generalization capability of the proposed methods: even though none of the queries in this dataset were observed during training, the term-based model is sufficiently robust to identify most heavily browsed sites for individual terms, which are then combined by the query-term model.

Overall, these results show that best performance is obtained using the method based on probabilistic prediction augmented with random walks for all amounts of training data, thus justifying the additional upfront computational effort that this method requires.

5.4 Learning To Rank

While the results in the previous section demonstrate that the proposed models are capable of leveraging large datasets of user search and browsing behavior to identify relevant websites for queries, they do not address the issue of practical usefulness of the methods in the context of improving search engine results. Modern search engines typically rely on ranking algorithms based on machine learning approaches, which allow incorporating hundreds and thousands of features that exploit diverse sources of evidence [7, 4]. These features may capture such signals as similarity between the query and document content, link structure and properties such as anchor text, overall page quality, and features derived from user interactions with the search engine. Therefore, to validate whether our proposed methods can aid actual web search ranking, we conducted experiments where RankNet, a supervised learning algorithm, was used to train a ranking function.

RankNet is based on a 2-layer neural net algorithm that is trained using pairwise relevance preferences with the aim of optimizing NDCG [7]. Experiments were conducted using the *HumanRanking* dataset described above containing 33,150 queries randomly sampled from query logs and accompanied by sets of webpages that were labeled by human judges on a five-point relevance scale. The dataset was partitioned into training, validation, and test subsets in 10:1:1 proportion, providing a test set of 2,762 queries. The baseline for the experiments used a large number of features that incorporate contents-based and link-based features, as well as features based on interactions with seach engine results (clickthrough). We evaluated the utility of our methods by augmenting the baseline features with relevance scores produced by our methods via Equations (1), (2), and (5), as well as their binary and log-based transforms.

Feature Set	NDCG@1	NDCG@3	NDCG@10
Baseline	0.622	0.635	0.691
Baseline+Rel_H	0.625	0.638	**0.695**
Baseline+Rel_P	**0.628**	**0.641**	**0.696**
Baseline+Rel_{P+RW}	**0.631**	**0.643**	**0.696**

Table 4: Learning to Rank with Trail-Based Features

Table 5.4 shows the results obtained from these experiments, where bold font indicates that improvements over the baseline are statistically significant according to a two-tailed t-test ($p < 0.05$). The results demonstrate that predictions produced by our methods lead to NDCG improvements when learning to rank.Gains are most pronounced at NDCG@1 and NDCG@3, where the enhanced probabilistic model leads to an almost single-point improvement in NDCG[2] The more modest improvements at NDCG@10 are explained by the fact that our methods are most accurate at identifying the top few authoritative websites that typically correspond to the few top results, while lesser-ranked sites are less useful when ordering results at lower ranks. We note that improvements are obtained over a baseline that already incorporates clickthrough-based features, demonstrating the utility of mining the browsing behavior from complete search trails over using just the logs of search result clicks.

6. FUTURE WORK

Our aim in this work was to obtain proof-of-concept results that showcase the unique benefits provided by combining complete trails of search and browsing data, and the methods we have proposed can be extended and improved in several directions. Alternative derivations of relevance functions based on training datasets of search

[2] NDCG is colloquially measured in percentage points.

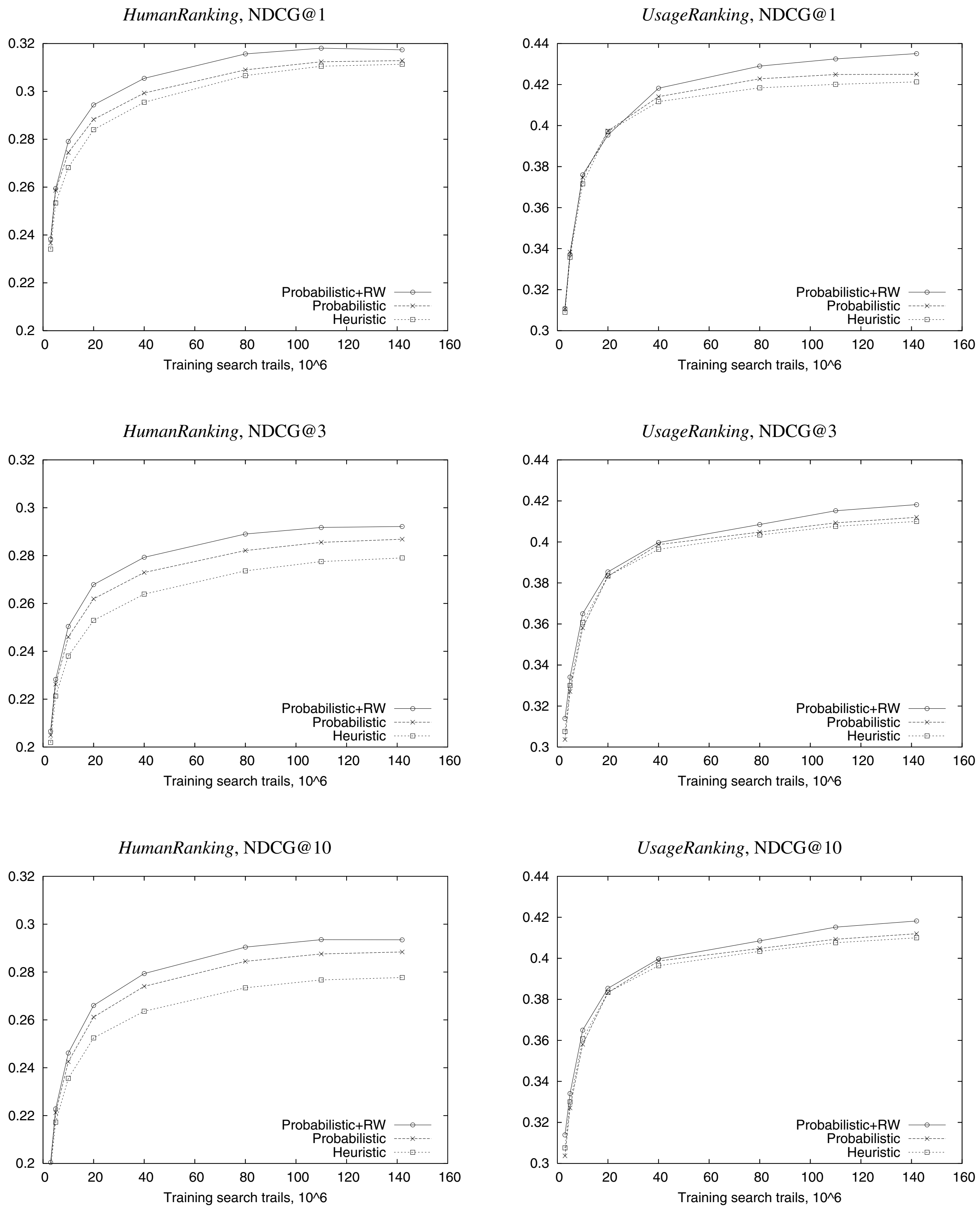

Figure 3: Ranking accuracy for varying amounts of training data

trails can be constructed both heuristically, as well as using different probabilistic formulations. In particular, exploration of language modeling techniques for defining more advanced query-term distributions that exploit document contents as well as query logs [49, 39, 42] is a promising avenue for future research, as is experimenting with variants of the random-walk formulation that employ transition schemes different from one used in Section 4.3.

On the application side, there is a number of tasks that can exploit query-specific document authority, transcending relevance estimation for web search. For example, it has been shown that user-validated authority may be useful for identification of web spam [13]. Because users are unlikely to visit non-informative resources often, and will leave them almost immediately, using activity logs may provide valuable evidence to web spam detection algorithms, leaving an interesting avenue for future work.

Finally, unlike “random surfer” or “directed surfer” models previously exploited by algorithms from the PageRank family [26, 14, 32], browsing behavior of real users provides an empirical distribution of walks over the web graph. Deriving graph-theoretic ranking algorithms based on traversal models learned from search and browsing behavior of real users is an exciting challenge for future work, and may lead to improved algorithms for static ranking and its applications, such as crawl prioritization.

7. CONCLUSIONS

We have proposed and evaluated heuristic and probabilistic algorithms for identifying relevant websites using the combined history of searching and browsing behavior of many Web users. The algorithms leverage implicit feedback from users’ post-search browsing activity, allowing us to estimate the topical relevance of Web sites. Through experimental evaluation we have shown that (i) training retrieval algorithms on interaction behavior from navigation trails following search engine result click-through leads to improved retrieval accuracy over training on only result click-through or search destinations, and (ii) exploiting aspects of navigation trail interaction as learned ranking feature in a web search ranking algorithm improves retrieval effectiveness. In addition, we have also demonstrated the importance of generalizable learning over query term lookup, the utility of the logarithmic transform of site dwell time over raw site dwell time and visitation counts, and the value of increased volumes of training data. Our research has profound implications for the design of Web search ranking algorithms and the improvement of the search experience for all search engine users, and we hope that it will encourage further work that leverages large datasets of search and browsing behavior.

8. REFERENCES

[1] A. Agarwal, S. Chakrabarti, and S. Aggarwal. Learning to rank networked entities. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-06)*, pages 14–23, 2006.

[2] E. Agichtein, E. Brill, and S. Dumais. Improving web search ranking by incorporating user behavior information. In *Proceedings of the 29th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR-06)*, pages 19–26, 2006.

[3] E. Agichtein, E. Brill, S. Dumais, and R. Ragno. Learning user interaction models for predicting web search result preferences. In *Proceedings of the 29th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR-06)*, pages 3–10, 2006.

[4] E. Agichtein and Z. Zheng. Identifying "best bet" web search results by mining past user behavior. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-06)*, pages 902–908, 2006.

[5] S. Agrawal, C. Cortes, and R. Herbrich, editors. *Proceedings of the NIPS 2005 Workshop on Learning to Rank*, 2005. http://web.mit.edu/shivani/www/Ranking-NIPS-05.

[6] D. Beeferman and A. Berger. Agglomerative clustering of a search engine query log. In *Proceedings of the 6th International Conference on Knowledge Discovery and Data Mining (KDD-00)*, pages 407–416, 2000.

[7] C. J. C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. N. Hullender. Learning to rank using gradient descent. In *Proceedings of 22nd International Conference on Machine Learning (ICML-2005)*, pages 89–96, 2005.

[8] V. Bush. As we may think. *Atlantic Monthly*, 3(2):37–46, 1945.

[9] S. K. Card, P. Pirolli, M. V. D. Wege, J. B. Morrison, R. W. Reeder, P. K. Schraedley, and J. Boshart. Information scent as a driver of web behavior graphs: results of a protocol analysis method for web usability. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*, pages 498–505, 2001.

[10] S. Chakrabarti, B. Dom, P. Raghavan, S. Rajagopalan, D. Gibson, and J. Kleinberg. Automatic resource compilation by analyzing hyperlink structure and associated text. In *Proceedings of the 7th International World Wide Web Conference (WWW-98)*, pages 65–74, 1998.

[11] M. Chalmers, K. Rodden, and D. Brodbeck. The order of things: activity-centered information access. In *Proceedings of the 7th International Conference on the World Wide Web (WWW-98)*, pages 359–367, 1998.

[12] S. Fox, K. Karnawat, M. Mydland, S. Dumais, and T. White. Evaluating implicit measures to improve web search. *ACM Transactions on Information Systems*, 23(2):147–168, 2005.

[13] Z. Gyöngyi, H. Garcia-Molina, and J. Pedersen. Combating web spam with trustrank. In *Proceedings of the 30th International Conference on Very Large Databases (VLDB-04)*, pages 576–587, 2004.

[14] T. Haveliwala. Topic-sensitive PageRank. In *Proceedings of the 11th International World Wide Web Conference (WWW-02)*, pages 517–526, 2002.

[15] B. J. Jansen and A. Spink. How are we searching the world wide web?: a comparison of nine search engine transaction logs. *Information Processing and Management*, 42(1):248–263, 2006.

[16] K. Järvelin and J. Kekäläinen. Cumulated Gain-based evaluation of IR techniques. *ACM Transactions on Information Systems*, 20(4):422–446, 2002.

[17] T. Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-02)*, Edmonton, Canada, 2002.

[18] T. Joachims, L. A. Granka, B. Pan, H. Hembrooke, and G. Gay. Accurately interpreting clickthrough data as implicit feedback. In *Proceedings of the 29th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR-06)*, pages 154–161, 2006.

[19] T. Joachims, H. Li, T.-Y. Liu, and C. Zhai, editors. *Proceedings of the ACM SIGIR 2007 Workshop on Learning*

to Rank for Information Retrieval, 2007. http://research.microsoft.com/users/LR4IR-2007.

[20] R. Jones, B. Rey, O. Madani, and W. Greiner. Generating query substitutions. In *Proceedings of the 16th International Conference on World Wide Web (WWW-2006)*, pages 387–396, 2006.

[21] D. Kelly and N. J. Belkin. Display time as implicit feedback: understanding task effects. In *Proceedings of the 27th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR-04)*, pages 377–384, 2004.

[22] D. Kelly and J. Teevan. Implicit feedback for inferring user preference: a bibliography. *ACM SIGIR Forum*, 37(2):18–28, 2003.

[23] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the Association for Computing Machinery*, 46(5):604–632, 1999.

[24] M. E. Maron and J. L. Kuhns. On relevance, probabilistic indexing and information retrieval. *Journal of the ACM*, 7(3):216–244, 1960.

[25] I. Matveeva, C. Burges, T. Burkard, A. Laucius, and L. Wong. High accuracy retrieval with multiple nested ranker. In *Proceedings of the 29th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR-06)*, pages 437–444, 2006.

[26] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank citation ranking: Bringing order to the Web. Technical report, Stanford University, 1998.

[27] S. Pandit and C. Olston. Navigationaided retrieval. In *Proceedings of the 16th International World Wide Web Conference (WWW-07)*, pages 391–400, 2007.

[28] P. Pirolli, J. Pitkow, and R. Rao. Silk from a sow's ear: extracting usable structures from the web. In *Proceedings of the ACM SIGCHI conference on Human Factors in Computing Systems*, pages 118–125, 1996.

[29] J. Pitkow and P. Pirolli. Life, death, and lawfulness on the electronic frontier. In *Proceedings of the ACM SIGCHI conference on Human factors in computing systems*, pages 383–390, 1997.

[30] F. Radlinsky and T. Joachims. Query chains: Learning to rank from implicit feedback. In *Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-05)*, pages 239–248, 2005.

[31] P. Resnick, N. Iacovou, M. Sushak, P. Bergstrom, and J. Reidl. GroupLens: An open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 Computer Supported Cooperative Work Conference*, New York, 1994. ACM.

[32] M. Richardson and P. Domingos. The Intelligent Surfer: Probabilistic combination of link and content information in PageRank. In *Advances in Neural Information Processing Systems 14*, pages 1441–1448, 2002.

[33] M. Richardson, A. Prakash, and E. Brill. Beyond PageRank: machine learning for static ranking. In *Proceedings of the 15th International World Wide Web Conference (WWW-06)*, pages 707–715, 2006.

[34] S. Robertson, H. Zaragoza, and M. Taylor. Simple BM25 extension to multiple weighted fields. In *Proceedings of the 13th ACM International Conference on Information and Knowledge Management (CIKM-04)*, pages 42–49, 2004.

[35] S. E. Robertson. The probability ranking principle in IR. In *Readings in Information Retrieval*, pages 281–286. Morgan Kaufmann Publishers Inc., 1997.

[36] G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval*. McGraw Hill, New York, 1983.

[37] B. M. Sarwar, G. Karypis, J. A. Konstan, and J. Riedl. Analysis of recommendation algorithms for e-commerce. In *ACM Conference on Electronic Commerce*, pages 158–167, 2000.

[38] U. Shardanand and P. Maes. Social information filtering: algorithms for automating word of mouth. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 210–217, 1995.

[39] X. Shen, B. Tan, and C. Zhai. Context-sensitive information retrieval using implicit feedback. In *Proceedings of the 28th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR-05)*, pages 43–50, 2005.

[40] C. Silverstein, H. Marais, M. Henzinger, and M. Moricz. Analysis of a very large web search engine query log. *SIGIR Forum*, 33(1):6–12, 1999.

[41] I. Soboroff, C. Nicholas, and P. Cahan. Ranking retrieval systems without relevance judgments. In *Proceedings of the 24th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR-01)*, pages 66–73, 2001.

[42] B. Tan, X. Shen, and C. Zhai. Mining long-term search history to improve search accuracy. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-06)*, pages 718–723, 2006.

[43] J. Teevan, C. Alvarado, M. S. Ackerman, and D. R. Karger. The perfect search engine is not enough: a study of orienteering behavior in directed search. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI-04)*, pages 415–422, 2004.

[44] A. Wexelblat and P. Maes. Footprints: history-rich tools for information foraging. In *Proceedings of the ACM SIGCHI conference on Human factors in computing systems*, pages 270–277, 1999.

[45] R. W. White, M. Bilenko, and S. Cucerzan. Studying the use of popular destinations to enhance web search interaction. In *Proceedings of the 30th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR-07)*, pages 159–166, 2007.

[46] R. W. White and S. M. Drucker. Investigating behavioral variability in web search. In *Proceedings of the 16th International Conference on World Wide Web (WWW-2006)*.

[47] R. W. White, I. Ruthven, and J. M. Jose. Finding relevant documents using top ranking sentences: an evaluation of two alternative schemes. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 57–64, 2002.

[48] G.-R. Xue, H.-J. Zeng, Z. Chen, Y. Yu, W.-Y. Ma, W. Xi, and W. Fan. Optimizing web search using web click-through data. In *Proceedings of the 13th ACM International Conference on Information and Knowledge Management (CIKM-04)*, pages 118–126, 2004.

[49] C. Zhai and J. Lafferty. A study of smoothing methods for language models. *ACM Transactions on Information Systems*, 22(2):179–214, 2004.