Improving Web Search Ranking by Incorporating User Behavior Information

Eugene Agichtein
Microsoft Research
eugeneag@microsoft.com

Eric Brill
Microsoft Research
brill@microsoft.com

Susan Dumais
Microsoft Research
sdumais@microsoft.com

ABSTRACT

We show that incorporating user behavior data can significantly improve ordering of top results in real web search setting. We examine alternatives for incorporating feedback into the ranking process and explore the contributions of user feedback compared to other common web search features. We report results of a large scale evaluation over 3,000 queries and 12 million user interactions with a popular web search engine. We show that incorporating implicit feedback can augment other features, improving the accuracy of a competitive web search ranking algorithms by as much as 31% relative to the original performance.

Categories and Subject Descriptors

H.3.3 Information Search and Retrieval – *Relevance feedback, search process*; H.3.5 Online Information Services – *Web-based services*.

General Terms

Algorithms, Measurement, Experimentation

Keywords

Web search, implicit relevance feedback, web search ranking.

1. INTRODUCTION

Millions of users interact with search engines daily. They issue queries, follow some of the links in the results, click on ads, spend time on pages, reformulate their queries, and perform other actions. These interactions can serve as a valuable source of information for tuning and improving web search result ranking and can compliment more costly explicit judgments.

Implicit relevance feedback for ranking and personalization has become an active area of research. Recent work by Joachims and others exploring implicit feedback in controlled environments have shown the value of incorporating implicit feedback into the ranking process. Our motivation for this work is to understand how implicit feedback can be used in a large-scale operational environment to improve retrieval. How does it compare to and compliment evidence from page content, anchor text, or link-based features such as inlinks or PageRank? While it is intuitive that user interactions with the web search engine should reveal at least *some* information that could be used for ranking, estimating user preferences in real web search settings is a challenging problem, since real user interactions tend to be more "noisy" than commonly assumed in the controlled settings of previous studies.

Our paper explores whether implicit feedback can be helpful in realistic environments, where user feedback can be noisy (or adversarial) and a web search engine already uses hundreds of features and is heavily tuned. To this end, we explore different approaches for ranking web search results using real user behavior obtained as part of normal interactions with the web search engine.

The specific contributions of this paper include:

- Analysis of alternatives for incorporating user behavior into web search ranking (Section 3).
- An application of a robust implicit feedback model derived from mining millions of user interactions with a major web search engine (Section 4).
- A large scale evaluation over real user queries and search results, showing significant improvements derived from incorporating user feedback (Section 6).

We summarize our findings and discuss extensions to the current work in Section 7, which concludes the paper.

2. BACKGROUND AND RELATED WORK

Ranking search results is a fundamental problem in information retrieval. Most common approaches primarily focus on similarity of query and a page, as well as the overall page quality [3,4,24]. However, with increasing popularity of search engines, implicit feedback (i.e., the actions users take when interacting with the search engine) can be used to improve the rankings.

Implicit relevance measures have been studied by several research groups. An overview of implicit measures is compiled in Kelly and Teevan [14]. This research, while developing valuable insights into implicit relevance measures, was not applied to improve the ranking of web search results in realistic settings.

SIGIR'06, August 6–11, 2006, Seattle, Washington, USA.

Closely related to our work, Joachims [11] collected implicit measures in place of explicit measures, introducing a technique based entirely on clickthrough data to learn ranking functions. Fox et al. [8] explored the relationship between implicit and explicit measures in Web search, and developed Bayesian models to correlate implicit measures and explicit relevance judgments for both individual queries and search sessions. This work considered a wide range of user behaviors (e.g., dwell time, scroll time, reformulation patterns) in addition to the popular clickthrough behavior. However, the modeling effort was aimed at predicting explicit relevance judgments from implicit user actions and not specifically at learning ranking functions. Other studies of user behavior in web search include Pharo and Järvelin [19], but were not directly applied to improve ranking.

More recently, Joachims et al. [12] presented an empirical evaluation of interpreting clickthrough evidence. By performing eye tracking studies and correlating predictions of their strategies with explicit ratings, the authors showed that it is possible to accurately interpret clickthroughs in a controlled, laboratory setting. Unfortunately, the extent to which previous research applies to real-world web search is unclear. At the same time, while recent work (e.g., [26]) on using clickthrough information for improving web search ranking is promising, it captures only one aspect of the user interactions with web search engines.

We build on existing research to develop robust user behavior interpretation techniques for the real web search setting. Instead of treating each user as a reliable "expert", we aggregate information from multiple, unreliable, user search session traces, as we describe in the next two sections.

3. INCORPORATING IMPLICIT FEEDBACK

We consider two complementary approaches to ranking with implicit feedback: (1) treating implicit feedback as independent evidence for ranking results, and (2) integrating implicit feedback features directly into the ranking algorithm. We describe the two general ranking approaches next. The specific implicit feedback features are described in Section 4, and the algorithms for interpreting and incorporating implicit feedback are described in Section 5.

3.1 Implicit Feedback as Independent Evidence

The general approach is to re-rank the results obtained by a web search engine according to observed clickthrough and other user interactions for the query in previous search sessions. Each result is assigned a score according to expected relevance/user satisfaction based on previous interactions, resulting in some preference ordering based on user interactions alone.

While there has been significant work on merging multiple rankings, we adapt a simple and robust approach of ignoring the original rankers' scores, and instead simply merge the rank orders. The main reason for ignoring the original scores is that since the feature spaces and learning algorithms are different, the scores are not directly comparable, and re-normalization tends to remove the benefit of incorporating classifier scores.

We experimented with a variety of merging functions on the development set of queries (and using a set of interactions from a different time period from final evaluation sets). We found that a simple rank merging heuristic combination works well, and is robust to variations in score values from original rankers. For a given query *q*, the implicit score IS_d is computed for each result *d* from available user interaction features, resulting in the implicit rank I_d for each result. We compute a merged score $S_M(d)$ for *d* by combining the ranks obtained from implicit feedback, I_d with the original rank of *d*, O_d:

$$S_M(d, I_d, O_d, w_I) = \begin{cases} w_I \dfrac{1}{I_d + 1} + \dfrac{1}{O_d + 1} & \textit{if implicit feedback exists for } d \\ \dfrac{1}{O_d + 1} & \textit{otherwise} \end{cases}$$

where the weight w_I is a heuristically tuned scaling factor representing the relative "importance" of the implicit feedback. The query results are ordered in by decreasing values of S_M to produce the final ranking. One special case of this model arises when setting w_I to a very large value, effectively forcing clicked results to be ranked higher than un-clicked results – an intuitive and effective heuristic that we will use as a baseline. Applying more sophisticated classifier and ranker combination algorithms may result in additional improvements, and is a promising direction for future work.

The approach above assumes that there are no interactions between the underlying features producing the original web search ranking and the implicit feedback features. We now relax this assumption by integrating implicit feedback features directly into the ranking process.

3.2 Ranking with Implicit Feedback Features

Modern web search engines rank results based on a large number of features, including content-based features (i.e., how closely a query matches the text or title or anchor text of the document), and query-independent page quality features (e.g., PageRank of the document or the domain). In most cases, automatic (or semi-automatic) methods are developed for tuning the specific ranking function that combines these feature values.

Hence, a natural approach is to incorporate implicit feedback features directly as features for the ranking algorithm. During training or tuning, the ranker can be tuned as before but with additional features. At runtime, the search engine would fetch the implicit feedback features associated with each query-result URL pair. This model requires a ranking algorithm to be robust to missing values: more than 50% of queries to web search engines are unique, with no previous implicit feedback available. We now describe such a ranker that we used to learn over the combined feature sets including implicit feedback.

3.3 Learning to Rank Web Search Results

A key aspect of our approach is exploiting recent advances in machine learning, namely trainable ranking algorithms for web search and information retrieval (e.g., [5, 11] and classical results reviewed in [3]). In our setting, explicit human relevance judgments (labels) are available for a set of web search queries and results. Hence, an attractive choice to use is a supervised machine learning technique to learn a ranking function that best predicts relevance judgments.

RankNet is one such algorithm. It is a neural net tuning algorithm that optimizes feature weights to best match explicitly provided pairwise user preferences. While the specific training algorithms used by RankNet are beyond the scope of this paper, it is described in detail in [5] and includes extensive evaluation and comparison with other ranking methods. An attractive feature of RankNet is both train- and run-time efficiency – runtime ranking can be quickly computed and can scale to the web, and training can be done over thousands of queries and associated judged results.

We use a 2-layer implementation of RankNet in order to model non-linear relationships between features. Furthermore, RankNet can learn with many (differentiable) cost functions, and hence can automatically learn a ranking function from human-provided labels, an attractive alternative to heuristic feature combination techniques. Hence, we will also use RankNet as a generic ranker to explore the contribution of implicit feedback for different ranking alternatives.

4. IMPLICIT USER FEEDBACK MODEL

Our goal is to accurately interpret *noisy* user feedback obtained as by tracing user interactions with the search engine. Interpreting implicit feedback in real web search setting is not an easy task. We characterize this problem in detail in [1], where we motivate and evaluate a wide variety of models of implicit user activities. The general approach is to represent user actions for each search result as a vector of features, and then train a ranker on these features to discover feature values indicative of relevant (and non-relevant) search results. We first briefly summarize our features and model, and the learning approach (Section 4.2) in order to provide sufficient information to replicate our ranking methods and the subsequent experiments.

4.1 Representing User Actions as Features

We model observed web search behaviors as a combination of a ``background'' component (i.e., query- and relevance-independent noise in user behavior, including positional biases with result interactions), and a ``relevance'' component (i.e., query-specific behavior indicative of relevance of a result to a query). We design our features to take advantage of aggregated user behavior. The feature set is comprised of *directly observed* features (computed directly from observations for each query), as well as query-specific *derived* features, computed as the deviation from the overall query-independent distribution of values for the corresponding directly observed feature values.

The features used to represent user interactions with web search results are summarized in Table 4.1. This information was obtained via opt-in client-side instrumentation from users of a major web search engine.

We include the traditional implicit feedback features such as clickthrough counts for the results, as well as our novel derived features such as the deviation of the observed clickthrough number for a given query-URL pair from the expected number of clicks on a result in the given position. We also model the browsing behavior *after* a result was clicked – e.g., the average page dwell time for a given query-URL pair, as well as its deviation from the expected (average) dwell time. Furthermore, the feature set was designed to provide essential information about the user experience to make feedback interpretation robust. For example, web search users can often determine whether a result is relevant by looking at the result title, URL, and summary – in many cases, looking at the original document is not necessary. To model this aspect of user experience we include features such as overlap in words in title and words in query (TitleOverlap) and the fraction of words shared by the query and the result summary.

Clickthrough features	
Position	Position of the URL in Current ranking
ClickFrequency	Number of clicks for this query, URL pair
ClickProbability	Probability of a click for this query and URL
ClickDeviation	Deviation from expected click probability
IsNextClicked	1 if clicked on next position, 0 otherwise
IsPreviousClicked	1 if clicked on previous position, 0 otherwise
IsClickAbove	1 if there is a click above, 0 otherwise
IsClickBelow	1 if there is click below, 0 otherwise
Browsing features	
TimeOnPage	Page dwell time
CumulativeTimeOnPage	Cumulative time for all subsequent pages after search
TimeOnDomain	Cumulative dwell time for this domain
TimeOnShortUrl	Cumulative time on URL prefix, no parameters
IsFollowedLink	1 if followed link to result, 0 otherwise
IsExactUrlMatch	0 if aggressive normalization used, 1 otherwise
IsRedirected	1 if initial URL same as final URL, 0 otherwise
IsPathFromSearch	1 if only followed links after query, 0 otherwise
ClicksFromSearch	Number of hops to reach page from query
AverageDwellTime	Average time on page for this query
DwellTimeDeviation	Deviation from average dwell time on page
CumulativeDeviation	Deviation from average cumulative dwell time
DomainDeviation	Deviation from average dwell time on domain
Query-text features	
TitleOverlap	Words shared between query and title
SummaryOverlap	Words shared between query and snippet
QueryURLOverlap	Words shared between query and URL
QueryDomainOverlap	Words shared between query and URL domain
QueryLength	Number of tokens in query
QueryNextOverlap	Fraction of words shared with next query

Table 4.1: Some features used to represent post-search navigation history for a given query and search result URL.

Having described our feature set, we briefly review our general method for deriving a user behavior model.

4.2 Deriving a User Feedback Model

To learn to interpret the observed user behavior, we correlate user actions (i.e., the features in Table 4.1 representing the actions) with the explicit user judgments for a set of training queries. We find all the instances in our session logs where these queries were submitted to the search engine, and aggregate the user behavior features for all search sessions involving these queries.

Each observed query-URL pair is represented by the features in Table 4.1, with values averaged over all search sessions, and assigned one of six possible relevance labels, ranging from "Perfect" to "Bad", as assigned by explicit relevance judgments. These labeled feature vectors are used as input to the RankNet training algorithm (Section 3.3) which produces a trained user behavior model. This approach is particularly attractive as it does not require heuristics beyond feature engineering. The resulting user behavior model is used to help rank web search results – either directly or in combination with other features, as described below.

5. EXPERIMENTAL SETUP

The ultimate goal of incorporating implicit feedback into ranking is to improve the relevance of the returned web search results. Hence, we compare the ranking methods over a large set of judged queries with explicit relevance labels provided by human judges. In order for the evaluation to be realistic we obtained a random sample of queries from web search logs of a major search engine, with associated results and traces for user actions. We describe this dataset in detail next. Our metrics are described in Section 5.2 that we use to evaluate the ranking alternatives, listed in Section 5.3 in the experiments of Section 6.

5.1 Datasets

We compared our ranking methods over a random sample of 3,000 queries from the search engine query logs. The queries were drawn from the logs uniformly at random by token without replacement, resulting in a query sample representative of the overall query distribution. On average, 30 results were explicitly labeled by human judges using a six point scale ranging from “Perfect” down to “Bad”. Overall, there were over 83,000 results with explicit relevance judgments. In order to compute various statistics, documents with label “Good” or better will be considered “relevant”, and with lower labels to be “non-relevant”. Note that the experiments were performed over the results already highly ranked by a web search engine, which corresponds to a typical user experience which is limited to the small number of the highly ranked results for a typical web search query.

The user interactions were collected over a period of 8 weeks using voluntary opt-in information. In total, over 1.2 million unique queries were instrumented, resulting in over 12 million individual interactions with the search engine. The data consisted of user interactions with the web search engine (e.g., clicking on a result link, going back to search results, etc.) performed after a query was submitted. These actions were aggregated across users and search sessions and converted to features in Table 4.1.

To create the training, validation, and test query sets, we created three different random splits of 1,500 training, 500 validation, and 1000 test queries. The splits were done randomly by query, so that there was no overlap in training, validation, and test queries.

5.2 Evaluation Metrics

We evaluate the ranking algorithms over a range of accepted information retrieval metrics, namely *Precision at K* (P(K)), *Normalized Discounted Cumulative Gain* (NDCG), and *Mean Average Precision* (MAP). Each metric focuses on a deferent aspect of system performance, as we describe below.

- **Precision at K**: As the most intuitive metric, P(K) reports the fraction of documents ranked in the top K results that are labeled as relevant. In our setting, we require a relevant document to be labeled “Good” or higher. The position of relevant documents within the top K is irrelevant, and hence this metric measure overall user satisfaction with the top K results.
- **NDCG at K:** NDCG is a retrieval measure devised specifically for web search evaluation [10]. For a given query *q*, the ranked results are examined from the top ranked down, and the NDCG computed as:

$$N_q = M_q \sum_{j=1}^{K} (2^{r(j)} - 1) / \log(1 + j)$$

Where M_q is a normalization constant calculated so that a perfect ordering would obtain NDCG of 1; and each *r(j)* is an integer relevance label (0=”Bad” and 5=”Perfect”) of result returned at position *j*. Note that unlabeled and “Bad” documents do not contribute to the sum, but will reduce NDCG for the query pushing down the relevant labeled documents, reducing their contributions. NDCG is well suited to web search evaluation, as it rewards relevant documents in the top ranked results more heavily than those ranked lower.

- **MAP:** Average precision for each query is defined as the mean of the precision at K values computed after each relevant document was retrieved. The final MAP value is defined as the mean of average precisions of all queries in the test set. This metric is the most commonly used single-value summary of a run over a set of queries.

5.3 Ranking Methods Compared

Recall that our goal is to quantify the effectiveness of implicit behavior for real web search. One dimension is to compare the utility of implicit feedback with other information available to a web search engine. Specifically, we compare effectiveness of implicit user behaviors with content-based matching, static page quality features, and combinations of all features.

- **BM25F:** As a strong web search baseline we used the BM25F scoring, which was used in one of the best performing systems in the TREC 2004 Web track [23,27]. BM25F and its variants have been extensively described and evaluated in IR literature, and hence serve as a strong, reproducible baseline. The BM25F variant we used for our experiments computes separate match scores for each “field” for a result document (e.g., body text, title, and anchor text), and incorporates query-independent link-based information (e.g., PageRank, ClickDistance, and URL depth). The scoring function and field-specific tuning is described in detail in [23]. Note that BM25F does not directly consider explicit or implicit feedback for tuning.
- **RN:** The ranking produced by a neural net ranker (RankNet, described in Section 3.3) that *learns* to rank web search results by incorporating BM25F and a large number of additional static and dynamic features describing each search result. This system automatically learns weights for all features (including the BM25F score for a document) based on *explicit* human labels for a large set of queries. A system incorporating an implementation of RankNet is currently in use by a major search engine and can be considered representative of the state of the art in web search.
- **BM25F-RerankCT**: The ranking produced by incorporating clickthrough statistics to reorder web search results ranked by BM25F above. Clickthrough is a particularly important special case of implicit feedback, and has been shown to correlate with result relevance. This is a special case of the ranking method in

Section 3.1, with the weight w_I set to 1000 and the ranking I_d is simply the number of clicks on the result corresponding to *d*. In effect, this ranking brings to the top all returned web search results with at least one click (and orders them in decreasing order by number of clicks). The relative ranking of the remainder of results is unchanged and they are inserted below all clicked results. This method serves as our baseline implicit feedback reranking method.

BM25F-RerankAll The ranking produced by reordering the BM25F results using *all* user behavior features (Section 4). This method learns a model of user preferences by correlating feature values with explicit relevance labels using the RankNet neural net algorithm (Section 4.2). At runtime, for a given query the implicit score I_r is computed for each result *r* with available user interaction features, and the implicit ranking is produced. The merged ranking is computed as described in Section 3.1. Based on the experiments over the development set we fix the value of w_I to 3 (the effect of the w_I parameter for this ranker turned out to be negligible).

- **BM25F+All:** Ranking derived by training the RankNet (Section 3.3) learner over the features set of the BM25F score as well as all implicit feedback features (Section 3.2). We used the 2-layer implementation of RankNet [5] trained on the queries and labels in the training and validation sets.
- **RN+All:** Ranking derived by training the 2-layer RankNet ranking algorithm (Section 3.3) over the union of all content, dynamic, and implicit feedback features (i.e., all of the features described above as well as all of the new implicit feedback features we introduced).

The ranking methods above span the range of the information used for ranking, from not using the implicit or explicit feedback at all (i.e., BM25F) to a modern web search engine using hundreds of features and tuned on explicit judgments (RN). As we will show next, incorporating user behavior into these ranking systems dramatically improves the relevance of the returned documents.

6. EXPERIMENTAL RESULTS

Implicit feedback for web search ranking can be exploited in a number of ways. We compare alternative methods of exploiting implicit feedback, both by re-ranking the top results (i.e., the BM25F-RerankCT and BM25F-RerankAll methods that reorder BM25F results), as well as by integrating the implicit features directly into the ranking process (i.e., the RN+ALL and BM25F+All methods which learn to rank results over the implicit feedback and other features). We compare our methods over strong baselines (BM25F and RN) over the NDCG, Precision at K, and MAP measures defined in Section 5.2. The results were averaged over three random splits of the overall dataset. Each split contained 1500 training, 500 validation, and 1000 test queries, all query sets disjoint. We first present the results over all 1000 test queries (i.e., including queries for which there are no implicit measures so we use the original web rankings). We then drill down to examine the effects on reranking for the attempted queries in more detail, analyzing where implicit feedback proved most beneficial.

We first experimented with different methods of re-ranking the output of the BM25F search results. Figures 6.1 and 6.2 report NDCG and Precision for BM25F, as well as for the strategies reranking results with user feedback (Section 3.1). Incorporating all user feedback (either in reranking framework or as features to the learner directly) results in significant improvements (using two-tailed t-test with p=0.01) over both the original BM25F ranking as well as over reranking with clickthrough alone. The improvement is consistent across the top 10 results and largest for the top result: NDCG at 1 for BM25F+All is 0.622 compared to 0.518 of the original results, and precision at 1 similarly increases from 0.5 to 0.63. Based on these results we will use the direct feature combination (i.e., BM25F+All) ranker for subsequent comparisons involving implicit feedback.

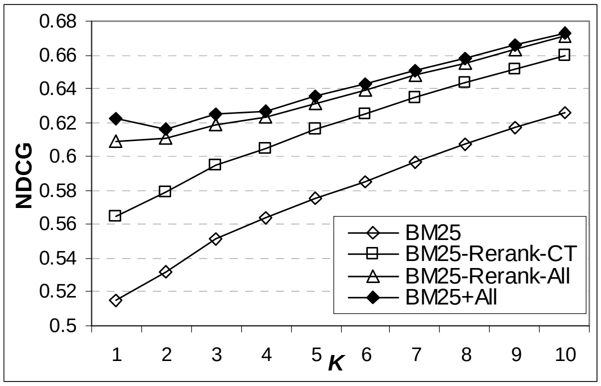

Figure 6.1: NDCG at *K* for BM25F, BM25F-RerankCT, BM25F-Rerank-All, and BM25F+All for varying *K*

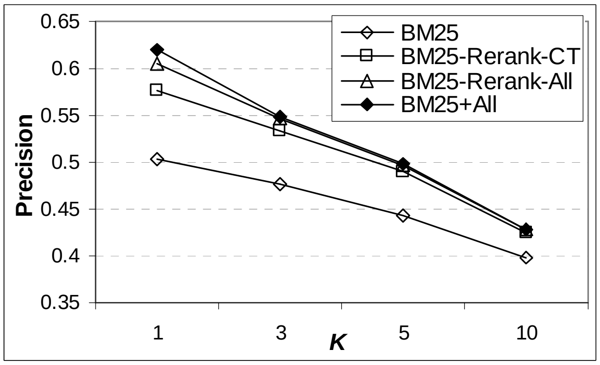

Figure 6.2: Precision at K for BM25F, BM25F-RerankCT, BM25F-Rerank-All, and BM25F+All for varying K

Interestingly, using clickthrough alone, while giving significant benefit over the original BM25F ranking, is not as effective as considering the full set of features in Table 4.1. While we analyze user behavior (and most effective component features) in a separate paper [1], it is worthwhile to give a concrete example of the kind of noise inherent in real user feedback in web search setting.

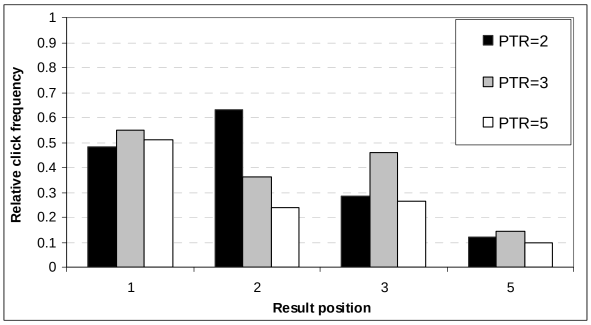

Figure 6.3: Relative clickthrough frequency for queries with varying Position of Top Relevant result (PTR).

If users considered only the relevance of a result to their query, they would click on the topmost relevant results. Unfortunately, as Joachims and others have shown, presentation also influences which results users click on quite dramatically. Users often click on results *above* the relevant one presumably because the short summaries do not provide enough information to make accurate relevance assessments and they have learned that on average top-ranked items are relevant. Figure 6.3 shows relative clickthrough frequencies for queries with known relevant items at positions other than the first position; the position of the top relevant result (PTR) ranges from 2-10 in the figure. For example, for queries with first relevant result at position 5 (PTR=5), there are more clicks on the non-relevant results in higher ranked positions than on the first relevant result at position 5. As we will see, learning over a richer behavior feature set, results in substantial accuracy improvement over clickthrough alone.

We now consider incorporating user behavior into a much richer feature set, RN (Section 5.3) used by a major web search engine. RN incorporates BM25F, link-based features, and hundreds of other features. Figure 6.4 reports NDCG at K and Figure 6.5 reports Precision at K. Interestingly, while the original RN rankings are significantly more accurate than BM25F alone, incorporating implicit feedback features (BM25F+All) results in ranking that significantly outperforms the original RN rankings. In other words, implicit feedback incorporates sufficient information to *replace* the hundreds of other features available to the RankNet learner trained on the RN feature set.

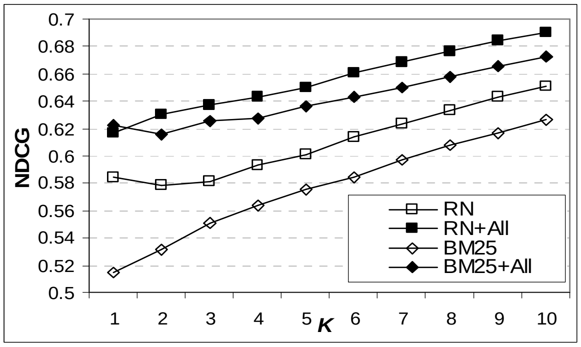

Figure 6.4: NDCG at *K* for BM25F, BM25F+All, RN, and RN+All for varying *K*

Furthermore, enriching the RN features with implicit feedback set exhibits significant gain on all measures, allowing RN+All to outperform all other methods. This demonstrates the complementary nature of implicit feedback with other features available to a state of the art web search engine.

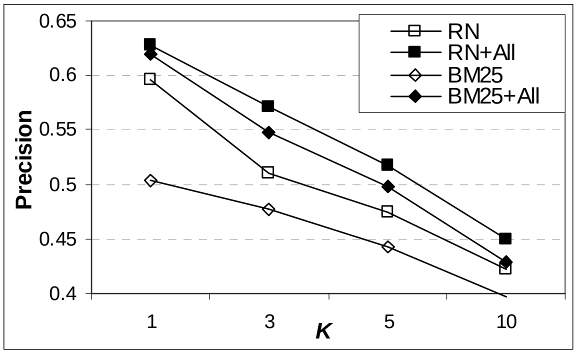

Figure 6.5: Precision at *K* for BM25F, BM25F+All, RN, and RN+All for varying *K*

We summarize the performance of the different ranking methods in Table 6.1. We report the Mean Average Precision (MAP) score for each system. While not intuitive to interpret, MAP allows quantitative comparison on a single metric. The gains marked with * are significant at p=0.01 level using two tailed t-test.

	MAP	Gain	P(1)	Gain
BM25F	0.184	-	0.503	-
BM25F-Rerank-CT	0.215	0.031*	0.577	0.073*
BM25F-RerankImplicit	0.218	0.003	0.605	0.028*
BM25F+Implicit	**0.222**	0.004	0.620	0.015*
RN	0.215	-	0.597	-
RN+All	**0.248**	0.033*	0.629	0.032*

Table 6.1: Mean Average Precision (MAP) for all strategies.

So far we reported results averaged across *all* queries in the test set. Unfortunately, less than half had sufficient interactions to attempt reranking. Out of the 1000 queries in test, between 46% and 49%, depending on the train-test split, had sufficient interaction information to make predictions (i.e., there was at least 1 search session in which at least 1 result URL was clicked on by the user). This is not surprising: web search is heavy-tailed, and there are many unique queries. We now consider the performance on the queries for which user interactions were available. Figure 6.6 reports NDCG for the subset of the test queries with the implicit feedback features. The gains at top 1 are dramatic. The NDCG at 1 of BM25F+All increases from 0.6 to 0.75 (a 31% relative gain), achieving performance comparable to RN+All operating over a much richer feature set.

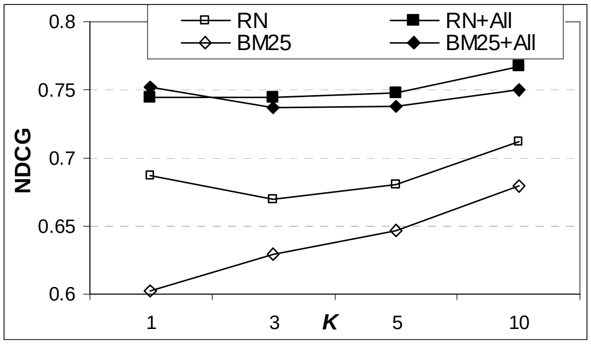

Figure 6.6: NDCG at K for BM25F, BM25F+All, RN, and RN+All on test queries with user interactions

Similarly, gains on precision at top 1 are substantial (Figure 6.7), and are likely to be apparent to web search users. When implicit feedback is available, the BM25F+All system returns relevant document at top 1 almost 70% of the time, compared 53% of the time when implicit feedback is not considered by the original BM25F system.

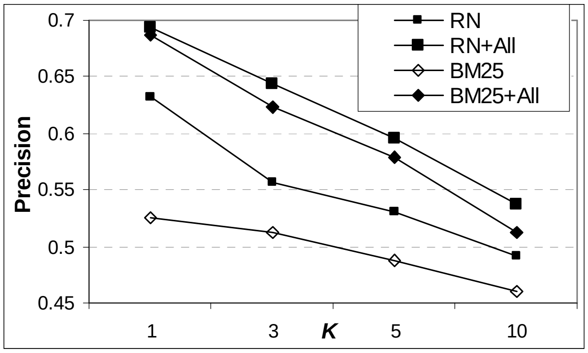

Figure 6.7: Precision at K NDCG at K for BM25F, BM25F+All, RN, and RN+All on test queries with user interactions

We summarize the results on the MAP measure for attempted queries in Table 6.2. MAP improvements are both substantial and significant, with improvements over the BM25F ranker most pronounced.

Method	MAP	Gain	P(1)	Gain
RN	0.269		0.632	
RN+All	**0.321**	**0.051 (19%)**	**0.693**	**0.061(10%)**
BM25F	0.236		0.525	
BM25F+All	**0.292**	**0.056 (24%)**	**0.687**	**0.162 (31%)**

Table 6.2: Mean Average Precision (MAP) on attempted queries for best performing methods

We now analyze the cases where implicit feedback was shown most helpful. Figure 6.8 reports the MAP improvements over the "baseline" BM25F run for each query with MAP under 0.6. Note that most of the improvement is for poorly performing queries (i.e., MAP < 0.1). Interestingly, incorporating user behavior information degrades accuracy for queries with high original MAP score. One possible explanation is that these "easy" queries tend to be navigational (i.e., having a single, highly-ranked most appropriate answer), and user interactions with lower-ranked results may indicate divergent information needs that are better served by the less popular results (with correspondingly poor overall relevance ratings).

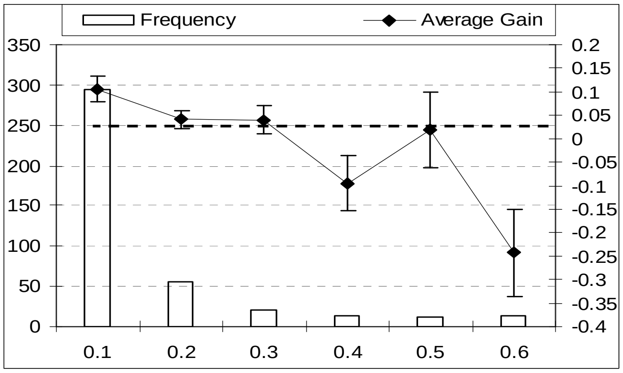

Figure 6.8: Gain of BM25F+All over original BM25F ranking

To summarize our experimental results, incorporating implicit feedback in real web search setting resulted in significant improvements over the original rankings, using both BM25F and RN baselines. Our rich set of implicit features, such as time on page and deviations from the average behavior, provides advantages over using clickthrough alone as an indicator of interest. Furthermore, incorporating implicit feedback features directly into the learned ranking function is more effective than using implicit feedback for reranking. The improvements observed over large test sets of queries (1,000 total, between 466 and 495 with implicit feedback available) are both substantial and statistically significant.

7. CONCLUSIONS AND FUTURE WORK

In this paper we explored the utility of incorporating noisy implicit feedback obtained in a real web search setting to improve web search ranking. We performed a large-scale evaluation over 3,000 queries and more than 12 million user interactions with a major search engine, establishing the utility of incorporating "noisy" implicit feedback to improve web search relevance.

We compared two alternatives of incorporating implicit feedback into the search process, namely reranking with implicit feedback and incorporating implicit feedback features directly into the trained ranking function. Our experiments showed significant improvement over methods that do not consider implicit feedback. The gains are particularly dramatic for the top *K*=1 result in the final ranking, with precision improvements as high as 31%, and the gains are substantial for all values of *K*. Our experiments showed that implicit user feedback can further improve web search performance, when incorporated directly with popular content- and link-based features.

Interestingly, implicit feedback is particularly valuable for queries with poor original ranking of results (e.g., MAP lower than 0.1). One promising direction for future work is to apply recent research on automatically predicting query difficulty, and only attempt to incorporate implicit feedback for the "difficult" queries. As another research direction we are exploring methods for extending our predictions to the previously unseen queries (e.g., query clustering), which should further improve the web search experience of users.

ACKNOWLEDGMENTS
We thank Chris Burges and Matt Richardson for an implementation of RankNet for our experiments. We also thank Robert Ragno for his valuable suggestions and many discussions.

8. REFERENCES

[1] E. Agichtein, E. Brill, S. Dumais, and R.Ragno, Learning User Interaction Models for Predicting Web Search Result Preferences. In *Proceedings of the ACM Conference on Research and Development on Information Retrieval (SIGIR)*, 2006

[2] J. Allan, HARD Track Overview in TREC 2003, *High Accuracy Retrieval from Documents*, 2003

[3] R. Baeza-Yates and B. Ribeiro-Neto, Modern Information Retrieval, Addison-Wesley, 1999.

[4] S. Brin and L. Page, The Anatomy of a Large-scale Hypertextual Web Search Engine, in *Proceedings of WWW*, 1997

[5] C.J.C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, G. Hullender, Learning to Rank using Gradient Descent, in *Proceedings of the International Conference on Machine Learning*, 2005

[6] D.M. Chickering, The WinMine Toolkit, *Microsoft Technical Report MSR-TR-2002-103*, 2002

[7] M. Claypool, D. Brown, P. Lee and M. Waseda. Inferring user interest. *IEEE Internet Computing*. 2001

[8] S. Fox, K. Karnawat, M. Mydland, S. T. Dumais and T. White. Evaluating implicit measures to improve the search experience. In *ACM Transactions on Information Systems*, 2005

[9] J. Goecks and J. Shavlick. Learning users' interests by unobtrusively observing their normal behavior. In *Proceedings of the IJCAI Workshop on Machine Learning for Information Filtering*. 1999.

[10] K Jarvelin and J. Kekalainen. IR evaluation methods for retrieving highly relevant documents. In *Proceedings of the ACM Conference on Research and Development on Information Retrieval (SIGIR)*, 2000

[11] T. Joachims, Optimizing Search Engines Using Clickthrough Data. In *Proceedings of the ACM Conference on Knowledge Discovery and Datamining (SIGKDD*), 2002

[12] T. Joachims, L. Granka, B. Pang, H. Hembrooke, and G. Gay, Accurately Interpreting Clickthrough Data as Implicit Feedback, *Proceedings of the ACM Conference on Research and Development on Information Retrieval (SIGIR)*, 2005

[13] T. Joachims, Making Large-Scale SVM Learning Practical. Advances in Kernel Methods, in *Support Vector Learning*, MIT Press, 1999

[14] D. Kelly and J. Teevan, Implicit feedback for inferring user preference: A bibliography. In *SIGIR Forum, 2003*

[15] J. Konstan, B. Miller, D. Maltz, J. Herlocker, L. Gordon, and J. Riedl. GroupLens: Applying collaborative filtering to usenet news. In *Communications of ACM*, 1997.

[16] M. Morita, and Y. Shinoda, Information filtering based on user behavior analysis and best match text retrieval. *Proceedings of the ACM Conference on Research and Development on Information Retrieval (SIGIR)*, 1994

[17] D. Oard and J. Kim. Implicit feedback for recommender systems. In *Proceedings of the AAAI Workshop on Recommender Systems*. 1998

[18] D. Oard and J. Kim. Modeling information content using observable behavior. In *Proceedings of the 64th Annual Meeting of the American Society for Information Science and Technology*. 2001

[19] N. Pharo, N. and K. Järvelin. The SST method: a tool for analyzing web information search processes. In *Information Processing & Management*, 2004

[20] P. Pirolli, The Use of Proximal Information Scent to Forage for Distal Content on the World Wide Web. In *Working with Technology in Mind: Brunswikian. Resources for Cognitive Science and Engineering*, Oxford University Press, 2004

[21] F. Radlinski and T. Joachims, Query Chains: Learning to Rank from Implicit Feedback. In *Proceedings of the ACM Conference on Knowledge Discovery and Data Mining (SIGKDD)*, 2005.

[22] F. Radlinski and T. Joachims, Evaluating the Robustness of Learning from Implicit Feedback, in *Proceedings of the ICML Workshop on Learning in Web Search*, 2005

[23] S. E. Robertson, H. Zaragoza, and M. Taylor, Simple BM25 extension to multiple weighted fields, in *Proceedings of the Conference on Information and Knowledge Management (CIKM)*, 2004

[24] G. Salton & M. McGill. Introduction to modern information retrieval. McGraw-Hill, 1983

[25] E.M. Voorhees, D. Harman, Overview of TREC, 2001

[26] G.R. Xue, H.J. Zeng, Z. Chen, Y. Yu, W.Y. Ma, W.S. Xi, and W.G. Fan, Optimizing web search using web click-through data, in *Proceedings of the Conference on Information and Knowledge Management (CIKM)*, 2004

[27] H. Zaragoza, N. Craswell, M. Taylor, S. Saria, and S. Robertson. Microsoft Cambridge at TREC 13: Web and Hard Tracks. In *Proceedings of TREC 2004*