Fighting Search Engine Amnesia: Reranking Repeated Results

Milad Shokouhi
Microsoft
milads@microsoft.com

Ryen W. White
Microsoft Research
ryenw@microsoft.com

Paul Bennett
Microsoft Research
pauben@microsoft.com

Filip Radlinski
Microsoft
filiprad@microsoft.com

ABSTRACT

Web search engines frequently show the same documents repeatedly for *different* queries within the same search session, in essence forgetting when the same documents were already shown to users. Depending on previous user interaction with the repeated results, and the details of the session, we show that sometimes the repeated results should be promoted, while some other times they should be demoted.

Analysing search logs from two different commercial search engines, we find that results are repeated in about 40% of multi-query search sessions, and that users engage differently with repeats than with results shown for the first time. We demonstrate how statistics about result repetition within search sessions can be incorporated into ranking for personalizing search results. Our results on query logs of two large-scale commercial search engines suggest that we successfully promote documents that are more likely to be clicked by the user in the future while maintaining performance over standard measures of non-personalized relevance.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Information Search and Retrieval—*Search Process, selection process*

General Terms

Algorithms, Measurement, Experimentation

Keywords

Repeated Results, User History, User Sessions, Click Prediction, User Modelling, Re-finding Queries

1. INTRODUCTION

When interacting with Web search engines, people frequently encounter the same results for different queries, both within a single session and across multiple sessions [33]. There are times when this repetition may be intentional, as is the case with re-finding, where users purposely seek the same Web document [30, 32]. However, as we will show, repeated results are shown to users much more often than can be explained by re-finding alone. Rather, results are often shown multiple times as users try to find the solutions to satisfy information needs – and most Web search engines exhibit a form of amnesia, without taking longer interactions with users into account. Previous research has largely treated sequences of queries independently [2, 20], or as weakly interacting by building models of user tasks or interests [37], rather than as a single continuing conversation between the user and search engine. Making this problem even more interesting, repetition of results can intuitively be interpreted in a few distinct ways.

Consider a document shown twice to a user for two related queries. The first time it was returned, the document must have been (1) not noticed, (2) noticed and considered but not clicked, or (3) clicked. If we can detect the first case, we may want to promote the document to help the user notice it as recurring and potentially relevant. If we can detect the second case, we may want to demote it. The third case is most interesting, as the user may be attempting to re-find the document (in which case it should be aggressively promoted), or the user may have decided that it is non-relevant (in which case it should be aggressively demoted). This leads us to the two key challenges we address in this paper: First, how do we know when a later query is similar enough to an earlier query to warrant inclusion considering previous displays of repeated documents? Second, how do we differentiate between re-finding and non-relevance?

Before describing our approach, it is important to verify that Web results are often shown repeatedly, and that behavior on them is different. Our analysis shows that repetition is both frequent, and has a substantial effect on user behavior. About 40% of multi-query sessions include at least one result being shown in the top-10 results more than once. When a result is skipped once, depending on position, it is 20% to 50% less likely to be clicked later – compared to the expected clickthrough rate (CTR) at the same position. But unnoticed results are about 10% more likely to be clicked when shown again later. Conversely, results previously clicked *once* are 10% to 50% less likely to be clicked when shown again while results previously clicked *three times* are up to 30% more likely to be clicked when shown again.

We conjecture that a method that can leverage the skip and click behavior of users to automatically promote and

SIGIR'13, July 28–August 1, 2013, Dublin, Ireland.

demote results could lead to better ranking. We therefore propose R-cube, a *context-aware* ranker enhanced by features generated from a user's interaction with repeatedly-displayed search results. We evaluate R-cube *offline* using log data from two large commercial search engines (one proprietary and one publicly available), and *online* via A-B testing with users of a commercial search engine. The results show significant gains from R-cube over competitive baselines.

After presenting related work in Section 2, and based on an analysis of logs in Section 3, we build models that leverage features of these repeated results and other features such as query similarity to predict click preferences between result pairs. Section 4 presents our re-ranking approach, and describes our evaluation data and metrics. We follow that by discussing our experimental results in Section 5, and providing concluding remarks in Section 6.

2. RELATED WORK

There is substantial prior research that relates to this work in a number of areas. We review each in turn.

Repeated Queries and Results. Searchers repeat queries and revisit the same results often. Previous research has shown that well over half of all of the Web pages a person visits are pages that they have visited previously (with some estimates of re-visitation volume reaching 80%) [8, 29], and a third of the queries issued to search engines involve a user re-finding a previously-found result [30]. Teevan *et al.* [30] explored how queries used to re-find changed and how well future clicks could be predicted following repeat queries. In a similar study and perhaps the most related to our work, Teevan *et al.* [32] proposed a model for *personal navigation*. The authors monitored the user's long-term history and showed that when the user repeatedly submits the same query and clicks on the same single result over time, the target URL can be regarded as a personal navigation destination. That is, if the user issues the same query again, it is likely that the same result gets clicked. Similarly, Dou *et al.* [12] consider re-ranking for identically repeated queries.

In our analysis, the click patterns on exactly repeated queries and results are a special case. In contrast to personal navigation [32] that only focuses on singleton clicked results over a long user history, we cover several classes of user feedback (click, skip, missed) based on the short session history. We model user behaviour on any repeated result, regardless of whether it was clicked or not, or even shown for the same query before.

Other studies have looked at temporal properties of an individual's repeated searches and clicks and aspects of repeat queries related to time [25], and how people re-find previously viewed Web pages across multiple sessions and within an individual session [30, 33].

Interpreting Clicks. In general, search engine result page clicks provide informative signals of when users are attracted to a search result. Joachims *et al.* [21] analysed users' decision processes using gaze tracking and compared implicit feedback from search-result clicks against manual relevance judgements. They found that clicks are informative but biased: Clicks as an absolute signal are biased by rank, but relative preferences derived from clicks are fairly accurate. Agichtein *et al.* [2] also used search and browsing data from a Web search engine to predict search result preferences but generalized the approach to model user behaviour beyond clicks, resulting in more accurate preference predictions.

Searcher models (e.g., [7, 10, 34]) track the user's state as they examine search results and use the observable events (e.g., clickthrough) to infer search result attractiveness and document relevance. The examination hypothesis [13] states that the likelihood that the user will click on a search result is influenced only by (i) whether the user examined the search result snippet and (ii) its attractiveness. Since users are biased towards clicking search results that are higher ranked, the examination hypothesis is used to isolate a search result's attractiveness from its position. The cascade hypothesis [10] assumes that a user always examines search results sequentially from top-to-bottom, and is used to determine whether a user examined the result. We use this approach here. Under this assumption, a user decides whether to click a result before examining the next result, preventing scenarios where the user returns to a higher-ranked search result after passing it by. Therefore, if users do not examine a particular search result, they will not examine any search results below it. Extensions of the cascade hypothesis allow for query sessions to comprise multiple clicks [17] or represent the probability that a user abandons a query session without clicking [7, 16]. Other searcher models avoid the cascade hypothesis entirely, allowing the user to examine results non-sequentially via modelling user browsing behaviour [14, 34], or modelling tasks, via the sequence of queries and clicks in a session [40].

Learning from Clicks. In addition to modeling search behavior, clicks have also been used to improve the quality of search engine ranking. Joachims [20] presents a method for automatically optimizing the retrieval quality of search engines using implicit feedback gathered from clickthrough data. He shows that his method is both effective and scalable, outperforming a state-of-the-art search engine given only limited training data. Radlinski and Joachims [23] study connected query chains to generate preference judgements from search engine logs. These judgements are used to learn search result rankings that outperform a static ranking function. Agichtein *et al.* [1] extend this work beyond search engine result pages (SERPs). Incorporating later user behaviour data, they show that it can be used to improve search engine performance.

User Interest Modelling. Knowledge of a particular user's interests and search context has been used to improve search. Short-term interests, expressed by actions such as recent queries or clicks, can improve retrieval performance [11, 27, 39]. They have also been used by White *et al.* [36, 37] to construct models of searchers' recent interests, and predict future interests. Teevan *et al.* [31] constructed user profiles of long-term interests from desktop documents, showing that this information could be used to re-rank results and improve relevance for individuals. Matthijs and Radlinski [22] constructed user profiles using users' browsing history, and evaluated their approach using an interleaving methodology. Bennett *et al.* [4] compared the effectiveness of features based on short-term and long-term history of the user for personalization, and found the latter particularly effective at the start of search sessions.

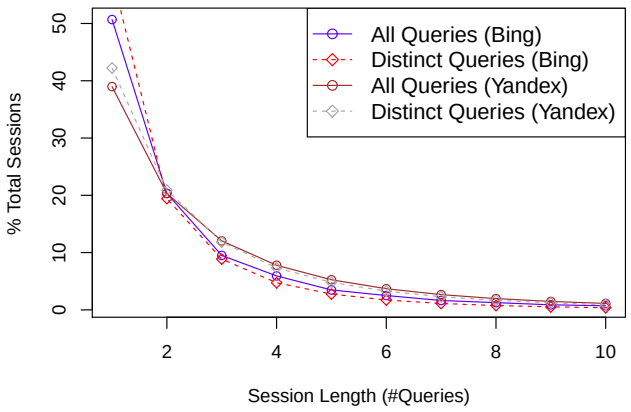

Figure 1: The distribution of the number of queries in sessions (session length). For distinct columns, repeated queries in each session are ignored.

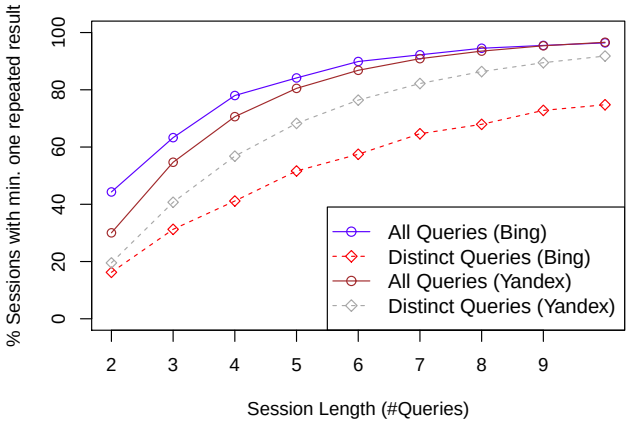

Figure 2: Fraction of sessions that include a repeated result as a function of length of session.

Our Contributions. Our research extends previous work in a number of ways. First, we focus specifically on repeatedly-displayed search results, arguing for treating the session as a continuous interaction. Second, we characterize the role that repeat results play within search sessions and the potential benefit from targeting repetition. This serves to motivate our work. Third, we focus on interpreting how users engage with results to determine whether to promote or demote repeated results. Previous work has only looked at repetition in identical repeated queries [12, 32]. Our proposal is more general, targeting the promotion or demotion of any repeat result from any position for any query, related to the current query or not. Finally, we demonstrate significant improvements in result re-ranking performance using our methods over competitive baselines on search logs from two large commercial search engines, including one dataset that is publicly available.

3. MOTIVATION AND HYPOTHESES

As a first step, we must verify that repetition happens frequently in search sessions, and that repeated results are treated differently by users. We perform this analysis on sample logs of two large commercial search engines. The first one consists of three weeks of log data from Bing sampled in November 2012. The logs contain a search session identifier, the query and the top 10 results returned by the engine. We take search sessions to comprise a sequence of queries and clicks, terminated by a 30-minute inactivity time-out, as in previous work (e.g. [23]). To facilitate reproducibility of our experiments and results, we also evaluate on a more readily available log. This second log consists of approximately 340 million queries in 44 million sessions from Yandex. These logs were shared by Yandex for participants of the Relevance Prediction Challenge[1] held in the scope of the WSDM-WSCD workshop in 2012 [26]. The Bing log contains anonymized user IDs, and preserves the full text of submitted queries. In Yandex data however, the user IDs are not available and query texts are replaced by unique numerical identifiers. Note that Bing and Yandex datasets were collected not only from different search engines, but also from users of different markets and languages.

Sessions and Repetition. Figure 1 shows that between 40 and 60% of sessions (depending on whether we focus on distinct[2] queries, and depending on the search engine) have single queries only, and the remaining 40%-60% have two queries or more. These numbers are comparable to the 46-54% split reported elsewhere [18]. Figure 2 shows that 16-44% of sessions (depending on the search engine) with two queries have at least one repeated result. This is large when we consider that the user in such short sessions is usually shown not more than 20 results. Repetition increases to almost all sessions with ten or more queries (or up to 75% when only distinct queries in the session are considered). Clearly repetition is frequent.

Classes of Repetition. For a given query and its results (subsequently referred to as an *impression*), each result must either be displayed for the first time in the session (new), or must have been shown for a previous query. For analysis, we split the repeated class into three groups[3]:

- Results previously displayed and *clicked*.
- Results previously displayed but *skipped*. We consider a result skipped when it was not clicked by the user, but at least one lower ranked result was clicked.
- Results previously displayed but *missed*. We consider a result missed when it was not clicked by the user and there were no clicked results at lower positions.

The clickthrough rates (CTR) for the Bing dataset in Figure 3 demonstrate that users interact with the results in each of these classes differently. The x-axis represents the result position, while the y-axis shows the average normalized CTR for all results in this class.[4] The numbers in the top plot are

[1] `http://imat-relpred.yandex.ru/en/`

[2] Repeated queries in the session are discarded.

[3] Note that a particular result may belong to more than one category when it appears more than twice in the session.

[4] We rescale all clickthrough graphs, based on the overall CTR@1 value averaged across all queries and impressions.

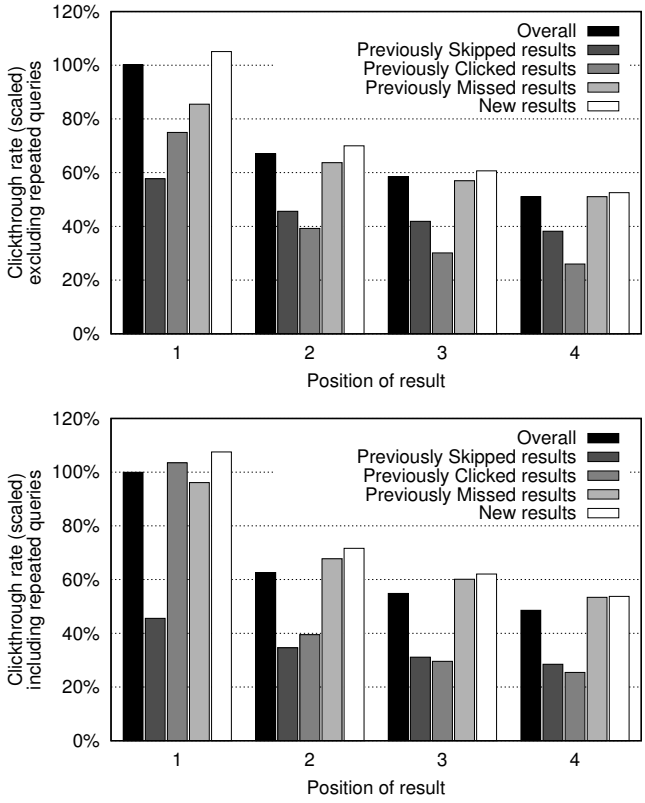

Figure 3: The clickthrough rates for different classes of results for the Bing dataset (scaled with respect to the overall CTR@1 rates across all queries and impressions). In the top plot, repeated queries in each session are excluded before calculating clickthrough rates. In the bottom plot, they are not.

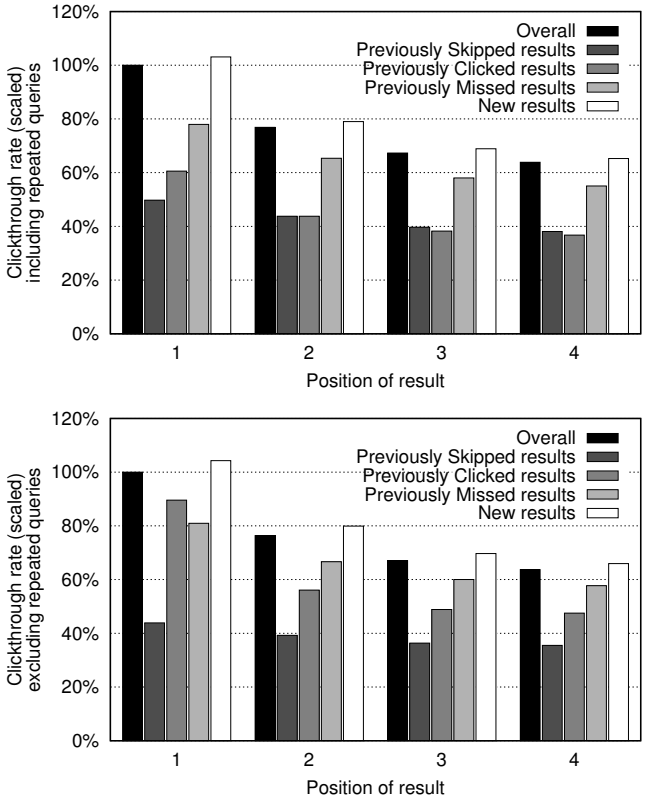

Figure 4: The clickthrough rates for different classes of results for the Yandex dataset (scaled with respect to the overall CTR@1 rates across all queries and impressions). In the top plot, repeated queries in each session are excluded before calculating clickthrough rates. In the bottom plot, they are not.

generated after excluding identical repeated queries to reduce the impact of personal navigation, while in the bottom plot all impressions are retained. The overall trends in the top plot are intuitive: repeated results have substantially lower CTR than those displayed for the first time in the session. Among the repeated classes, those that are likely to have been missed by the user for the earlier impressions are more likely to be clicked, followed by those that have been displayed but skipped. The only exception is the CTR at position 1. This is due to personal navigation [32], where users frequently re-find the same URL but with a different query. Note that although personal navigation was strictly defined for identical query-result pairs, users sometimes use different reformulations in a session to navigate to the same target. This is particularly common for popular navigational destinations (e.g. using `facebook` and `facebook login` interchangeably to navigate to `www.facebook.com`). Similar observations were reported by Teevan *et al.* [30] suggesting that in about 5% of queries, users re-find their navigational destination by issuing different queries.

The bottom plot in Figure 3 shows the impact of including repeated queries on CTR. It reveals two main differences compared to the top chart, where repeated queries were excluded. First, there is an increase in the CTR at the top rank position for previously-clicked results. This can be attributed to within-session personal navigation behaviour using identical queries. Second, there is a consistent drop in CTR across all four rank positions for results that were previously skipped. From this observation we can infer that once a user has skipped a result for a particular query, it is unlikely that they will click on it when it is returned again for the same query. Equivalent analysis on Yandex data shows very similar trends in Figure 4.

We continue by studying the repetition categories separately. First, consider results that have been skipped. Figure 5 based on the Bing (top) and Yandex (bottom) datasets shows that, the more often a result has been skipped in a session, the less likely it is to be clicked when repeated again. As before, the x-axis is the position, and the CTR curves are scaled with respect to the average CTR across all impressions. The results show that the expected CTR@1 drops by almost 60% when a result is skipped once in the session. That is, the conditional click probability of a document displayed at position one given that is skipped once before in the session, is 60% lower than the overall CTR figures at position one. We repeated the analysis for distinct queries in the session and the trends remained the same, hence are not presented here separately. Likewise, we noticed that skipped and missed results have similar CTR distributions across different positions, with the CTR values being generally higher for the latter category. In addition, the missed results have a relatively less skewed CTR distribution, and

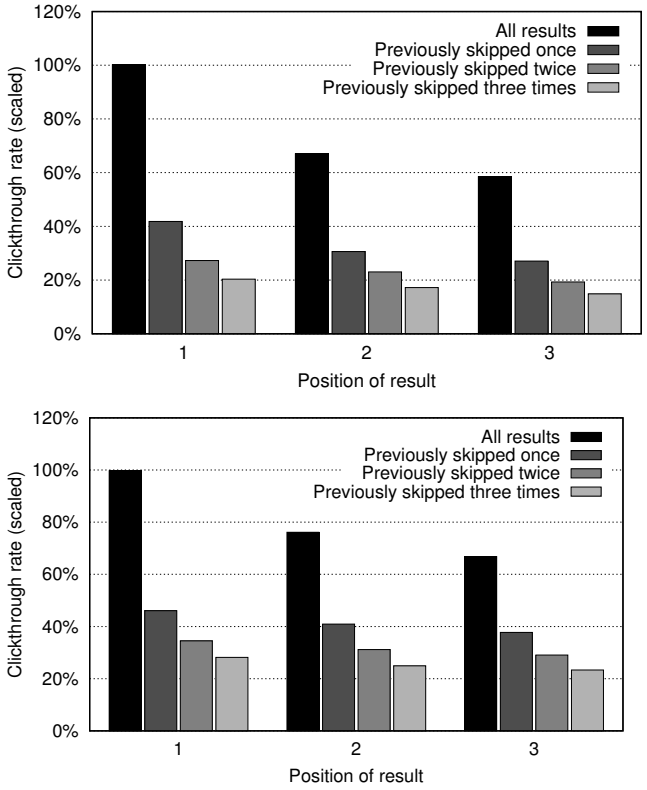

Figure 5: Clickthrough rate of repeated results as a function of the number of times they have been skipped before in the session, based on the Bing (top) and Yandex (bottom) logs.

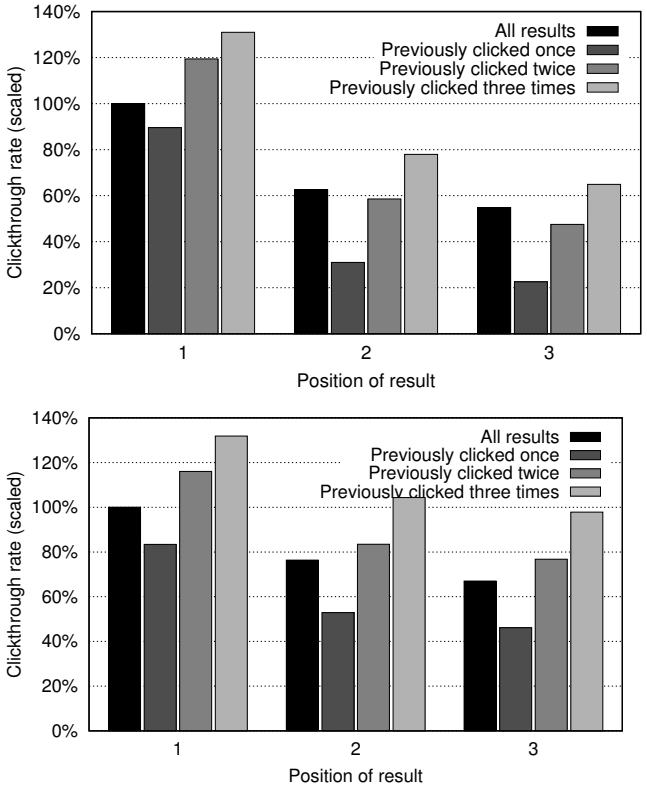

Figure 6: Clickthrough rate of repeated results as a function of the number of times they have been clicked before in the session, based on the Bing (top) and Yandex (bottom) logs.

are substantially more likely to be clicked at lower positions (the plots are omitted for brevity).

Figure 6 shows the conditional click probability of results given their position and the total number of prior clicks they have received in the session. We see that the more a result is clicked in the session, the more likely it is going to be clicked again. If a result was previously clicked just once, its expected CTR is lower than the average. However, once a result is clicked for the second time, the CTR increases substantially as it is more likely to fall in the personal navigation category. Excluding repeated queries (not shown here) led to lower CTR values for the clicked results overall but the general trends remained the same.

Beyond the scenarios covered here, there could be several other factors that affect the click probability of repeated results. For instance, in Figure 7 we can find a clear positive correlation between the amount of time spent on a result when it has been clicked once previously in the session, and the likelihood of it being clicked again. The unit of time is not specified in Yandex logs, hence the dwell time analysis is based on the Bing data only. Here, each column represents the total amount of time *dwelled* on the result. Shorter dwell times may reflect dissatisfaction with the content of the result [15] and make re-visitation less likely. We now take the insights obtained from this analysis, and develop features that can be used to improve the ranking of repeated results.

4. RE-RANKING REPEATED RESULTS

Thus far, we have demonstrated that repeated results have different click likelihoods than those that are shown for the first time in a session. We also showed that depending on the number of previous clicks, number of prior displays, and the total dwell time, the click probability of repeated results may vary significantly. In this section we describe how such features can be used to train a model for re-ranking search results in sessions. We will refer to this trained model as R-cube, as it involves Re-ranking Repeated Results.

Collecting relevance labels for evaluating *context-sensitive* rankers such as R-cube is not trivial. We need a personalized set of judgements in which the same query-result pair may be assigned with different labels depending on the search context and preferences of each user. Fox *et al.* [15] proposed using implicit measures such as satisfied clicks (SAT-Clicks) for evaluation and since then, this approach has become a common practice for evaluating personalized search systems [3, 4, 9]. We follow the same *log-based* approach for generating training data and assigning labels. First we sample a set of *impressions* from query logs. Each impression consists of a query $\mathcal{Q}$, the unique identifier of the user who issued $\mathcal{Q}$, and the set of results that were returned for $\mathcal{Q}$ on that impression. Additionally, click statistics for that impression are collected for assigning labels, and contextual features such as previous queries and clicks are used for generating ranking features. As in previous work [3, 4], in each impression, we assign positive labels to results that had SAT-Clicks, and

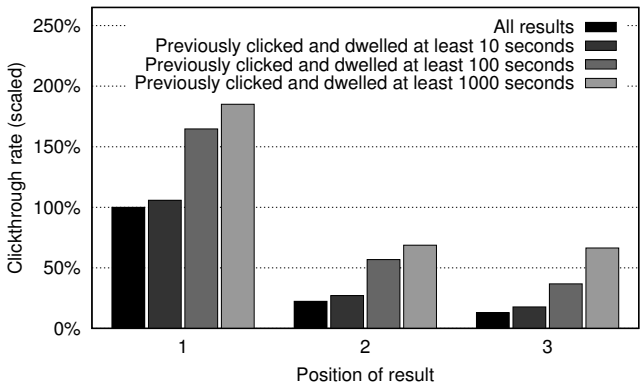

Figure 7: Clickthrough rate of repeated results as a function of the amount of dwell time spent on them after their first click in the session. Each column represents the total amount of time *dwelled* on the result in common logarithmic scale (Bing dataset).

consider the other results as non-relevant. To define a SAT-Click, we follow the definition suggested in previous work [28, 35] and consider last clicks in the session, and clicks with at least 30 seconds dwell time as SAT-Clicks.[5]

To measure the effectiveness of R-cube features for re-ranking, we sample a set of impressions from logs, generate the features for each impression, assign the labels as described above, and finally train a ranker. We chose to use LambaMART [38] for the preference learning. LambdaMART is an application of the LambdaRank approach [5], which gives a methodology for optimizing a variety of non-continuous ranking objective functions, to gradient-boosted decision trees. LambdaMART and generally gradient-boosted decision trees have been successful over a number of information retrieval problems – most notably the Yahoo! Learning to Rank challenge Track 1 [6] where LambdaMART was a key component of the winning entry.

We used our implementation's default settings for key parameters. These were, number of leaves=70, minimum documents per leaf = 2000, number of trees = 500, learning rate = 0.3 and used 10% of the training set as validation data. Note that for this study, neither the particular choice of learning algorithm nor the parameters are crucial. Our primary concern is the demonstration that it is possible to capture improved accuracy by using a learned model over the key factors we have described.

As our evaluation metrics, we compute the *mean reciprocal rank* (MRR) and *mean average precision* (MAP) of search results with respect to the log-based labels over all impressions. The Yandex dataset was released with limited set of generic (context-sensitive) relevance labels. In the following sections we also measure the impact of re-ranking with respect to those labels.

R-Cube Features. Our observations in the previous sections highlighted unique characteristics of repeated results. Those inspired us to develop several features that capture various properties of repeated results. Detailed description of our features are provided in Table 1. In general, we have four groups of features:

- (♢) Click features : These features are developed based on the number and position of previous clicks on the results (e.g. PrevClicked).
- (♡) Display features: These are similar to the click features except that they focus on the display positions regardless of whether the repeated result was previously clicked or not (e.g. PrevShown).
- (♣) Query similarity features: These features are designed to distinguish between repeated queries, personal navigation, re-finding, and query reformulation for exploring new information (e.g. RepeatQuery).
- (♠) Other features: These are features that do not belong to any of the categories above (e.g. Position).

Data. To evaluate our approach quantitatively, we use the Bing and Yandex query logs described in Section 3. For Bing analysis, we use the second week of logs (from November 8th to November 15th, 2011) for training, and the third week (between November 16th and November 23rd) for testing. The first week was used for analysing the CTR patterns that we covered in earlier sections. In total, there were $6,672,701$ impressions in the testing subset of Bing dataset from which $1,273,577$ had R-cube features (19.1%). Other impressions were either the first query of their sessions or did not have any repeated results. We report the result on this segment, and the overall impact across the entire traffic can be re-scaled accordingly.

Similarly, on the Yandex testbed, we split the dataset into two sets for training and testing. Splitting is done according to the SessionID meta-data in the logs. We sampled sessions with SessionID smaller than $3E+07$ and assigned them to the training set, and sampled from the remainder to generate our testing set. In total, there were $31,814,133$ impressions in our testing subset dataset from which $5,580,720$ had R-cube features (17.5%).

Baselines. In our experiments, we use three baselines. The first baseline (*Default*) involves no re-ranking: This is just the default order that search results were retrieved by Bing and Yandex search engines.

For our second baseline (*ClickHistory*) we train a ranker with three features: {Score, Position, and Click-history}. The former two features are common among all our experiments. The Click-history feature represents the overall long-term click counts for each result on a per query basis (without regard to which user clicked which result). Hence, for each result pair in our test data, the document that was historically clicked most by all users for the current query gets the highest Click-history value. Note that this baseline only affects queries that have been observed previously, and where the same results returned currently were previously clicked. In other cases, the default ordering is preserved. This is slightly different than simply re-ranking results according to their past click frequency. Here, the ranker may *choose* not to apply such re-ranking depending on the differences in the original Scores.

[5]Each query in the Yandex dataset is assigned with a numerical value as time-stamp. However, no extra information is disclosed about how these values were computed. We therefore relax our positive labelling criteria from SAT-Clicks to Clicks on this dataset.

Table 1: The features used in R-cube for re-ranking search results. The suit in each bracket represents the experimental group the corresponding feature belongs too. Note that *query similarity* features (♣) are not available in experiments using the Yandex data.

Feature	Description
PrevClicked (♢)	How many times the document is clicked earlier in the session.
PrevClickedMRR (♢)	Similar to PrevClicked, but with a reciprocal rank discount determined according to the clicked positions.
PrevShown (♡)	How many times the result is shown earlier in the session.
PrevShownMRR (♡)	Similar to PrevShown, but with a reciprocal rank discount determined according to the display positions.
PrevMissed (♡)	How many times the result is *missed* earlier in the session.
PrevMissedMRR (♡)	Similar to PrevMissed, but with a reciprocal rank discount determined according to the display positions.
PrevSkipped (♡)	How many times the result is *skipped* earlier in the session.
PrevSkippedMRR (♡)	Similar to PrevSkipped, but with a reciprocal rank discount determined according to the display positions.
MaxQSim (♣)	Max. n-gram similarity of current query with any of the previous queries in the session.
AvgQSim (♣)	Average n-gram similarity of current query with previous queries in the session.
PrevQSim (♣)	n-gram similarity of current query with the previous query in the session.
MaxClkQSim (♣)	Max. n-gram similarity of current query with any previous query in the session where the same URL is clicked.
AvgClkQSim (♣)	Average n-gram similarity of current query with previous queries in the session where the same URL is clicked.
PrevClkQSim (♣)	n-gram similarity of current query with the previous query in the session where the same URL is clicked.
RepeatQuery (♣)	1 if the query has appeared before in the session and 0 otherwise.
QueryNo (♠)	Number of queries submitted in the session including the current query.
Position (♠)	Display rank of the URL before re-ranking.
Score (♠)	Original score of the URL before re-ranking.
NumSessionClicks (♠)	Total number of clicks in the previous impressions of the session.
NumRepAbove (♠)	Number of repeated documents in the current impression that are ranked at or above the current position.
PrevDwell (♠)	Total amount of dwell time spent on this URL on previous clicks.

Finally, in our third baseline (*Personal Navigation*) [32], we train a re-ranking model based on the original Score, Position, and a Personal Navigation feature. The *Personal Navigation* feature is only applicable on repeated queries, and counts the number of times a particular result has been clicked for the same query previously in the session.[6]

Our last two baselines can respectively be regarded as a proxy for non-personalized re-ranking models based on aggregated click frequencies, and repetition-aware state-of-the-art models for search personalization.

5. EXPERIMENTS

We compare the effectiveness of R-cube features in personalizing search results against the three baselines. The results on the Yandex and Bing datasets are respectively summarized in Tables 2 and 3. We evaluate each model in terms of *mean average precision* (MAP) and *mean reciprocal rank* (MRR). These are computed based on the SAT-Click labels generated offline as described in Section 4.

On Yandex data (Table 2) R-cube significantly outperforms all baselines on both metrics ($p < 0.01$). Compared to the no re-ranking (Default) baseline, R-cube improves MRR by 2.1% and increases MAP by 3.2%. Other re-ranking baselines also outperform the default ranker but their gains are significantly smaller. In contrast to the ClickHistory and Personal-Navigation baselines, R-cube considers all types of repetition including *misses* and *skips* in the session. To investigate the impact of such features on the effectiveness of R-cube we train a ranker without them, using the click-based (♢) features only. The evaluation results for this ranker are shown in the last row of Table 2. While R-cube still manages to outperform all baselines by using the click features only, the gains are reduced by half, confirming that modelling other types of repetition is indeed useful for re-ranking.

[6]In the original *Personal Navigation* paper [32], re-ranking rules are hard-coded and require clicks to be *the only clicks* in impression. Given that we use a supervised model for re-ranking, we relax the only click rule to improve coverage.

The results on Bing data are presented in Table 3. Due to proprietary nature of this dataset, we only report the relative gains and losses against the default no-reranking baseline and the absolute MAP and MRR values are not disclosed. Overall, the trends are similar to those on the Yandex dataset. R-cube significantly outperforms all other baselines ($p < 0.01$) on both MAP and MRR. One interesting observation is that the ClickHistory baseline does not show any improvement over the Default ranker. This is to some extent expected since the Default ranker – which is the production ranker of a commercial search engine – is already trained to take clicks into account. We also suspect that the Yandex data has been sampled over a longer period compared to our Bing dataset which is sampled over three weeks. The details of sampling have not been disclosed by Yandex, but their dataset is considerably larger than our Bing data in terms of the number of sessions and impressions.

As in the previous testbed, we investigate the performance of R-cube when only the Click-based (♢) features are used for training and we show the results in the bottom section of the Table 3. The overall trends are similar; R-cube remains as the most effective re-ranking model but the gains are substantially reduced.

While the Yandex data does not contain the text of queries, that information is available in our Bing dataset and allows us to measure the importance of *query similarity features* (♣) separately. The last row in Table 3 includes the MRR and MAP values of a R-cube ranker that is trained without the query similarity features. The results suggest that while the query similarity features contribute positively to the gains, their importance is less significant than the *display features* (♡).

Win-Loss analysis. In addition to the overall evaluation numbers presented above, we investigate the average number of positions that the SAT-Clicked results are moved by R-cube in the ranking. Promoting SAT-Clicks is consid-

Table 2: The MRR and MAP of the baselines and our approach (R-cube) on Yandex data. The differences between the baselines and R-cube are all statistically significant according to a t-test ($p < 0.01$). The lower part shows impact Click features on R-cube effectiveness. The numbers in the brackets show the relative gain/loss against the Default (no-reranking) baseline.

Model	MRR		MAP	
Default	0.696		0.536	
Personal Nav. [32]	0.700	(▲0.4%)	0.540	(▲0.7%)
ClickHistory	0.698	(▲0.3%)	0.537	(▲0.2%)
R-cube	0.711	(▲2.1%)	0.553	(▲3.2%)
R-cube (♢ feats.)	0.704	(▲1.1%)	0.544	(▲1.5%)

Table 3: The MRR and MAP of the baselines and our approach (R-cube) on the Bing dataset. All differences between the baselines and R-cube are statistically significant according to a t-test ($p < 0.01$). The lower part shows impact of different feature groups on R-cube effectiveness. The numbers in the brackets show the relative gain/loss against the Default baseline. Due to the proprietary nature of the dataset, only the relative gains and losses are shown.

Model	MRR		MAP	
Personal Nav. [32]	-	(▲1.2%)	-	(▲3.3%)
ClickHistory	-	(0.0%)	-	(0.0%)
R-cube	-	(▲3.4%)	-	(▲7.1%)
R-cube (♢ feats.)	-	(▲1.4%)	-	(▲3.7%)
R-cube (♢, ♡ feats.)	-	(▲3.0%)	-	(▲6.3%)

ered as a win and contributes positively to MAP and MRR while demoting a SAT-Click would have the opposite effect. Figure 8 depicts the percentage of SAT-Clicks that moved up, or down, by the given number of positions. The x-axis represents the number of positions moved while the y-axis shows the percentage of SAT-Clicks moved by that many positions. The blue bars are used for specifying the promoted SAT-Clicks (good re-ranking), and the red bars are used to represent demoted (loss) cases.

One both testbeds, the great majority of changes are by one position. On the Bing dataset (top), 63% of SAT-Click moves are by one position, from which 46% are correctly promoted while 17% are incorrectly demoted – 2.7 win/loss ratio. On Yandex data (bottom) the win/loss ratio for one-position changes is 1.6 (43% promotion versus 26% demotion). On both datasets, about 90% of changes are by three positions or less and the total win/loss ratio is about 2 to 1.

Click history as re-ranking feature. R-Cube takes advantage of result repetition patterns in session history and uses them for re-ranking. In contrast, ClickHistory is based on aggregated click statistics across all users. Here, we investigate whether combining these two signals would lead to further improvements. Table 4 includes the results of this analysis on the Bing and Yandex datasets where we train a ranker with both ClickHistory and R-cube features for comparison. The original performance numbers for R-cube from Tables 2 and 3 are added as reference. As expected, the results show that combining short-term session repetition and long-term aggregate history features improves performance even further. The gains on the Yandex testbed are more noticeable. This is consistent with the trends observed earlier in Tables 2 and 3 that suggested that the ClickHistory features were relatively more effective on the Yandex dataset.

Table 4: The impact of adding *historical clicks* to R-cube on MRR. All differences compared to the default baseline are statistically significant ($P < 0.01$).

Model	MRR		MAP		
R-cube	-	(▲3.4%)	-	(▲7.1%)	Bing
+ historical clicks	-	(▲3.5%)	-	(▲7.2%)	
Default	0.696		0.536		Yandex
R-cube	0.711	(▲2.1%)	0.553	(▲3.2%)	
+ historical clicks	0.716	(▲2.9%)	0.559	(▲4.2%)	

Table 5: Evaluation results based on relevance labels using the subset of Yandex testing data for queries with relevance judgements present. Except for Personal Navigation, all differences compared to the default baseline are statistically significant ($P < 0.01$).

Model	DCG@3	DCG@10
Default	5.5	7.1
Personal Nav. [32]	5.5	7.1
ClickHistory	5.4	7.0
R-cube	5.4	7.0

Relevance-based Evaluation. In our experiments so far, we have only focused on click-based and personalized evaluation. The Yandex dataset provides binary relevance judgements for a subset of query-result-region triples, which allows us to evaluate using typical non-personalized relevance metrics. Note that clicks are inherently specific to each user and many of our improvements are due to users' personal feedback based on what they have already seen. Judges were not asked what should change in a ranking if a user previously skips or clicks particular results, hence we do not expect to see large relevance improvements here. However, we do need to validate that re-ranking does not tend to demote many relevant documents for later queries.

We filter the sessions in our Yandex testing dataset to only include queries for which have at least one result judged as relevant. We use the R-cube model learned based on click preference labels explained earlier to do full re-ranking of the results for all queries in the testing set. The DCG [19] results in Table 5 show that R-cube does not substantially degrade the ranking quality in terms of relevance metrics (although the 1.8% and 1.4% drops in DCG@3 and DCG@10 are statistically significant by the t-test). Since significance may relate to large sample sizes, we measured effect size with Cohen's d and found that all $d \leq 0.04$, suggesting almost no effect. Overall, these results are encouraging as they show that although R-cube is trained towards a click-based metric, the impact on relevance metrics is minimal.

A-B test on live traffic. We performed a second evaluation of R-cube using interleaving whereby results from different

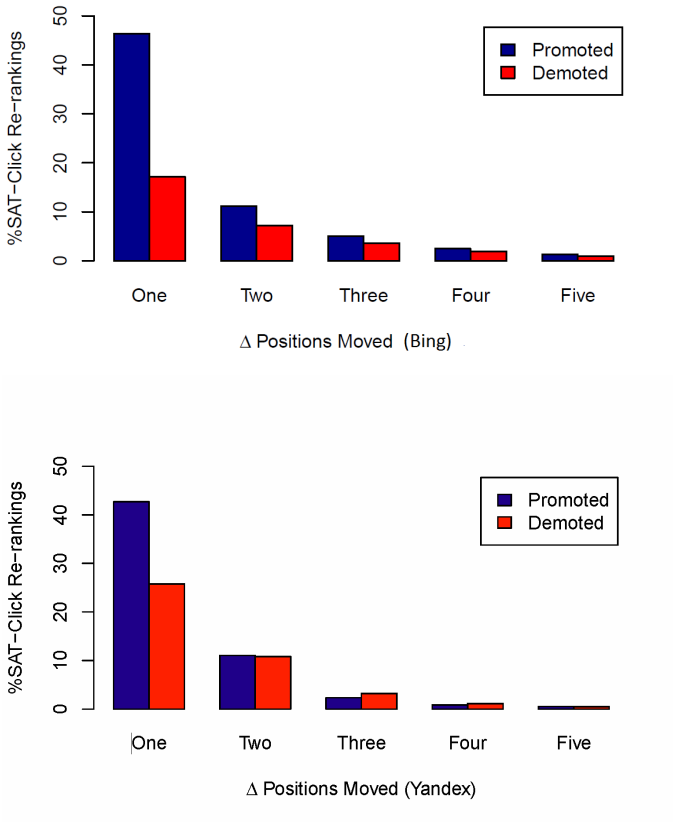

Figure 8: The distribution of re-ranked SAT-Click results with respect to their original positions on the Bing (top) and Yandex (bottom) datasets. The x-axis shows the number of positions moved and the y-axis represent the percentage of re-ranked SAT-Clicks that fall into that category. The blue bars show cases where as SAT-Click result was promoted (good re-ranking), while the red bars denote cases where the SAT-Click result was demoted (loss) with respect to its original position.

rankers are interwoven [24]. The results produced by re-ranking the top-10 Web results based on R-cubed scores were interleaved with those produced by a current commercial ranker and shown to a small, randomly selected, fraction of users of the Bing search engine. The Team Draft interleaving algorithm [24] was used, randomly allocating each result position to R-cube or the baseline. Whenever a user clicked on a result, the click was credited to the approach that had selected that result for that position. For each query, the number of clicks credited to each ranker was compared, with the ranker with more clicks receiving credit for the query (unless there was a tie due to an equal number of clicks). The random allocation of positions to rankers ensures that Team Draft interleaving produces an unbiased preference for one of the two alternatives (or a tie) [24].

We performed this evaluation for five days in June 2012, presenting the interleaved ranking for approximately 370,000 queries. We found that when the two rankers differed at or above the clicked rank, the R-cubed ranking was preferred for 53.8% of queries, and the baseline was preferred for 46.2% of queries. This preference is statistically significantly different from 50% ($p<0.001$) using a binomial sign test. The two rankers differed by the clicked position for 5.8% of queries, and the baseline and R-cubed returned a different top result for 4.5% of queries. We note that differences in rank order between R-cubed and the baseline occurred for roughly one third of queries, however Web search users predominantly click on higher-position results thus the fraction of queries affected is small. Overall, the results from live traffic indicate a statistically significant user preference for the ranker which is enhanced by R-cube features.

6. CONCLUSIONS

In this paper, we observed that the same results are often shown to users multiple times during search sessions. We showed that there are a number of effects at play, which can be leveraged to improve information retrieval performance. In particular, previously skipped results are much less likely to be clicked, and previously clicked results may or may not be re-clicked depending on other factors of the session.

We presented a number of features and a learning framework that targets the repeated results in the session for re-ranking. We conducted our experiments over query logs of two large commercial search engines (one proprietary and on publicly-available, for repeatability) and demonstrated that R-cube outperforms our experimental baselines. We showed that combining the long-term click-statistic signals in our re-ranking model increased the gains further, and also reported that the overall impact of re-ranking according to non-personalized relevance metrics is minimal. Although the R-cube model was optimized for personal, click-based judgements, demonstrating that it does not harm general relevance in the process is also important. In addition, we also performed an online interleaving analysis on the live traffic of a commercial search engine, and the results suggested a significant user preference towards the ranker enhanced by R-cube features versus the default production ranker that did not have these features.

Overall, our findings demonstrate the utility of the R-cube approach via multiple evaluation methods. It is clear from these findings that modeling different aspects of repetition (not just clicks, but also misses and skips), as well as other contextual features such as query similarity, are important in leveraging repetition and the behaviors associated with it. Future work will consider richer feature sets and long-term repetition of search results in, for example, cross-session search tasks. In addition, it would be interesting to explore repetition in other datasets such as those released with the TREC session track.[7]

7. REFERENCES

[1] E. Agichtein, E. Brill, and S. T. Dumais. Improving web search ranking by incorporating user behavior information. In *Proc. SIGIR*, pages 19–26, 2006.

[2] E. Agichtein, E. Brill, S. T. Dumais, and R. Ragno. Learning user interaction models for predicting web search result preferences. In *Proc. SIGIR*, 2006.

[7] `http://ir.cis.udel.edu/sessions`

[3] P. Bennett, F. Radlinski, R. White, and E. Yilmaz. Inferring and using location metadata to personalize web search. In *Proc. SIGIR*, pages 135–144, 2011.

[4] P. N. Bennett, R. W. White, W. Chu, S. T. Dumais, P. Bailey, F. Borisyuk, and X. Cui. Modeling the impact of short- and long-term behavior on search personalization. In *Proc. SIGIR*, pages 185–194, Portland, OR, 2012.

[5] C. Burges, R. Ragno, and Q. Le. Learning to rank with non-smooth cost functions. In *Proc. NIPS*, 2006.

[6] C. J. C. Burges, K. M. Svore, P. N. Bennett, A. Pastusiak, and Q. Wu. Learning to rank using an ensemble of lambda-gradient models. *JMLR*, 14:25–35, 2011.

[7] O. Chapelle and Y. Zhang. A dynamic bayesian network click model for web search ranking. In *Proc. WWW*, pages 1–10, 2009.

[8] A. Cockburn, S. Greenberg, S. Jones, B. Mckenzie, and M. Moyle. Improving web page revisitation: Analysis, design and evaluation. *IT & Society*, 1(3):159–183, 2003.

[9] K. Collins-Thompson, P. N. Bennett, R. W. White, S. de la Chica, and D. Sontag. Personalizing web search results by reading level. In *Proc. CIKM*, pages 403–412, Glasgow, UK, 2011.

[10] N. Craswell, O. Zoeter, M. Taylor, and B. Ramsey. An experimental comparison of click position-bias models. In *Proc. WSDM*, pages 87–94, 2008.

[11] M. Daoud, L. Tamine-Lechani, M. Boughanem, and B. Chebaro. A session based personalized search using an ontological user profile. In *Proc. SOC*, 2009.

[12] Z. Dou, R. Song, and J. R. Wen. A large-scale evaluation and analysis of personalized search strategies. In *Proc. WWW*, pages 581–590, 2007.

[13] G. E. Dupret and C. Liao. A model to estimate intrinsic document relevance from the clickthrough logs of a web search engine. In *Proc. WSDM*, pages 181–190, 2010.

[14] G. E. Dupret and B. Piwowarski. A user browsing model to predict search engine click data from past observations. In *Proc. SIGIR*, pages 331–338, 2008.

[15] S. Fox, K. Karnawat, M. Mydland, S. Dumais, and T. White. Evaluating implicit measures to improve web search. *ACM TOIS*, 23:147–168, 2005.

[16] F. Guo, C. Liu, A. Kannan, T. Minka, M. Taylor, Y.-M. Wang, and C. Faloutsos. Click chain model in web search. In *Proc. WWW*, pages 11–20, 2009.

[17] F. Guo, C. Liu, and Y. M. Wang. Efficient multiple-click models in web search. In *Proc. WSDM*, pages 124–131, 2009.

[18] B. J. Jansen, A. Spink, C. Blakely, and S. Koshman. Defining a session on web search engines: Research articles. *JASIST*, 58(6):862–871, Apr. 2007.

[19] K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM TOIS*, 20(4):422–446, 2002.

[20] T. Joachims. Optimizing search engines using clickthrough data. In *Proc. KDD*, pages 132–142, 2002.

[21] T. Joachims, L. Granka, B. Pan, H. Hembrooke, F. Radlinski, and G. Gay. Evaluating the accuracy of implicit feedback from clicks and query reformulations in web search. *ACM TOIS*, 25(2), 2007.

[22] N. Matthijs and F. Radlinski. Personalizing web search using long term browsing history. In *Proc. WSDM*, pages 25–34, 2011.

[23] F. Radlinski and T. Joachims. Query chains: Learning to rank from implicit feedback. In *Proc. KDD*, pages 239–248, 2005.

[24] F. Radlinski, M. Kurup, and T. Joachims. How does clickthrough data reflect retrieval quality? In *Proc. CIKM*, pages 43–52, 2008.

[25] M. Sanderson and S. Dumais. Examining repetition in user search behavior. In *Proc. ECIR*, 2007.

[26] P. Serdyukov, N. Craswell, and G. Dupret. Wscd 2012: workshop on web search click data 2012. In *Proc. WSDM*, pages 771–772, 2012.

[27] X. Shen, B. Tan, and C. Zhai. Context-sensitive information retrieval using implicit feedback. In *Proc. SIGIR*, pages 43–50, 2005.

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

[29] L. Tauscher and S. Greenberg. How people revisit web pages: Empirical findings and implications for the design of history systems. *International Journal of Human-Computer Studies*, 47(1):97–137, 1997.

[30] J. Teevan, E. Adar, R. Jones, and M. Potts. Information re-retrieval: Repeat queries in yahoo's logs. In *Proc. SIGIR*, pages 151–158, 2007.

[31] J. Teevan, S. T. Dumais, and E. Horvitz. Personalizing search via automated analysis of interests and activities. In *Proc. SIGIR*, pages 449–456, 2005.

[32] J. Teevan, D. Liebling, and G. R. Geetha. Understanding and predicting personal navigation. In *Proc. WSDM*, 2011.

[33] S. Tyler and J. Teevan. Large scale query log analysis of re-finding. In *Proc. WSDM*, 2010.

[34] K. Wang, N. Gloy, and X. Li. Inferring search behaviors using partially observable markov (pom) model. In *Proc. WSDM*, pages 211–220, 2010.

[35] K. Wang, T. Walker, and Z. Zheng. Pskip: estimating relevance ranking quality from web search clickthrough data. In *Proc. KDD*, pages 1355–1364, 2009.

[36] R. White, P. Bailey, and L. Chen. Predicting user interests from contextual information. In *Proc. SIGIR*, pages 363–370, 2009.

[37] R. White, P. Bennett, and S. Dumais. Predicting short-term interests using activity-based search context. In *Proc. CIKM*, pages 1009–1018, 2010.

[38] Q. Wu, C. Burges, K. Svore, and J. Gao. Adapting boosting for information retrieval measures. *Journal of Information Retrieval*, 13:254–270, 2009.

[39] B. Xiang, D. Jiang, J. Pei, X. Sun, E. Chen, and H. Li. Context-aware ranking in web search. In *Proc. SIGIR*, pages 451–458, 2010.

[40] Y. Zhang, W. Chen, D. Wang, and Q. Yang. User-click modeling for understanding and predicting search-behavior. In *Proc. KDD*, pages 1388–96, 2011.