Reducing Click and Skip Errors in Search Result Ranking

Jiepu Jiang
Center for Intelligent Information Retrieval
College of Information and Computer Sciences
University of Massachusetts Amherst
jpjiang@cs.umass.edu

James Allan
Center for Intelligent Information Retrieval
College of Information and Computer Sciences
University of Massachusetts Amherst
allan@cs.umass.edu

ABSTRACT

Search engines provide result summaries to help users quickly identify whether or not it is worthwhile to click on a result and read in detail. However, users may visit non-relevant results and/or skip relevant ones. These actions are usually harmful to the user experience, but few considered this problem in search result ranking. This paper optimizes relevance of results and user click and skip activities at the same time. Comparing two equally relevant results, our approach learns to rank the one that users are more likely to click on at a higher position. Similarly, it demotes non-relevant web pages with high click probabilities. Experimental results show this approach reduces about 10%–20% of the click and skip errors with a trade off of 2.1% decline in nDCG@10.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*retrieval models*

Keywords

Click, interactive search, web search, search result ranking

1. INTRODUCTION

The “10 blue links” paradigm has been ruling search engines for decades. It is an interaction mode where systems deliver results through a search engine result page (SERP), filled with a ranked list of summaries. These summaries are previews of result web pages, usually customized to queries, such that users can spend a small effort assessing whether it is worthwhile or not to click on a link and read in detail. Almost all current search engines adopt this mode.

The accuracy of this interaction mode greatly affects user experience. Although sometimes searchers learn relevant information solely from the SERP (e.g., “good abandonment” [5, 11, 23]), usually they need to read the result documents to satisfy their information needs. In such a case, it helps little to rank a relevant result at the top if users would not click on it. Skipping a relevant result may even have a slight adverse effect, because it requires effort to examine the summary. Similarly, clicking a non-relevant result costs the searchers' effort and may cause frustration, which seems even more harmful than skipping a non-relevant one.

One straightforward solution to the problem is to generate better search result summaries, such that searchers can make more accurate click and skip decisions. Much work has been done in this direction [29, 32], but reported click accuracy is not satisfying. For example, Yilmaz et al. [33] reported that in a commercial search engine query log, the probability of clicking on results judged as *Perfect*, *Excellent*, *Good*, *Fair*, and *Bad* are 0.94, 0.71, 0.55, 0.45, and 0.49, respectively.

Another solution is to provide answers to queries directly [6, 10]. For example, nowadays many search engines can answer “Sunday, June 21” for the query “2015 father's day”. Search engines usually display direct answers on the top area of the SERP, followed by the conventional “10-blue links”. Correct answers free users from reading results. However, it remains unclear how many queries can be satisfied by direct answers. For example, it seems difficult to generate answers for queries such as “new Macbook vs. Surface”.

In this paper, we look into the problem from a new angle—we demote results with high risks of click and skip errors in search result ranking to optimize users' click behavior. Comparing two equally relevant results, our approach learns to rank the one searchers are more likely to click on at a higher position. This reduces the chances of skipping relevant results. Similarly, we rank non-relevant results with high click probabilities to lower positions to reduce click errors. Similar to existing approaches, we also rank more relevant results on top of less relevant and non-relevant ones in order to maintain high relevance of the ranked lists.

This approach is complementary to existing solutions. For example, one can employ a new snippet generation algorithm to produce better summaries, but these summaries will still vary in risks of click and skip errors. In such case, our approach can further assist the new snippet generation algorithm by optimizing click and skip interaction. Similarly, while showing direct answers, search engines can still apply our approach to the conventional “10-blue links” as a back-up. In addition, our approach is the only known solution for situations where search providers do not have direct control over result summaries (e.g., metasearch engines).

Our approach assumes the existence of two types of ground truths—result relevance and click probability. In this paper, we rely on editorial relevance judgments for the former, and we estimate the latter from search logs. We design a metric,

WSDM'16, February 22–25, 2016, San Francisco, CA, USA.
© 2016 ACM. ISBN 978-1-4503-3716-8/16/02…$15.00
DOI: http://dx.doi.org/10.1145/2835776.2835838

click-sensitive nDCG, to measure the quality of a ranked list taking into account both relevance and click and skip errors. We train ranking models to optimize this metric. Our approaches use features such as the characteristics of summaries and past user activities in the session. Experimental results on the TREC session track dataset show our approach can reduce about 10% to 20% of the click and skip errors with a trade off of 2.1% decline in nDCG@10.

2. RELATED WORK

Our study is related to previous work in three areas. The first area is approaches for estimating unbiased click probability of results from search logs. Previous studies modeled click probability as depending on the result's rank (position-based models [4, 25]) and the quality of other results on the same SERP (cascade models [9, 14, 17, 19]). The estimated click probability is usually referred to as *attractiveness* [17]. It has been widely used as surrogates for editorial relevance judgments to train ranking models [4], or as features to rank search results [3]. We use similar approaches to estimate the click probability ground truth. But our ranking models do not use past clicks of results as features due to the lack of a secondary click probability ground truth for evaluation.

Our ranking approach differentiates click probability (attractiveness) and result relevance. This is also related to previous work that estimates both click probability and post-click satisfaction from search logs [9, 16, 19, 36]. Chapelle et al. [9] introduced the product of result click probability and post-click satisfaction probability as a measure of result relevance. They found that using this measure to train ranking models can improve the relevance of results (evaluated using editorial relevance judgments) compared with those trained using only click probability. Our ranking models are trained to optimize a similar metric, but are different in that, 1) we use the product of click probability and editorial relevance judgments for the gain of relevant results, 2) we further apply a penalty to non-relevant results, weighted by click probabilities, and 3) we evaluate by looking into both relevance of results and the risks of click and skip errors.

Another related area is studies of the influence of result summary characteristics on user click behavior. Tombros et al. [29] applied query-biased summaries to search systems, which improved users' accuracies and speed of clicking results. White et al. [32] came to a similar conclusion. Cutrell et al. [15] found that summaries of different lengths can lead to different click accuracies. Yue et al. [34] found that clicks are significantly biased towards results with more attractive titles. White et al. [31] found that when searching medical-related issues, users are significantly more likely to click on captions containing potentially alarming medical terminology such as "heart attack" and "medical emergency". A few previous works also predicted click behavior based on summary characteristics. For example: Clarke et al. [12] used features related to results' title, snippets, and URLs to predict click behavior; Hofmann et al. [20] used result summary features to correct attractiveness bias in online experiments. Our work is related to these previous studies in that we also rely on result summary characteristics as ranking features.

In addition, click behavior is also affected by context factors. For example: Cutrell et al. [15] found that users have different click behavior patterns in navigational and informational search tasks; Shokouhi et al. [27] studied how repeated results in different queries of a session were clicked by users; Jiang et al. [21] reported different browsing and clicking patterns between the first query and follow up queries in a session. This suggests that our task may benefit from the large body of work on contextual search [1]. Specifically, our ranking models use features related to past user activities in a search session, which is similar to previous work of contextual search using local session search history [18, 24, 26], although we deal with a different task. Here we only consider recent search context in a search sessions due to the lack of large scale real search logs. But the task may also benefit from longer-term search contexts, as suggested by previous work on revisiting behavior [2, 28].

3. CLICK AND SKIP ERRORS

3.1 Definition and Goal

We study the problem in the de facto standard search result page (SERP) setting. The search engine shows a ranked list of summaries linking to the result documents. Searchers first examine the summaries and then make decisions about whether or not to click on the links and read the web pages in detail. A result's summary usually consists of its title, URL, and a query-biased snippet [29, 32].

We consider two related, but different notions: *relevance* is the actual usefulness of a result document, and we assume it can only be acquired after users click on the result and read its content; *attractiveness* is how useful the result summary appears to users before they actually read its content. We quantify the attractiveness of a result by its click probability given that users examined its summary. This is similar to previous work [9, 17]. Relevance and attractiveness are correlated, but do not fully agree with each other.

We define *click error* as clicks on non-relevant results, and *skip error* as users skipping relevant results without clicking after viewing summaries. Let p be the probability of clicking on a result. If the result is not relevant, p quantifies the risk of the click error. For a relevant result, $1 - p$ measures the risk of the skip error. Click and skip errors happen when relevance and attractiveness conflict with each other.

Most existing work ranks search results by relevance. Attractiveness is usually adopted as a cheap surrogate for relevance to substitute or complement editorial relevance judgments [9]. In contrast, we do not use attractiveness as a surrogate for relevance, but as an indicator for whether or not users can effectively interact with the summary (e.g., making correct click and skip decisions). We believe that the discrepancy between relevance and attractiveness is harmful and should be avoided in search result ranking. This is a key difference between our approach and previous work.

We perform search result ranking that satisfy rules R_1–R_3. To facilitate discussion, we further introduce a few notations: D_i refers to a document; g_i is a monotonic increasing function of r_i for the gain of reading a document D_i with relevance r_i; s is the cost to examine a summary; c is the cost to click on and read a result document. Here we assume s and c are constants for all documents for simplicity.

[$\boldsymbol{R_1}$] If $p_1 \geq p_2$ and $r_1 \geq r_2 > 0$, rank D_1 over D_2.

R_1 is the case that both relevance and attractiveness prefers D_1 over D_2. In this case, it ensures optimized search efficiency (here it is intuitively defined as the ratio between gain and cost) to rank D_1 over D_2. Let g_1 and g_2 denote the gain of reading D_1 and D_2, and $g_1 \geq g_2$ because $r_1 \geq r_2$. The expected gain on D_1 and D_2 are $p_1 g_1$ and $p_1 g_2$, respectively, and $p_1 g_1 \geq p_1 g_2$. The expected cost of users on D_1 and D_2 are $s + cp_1$ and $s + cp_2$, including both the cost of examining summaries (s) and the expected cost to click on and read the documents (cp_1 and cp_2). It is easy to prove the following inequation, where the left and the right sides are search efficiency on D_1 and D_2, respectively. It suggests that ranking D_1 over D_2 can ensure better search efficiency at higher ranks. Similar to existing approach, R_1 ranks more relevant results over less relevant and non-relevant ones. In addition, R_1 reduces skip errors—D_1 has a smaller risk of skip error comparing to D_2, because $1 - p_1 \leq 1 - p_2$.

$$\frac{p_1 g_1}{s + cp_1} \geq \frac{p_2 g_1}{s + cp_2} \geq \frac{p_2 g_2}{s + cp_2}$$

[$\boldsymbol{R_2}$] If $r_1 > 0$ and $r_2 = 0$, rank D_1 over D_2.

R_2 is straightforward—ranking relevant results over non-relevant ones regardless of attractiveness. It is because users cannot benefit from D_2, but they have the chance to benefit from D_1, although the chance varies by click probability.

[$\boldsymbol{R_3}$] If $r_1 = r_2 = 0$ and $p_1 < p_2$, rank D_1 over D_2.

R_3 ensures that non-relevant results with lower click probabilities will be ranked at a higher position than those with higher click probabilities. This reduces the chances of click error and users' costs. In this case, users benefit from neither documents, but would have a higher expected cost on D_2 (because $s + cp_2 > s + cp_1$).

These rules summarize the main idea of our ranking approach. As we discussed, with mild assumptions, these rules reduce click and skip errors and maintain optimized search efficiency (gain/cost ratio). Similar to existing approaches, R_1 and R_2 also prefer more relevant results over less relevant and non-relevant ones. Yet the rules are not comprehensive enough for ranking any pairs of results.

For the rest of the cases (i.e., $r_1 > r_2 > 0$ and $p_1 < p_2$, the cases that relevance does not agree with attractiveness), we heuristically rank the result with a greater expected gain (gp) at a higher position. For example, we rank D_2 over D_1 if $g_2 p_2 > g_1 p_1$. This is reasonable to some degree, because (under our assumptions) it requires both relevance and click to acquire useful information. Whereas this sometimes conflicts with conventional ranking approaches that prefer more relevant results over less relevant ones. For example, it may prefer a marginally relevant result with a high click probability over a highly relevant one with a low click probability. In addition, it may rank results with greater risks of skip errors (lower click probability) at higher positions if they are relevant enough. Due to these reasons, we do not summarize this heuristic as a rule. We feel it requires further discussion, but we left it for future work, because in our dataset only a small fraction of results fall into this condition.

3.2 Dataset

To perform such ranking of search results, we need ground truth for result relevance and click probabilities. We use the TREC 2014 session track dataset [8], because it provides the largest publicly accessible non-anonymous query logs with relevance judgments at the time of our study. We estimate the click probability ground truth from the search logs.

The TREC 2014 session track dataset [8] collected 1,257 search sessions and 4,666 queries on 60 different search tasks. A search session includes a sequence of queries for the search task, the results and summaries shown on the SERPs, and users' search activities (e.g., clicks). Users of these sessions are paid workers from Amazon Mechanical Turk. They were shown the task descriptions and were requested to work on the tasks using an experimental search system for at least 3 minutes. Their search interaction was recorded. The session track removes the SERP and searcher interaction information for the last query of each session, because the goal was to evaluate techniques of retrieving results for the last query.

The experimental search system was built on Indri[1] and worked on a full index of the ClueWeb12 dataset[2]. The system ranks results by matching query terms and phrases to web pages' contents, titles, URLs, and anchor texts linking to the web pages. Due to the large size of the ClueWeb12 dataset (about 733 million web pages), a result caching strategy was adopted to ensure that the system could respond requests in at most 6 seconds [8]. Result snippets are generated using Indri's built-in functions. According to its source code[3], Indri constructs result snippets by selecting document fragments with the highest coverage of query terms, with a bias to the beginning of documents. This is similar to many other approaches [29]. Result relevance was judged by assessors from NIST. The pool of judgment involved all results displayed to the searchers, as well as the top-ranked results from participants' runs. Due to its large volume, only parts of the sessions were included into the judgment pool.

Carterette et al. [8] introduced more details of the dataset.

3.3 Estimating Click Probabilities

This section describes the approach of estimating results' click probabilities in the TREC dataset. For a query q and a result D, the goal is to estimate the likelihood that users, when searching for q, would click on D after viewing its summary. This probability, $P_{click}(D)$ is estimated as Equation 1, where: S_i is the collection of q's SERPs containing D; $P_{view}(D, S_i)$ is the probability that users viewed D's summary on S_i; $c(D, S_i)$ equals to 1 if users clicked on D on S_i, or 0 otherwise. The numerator counts the expected number of times D has both been viewed and clicked. The denominator is the expected number of times D has been viewed.

$$P_{click}(D) = \frac{\sum_{S_i} P_{view}(D, S_i) c(D, S_i)}{\sum_{S_i} P_{view}(D, S_i)} \qquad (1)$$

$P_{view}(D, S_i)$ comes from a logistic regression model trained to predict result fixation obtained using eye-tracking devices. The training data comes from Jiang et al.'s study [21]. We label D as viewed on S_i if there is an observed eye fixation longer than 100ms on the result's area. The logistic regression model makes use of the 10 features described in Table 1. We do not use characteristics of the result for prediction to avoid overlap with the ranking features in Section 4.

We evaluate the accuracy of the $P_{view}(D, S_i)$ model using a 10-fold cross validation setting. The dataset involves eye fixation evidence for 3,492 results, where 52% have at least one fixation (have been viewed) and 32% were clicked. The logistic regression model can achieve 0.74 average accuracy on the 10 folds ($SD = 0.024$). We use this dataset to train $P_{view}(D, S_i)$ models mainly because the search tasks performed in the user study also come from the TREC session track. We expect the $P_{view}(D, S_i)$ model to be generalizable to the TREC session track dataset.

[1] http://www.lemurproject.org/

[2] http://www.lemurproject.org/clueweb12.php/

[3] See src/SnippetBuilder.cpp (line 146-197) in Indri 5.8.

Table 1: Features for predicting $P_{view}(D, S)$.

$rank$	Rank of the result D.
$\log(rank) + 1$	Rank of the result in log base.
$P_{view}(rank)$	Probability of viewing results at the rank.
$P_{view}(rank, k)$	Probability of viewing results at the rank, given the lowest clicked rank on S is k.
$c(rank)$	Clicked on the result D?.
$c(rank{\pm}n)$	Clicked on the result at n higher or lower ranks of D? $n = 1, 2$.
$c(rank{+})$	Clicked on any results at a lower rank?

The accuracy of the click probability estimation in Equation (1) depends on both the accuracy of $P_{view}(D, S_i)$ and the size of S_i. In the TREC session track dataset, on average D has 2.76 impressions. To handle the limited observations in the dataset, we further smooth the estimation as Equation 2, where $P_{click}(R)$ is the probability to click on results with D's relevance label (R). The values of $P_{click}(R)$ comes from those reported by Yilmaz et al. [33], where they also labeled results using five levels of relevance. The probabilities to click on results with relevance labels from 4 to 0 are 0.94, 0.71, 0.55, 0.45, and 0.49, respectively. We set μ to 1.

$$P_{click}(D) = \frac{\sum_{S_i} P_{view}(D, S_i)c(D, S_i) + \mu P_{click}(R)}{\sum_{S_i} P_{view}(D, S_i) + \mu} \tag{2}$$

Note that many click models can also serve the purpose—estimating click probabilities. Here we do not mean to compare with them or claim superiority of any approach. We adopt the described approach mainly because the limited number of repeated observations in the dataset makes it difficult to train and evaluate click models. Our approach is probably a better one for this very specific situation, because at least $P_{view}(D, S_i)$ can be trained and evaluated using another reliable dataset with similar search settings. This is not a limitation of our approach, but a compromise to the data we have. We suggest readers to adopt large-scale and real search logs and the state-of-the-art click models [9, 16, 19, 30] to estimate click probability ground truth.

4. FEATURES

This section introduces features for ranking search results. We focus on features using information such as textual and structural characteristics of summaries. We do not use past clicks for the query-result pair outside the scope of the current session. This ensures that our approach can be applied to new queries and tail queries, for which past click information is not available in large scale. But it should be noted that for many popular queries and results, such information is available and can be adopted as ranking features as well.

Table 2 lists four categories of features adopted in our study. We analyze features by correlating with click probabilities among relevant and non-relevant results separately. We measure correlations using Pearson's r. Dark and light shadings stand for $p < 0.01$ and $p < 0.05$, respectively.

4.1 Query-independent Features

Table 3 reports correlations between query-independent features and click probabilities. Results show users are more likely to click on result summaries with shorter titles and URLs, but we did not find any significant correlations between click probabilities and snippet length.

Table 3: Correlation: query-independent features and P_{click}.

Features	**Pearson's r with P_{click}**	
	Relevant	**Non-relevant**
length title	−0.120	−0.067
length URL directory	−0.106	−0.036
%word <a>	−0.105	0.080
%word emphasis	0.049	−0.080
%word heading	−0.045	−0.077
%word <li>	0.055	0.088
%word <p>	0.136	0.020
max TF×IDF snippet	0.099	−0.028
pagerank	0.143	−0.067
spamrank	−0.092	−0.009

Table 4: Correlation: current query features and P_{click}.

Features	**Pearson's r with P_{click}**	
	Relevant	**Non-relevant**
QL	0.283	0.074
%qterm title	0.266	0.119
%qterm snippet	0.197	0.140
%qterm URL	0.208	0.069
min window all qterms	0.076	−0.021
min window bigrams	0.116	−0.103
min window trigrams	0.122	−0.011
#win k=10 all qterms	0.319	0.143
#win k=10 bigrams	0.187	−0.040
#win k=10 trigrams	0.135	−0.009

Click probability also correlates with web page structure. Results show that searchers are more likely to click on relevant results with a high proportion of actual content (e.g., texts in <p> tags) and a low percentage of navigational texts (e.g., <a> tags), but click errors often happen in web pages with a high proportion of navigational texts.

Click probability also correlates with the informativeness of words in summaries (e.g., as measured by the TF×IDF value of words). Web page authority (by page rank) is also positively correlated with click probability for relevant results, but negatively correlated for non-relevant ones.

4.2 Features Using Current Query

Table 4 reports correlations between click probability and features using information of the current query.

We adopt features measuring the similarity between query and result summary or web page, including coverage of query terms in summary title, snippet, and URL, and the query likelihood (QL) scores [35]. When matching terms in URLs, we count term occurrence by whether or not it is a subsequence of the URL. The original QL scores are not comparable between different queries. Therefore we normalize QL scores by query length, i.e., each query term is assigned a weight $1/|Q|$, where $|Q|$ is the length of the query. We found that all these features are positively correlated with click probabilities, regardless of among relevant or non-relevant results. Yet the correlations are consistently stronger in relevant results. Similar to the web page structure features in Section 4.1, the proportion of query terms in different HTML tags are also correlated with results' click probability.

We also use features matching query term phrases in web pages, including the number of windows of size k covering all query terms, and the minimum window size covering all query terms. We set the minimum window size to the docu-

Table 2: A summary of all ranking features in four different categories.

Query independent features	
Length	Length of summary title/snippet, with or without stop words.
Length_URL	Length of the URL (by characters/by levels of directories).
#unique_words	Number of unique words in title/snippet.
%stopwords	Percent of stop words in title/snippet.
%wordstag	Percent of words in <a>, <p>, <li>, <td>, heading tags, and emphasis tags (e.g., <b>, <em>).
TF/IDF	Average/max/min values of TF/IDF/TF×IDF for words in result summary title/snippet.
#fragment	Number of document fragments in the snippet.
pagerank	Pagerank for the web page (in percentile).
spamrank	Waterloo spam rank scores [13] for the webpage (in percentile).
Features using the current query	
%qterm	Percent of query terms in title/snippet/URL.
#qtermtag	Frequency of query terms in different HTML tags.
%qtermtag	Percent of query terms in different HTML tags.
QLscore	Normalized query likelihood score of the summary/web page.
#window(k)	Number of windows of size k in web page covering all terms, any bigrams, or any trigrams in query.
minwindow	Minimum window size in web page covering all terms, any bigrams, or any trigrams in queries.
Features using the past queries within the same session	
%past qterm	Percent of past query terms in title/snippet/URL. Separately consider ADD/KEEP/RMV terms.
%past qterm by Q	%past qterm considering two specific types of queries: 1) past queries w/ user clicks; 2) past query reformulations whose second query has clicks.
%past SERP term	Coverage of words from past clicked/skipped SERP titles in the summary's title/snippet.
QLscore past clicks	Normalized query likelihood score of past queries, past queries with clicks, past clicked results.
Repeated Result	Whether the same URL was clicked or skipped by the searcher on previous SERPs.
Situational features	
#previous clicks	The number of past clicks in the session. Separately consider SAT (>30s) and DSAT clicks (<15s).
#previous queries	The number of previous queries. We separately consider queries with clicks and without clicks.

Table 5: Correlation: past query features and P_{click}.

Features	**Pearson's r with P_{click}**	
	Relevant	**Non-relevant**
%past q	0.030	0.028
%past q clicked	0.037	0.125
%past q ADD	0.234	0.205
%past q ADD click-click	0.190	0.188
%past q ADD noclick-click	0.098	0.100
%past q RMV	0.129	0.106
%past q RMV click-noclick	0.026	0.108
%past q RMV click-click	0.055	0.098
QL pastq	0.018	0.075
QL pastq w/ click	−0.022	0.164
QL clicked snippet	−0.030	0.156

ment length when not all query terms occur in the web page. Most of these features are useful, as suggested by Table 4.

4.3 Features Using Session Information

We use the coverage of past query terms in summary title, snippet, and URL as features. We compute the coverage of terms for each past query and use the mean value as features. Following Guan et al.'s work [18], we also separately consider three types of terms in past query reformulations. For a query q_1 and its reformulation q_2, ADD terms are those in q_2 but not in q_1, KEEP terms are those in both queries, and RMV terms are those in q_1 but not in q_2. We compute the coverage of ADD, KEEP, and RMV terms for each past query reformulation and use their mean values as features. When calculating these features, we also consider different types of query reformulations, e.g., from a query without click to another with clicks (noclick-click). Table 5 shows that most past query similarity features are correlated with click probability of non-relevant results, while only a small number of them have significant correlations with click probability for relevant results. Results in Table 4 and Table 5 suggest that current query features are more correlated with click activity in relevant results, but past query features are more predictive of those in non-relevant results.

4.4 Situational Features

We also suspect searchers' click and skip decisions are related to situational factors that are independent of any specific results, i.e., in a certain period of a session, searchers are more likely to click on or skip results. Situational features include: the number of past clicks, SAT clicks (dwell time > 30s), and DSAT clicks (dwell time < 15s); the number of previous queries and those with clicks. Both features are correlated with click probabilities in non-relevant results, while the correlations are not significant in relevant results.

5. RANKING

We rank search results considering both relevance and the risks of click and skip errors. The task would be easy if we can make perfect predictions on relevance and click probabilities. In such case, we can simply apply rules in section 3 and rank results by predicted relevance and click probabilities. Unfortunately, making such perfect predictions seems beyond the ability of current technology.

We tackle this challenge by training LambdaMART rankers [7] to optimize an nDCG style metric. This metric encodes the ranking rules in Section 3. It also applies position-based discounts to put an emphasis on top-ranked results.

5.1 Click Sensitive nDCG (cs-nDCG)

Let $L = D_1, D_2, \ldots, D_n$ be a ranked list of n results. We measure D_i's contribution to the quality of the ranked list by $g(D_i)$, as in Equation 3, where: r_i is the relevance grade for D_i; p_i is the probability to click on D_i after viewing its summary, as we described in Section 3.3; g_p is the penalty to click on a non-relevant result.

$$g(D_i) = \begin{cases} (2^{r_i} - 1)p_i & r_i > 0 \\ -g_p p_i & r_i = 0 \end{cases} \tag{3}$$

Equation 3 weights the gain of a relevant result by its click probability. It uses the same gain function as in nDCG. In addition, Equation 3 sets an adverse effect for a clicked non-relevant results, which can set off the positive contribution (gain) of relevant results. The scale of the adverse effect is controlled by a parameter g_p, and we weight the adverse effect of a result by its click probability as well.

Similar to nDCG, we sum up the contribution of results at each rank, with a position-based discount function. We refer to the sum as cs-DCG for the ranked list, as in Equation (4). The position-based discount function is the same as the one used in nDCG, which ensures top-ranked results have a greater impact on the quality of the ranked list.

$$\text{cs-DCG}(L) = \sum_{i=1}^{n} \frac{g(D_i)}{\log_2(i+1)} \tag{4}$$

Click sensitive nDCG (cs-nDCG) for a ranked list is calculated by normalizing its cs-DCG to the range $[0, 1]$. Unlike DCG, cs-DCG may take negative values due to the penalty of clicking non-relevant results. Thus, we normalize cs-DCG by both its lower and upper bounds, as in Equation 5. The ideal ranked list (L_{best}) and the worst ranked list (L_{worst}) can be constructed by sorting documents by descending and ascending order of $g(D_i)$.

$$\text{cs-nDCG}(L) = \frac{\text{cs-DCG}(L) - \text{cs-DCG}(L_{worst})}{\text{cs-DCG}(L_{best}) - \text{cs-DCG}(L_{worst})} \tag{5}$$

When training LambdaMART models using cs-nDCG, we update using the following λ-gradients, where Δcs-nDCG is the cs-nDCG value gained by swapping a pair of results D_i and D_j in the ranked list.

$$\lambda_{ij} = S_{ij} \left| \Delta\text{cs-nDCG} \frac{\partial C_{ij}}{\partial o_{ij}} \right| \tag{6}$$

5.2 Properties of cs-nDCG

The gain function of the cs-nDCG metric, as described in Equation 3, ensures that the metric prefers search ranking conforming to the rules discussed in Section 3.1.

The metric satisfies R_1. When $p_1 \geq p_2$ and $r_1 \geq r_2 > 0$:

$$g(D_1) - g(D_2) = (2^{r_1} - 1)p_1 - (2^{r_2} - 1)p_2 > 0$$

Therefore, replacing D_2 by D_1 can increase the value of the metric. The metric also satisfies R_2, because when $r_1 > 0$ and $r_2 = 0$:

$$g(D_1) - g(D_2) = (2^{r_1} - 1)p_1 + g_p p_2 > 0$$

The metric also satisfies R_3, because when $r_1 = r_2 = 0$ and $p_1 < p_2$:

$$g(D_1) - g(D_2) = g_p(p_2 - p_1) > 0$$

Now we discuss the cases that are not included in R_1–R_3, i.e., $r_1 > r_2 > 0$ and $p_1 < p_2$. We analyze the chances that the metric will rank a less relevant result over a more relevant one, which is conflicting with existing relevance ranking of search results. In order to make the metric prefer D_2 over D_1, we should have:

$$\frac{p_2}{p_1} > \frac{2^{r_1} - 1}{2^{r_2} - 1}$$

Practically, when $r_1 = r_2 + 1$, the ratio p_2/p_1 needs to be at least as high as 15/7 (the case when $r_2 = 3$ and $r_1 = 4$) to make the metric prefer D_2 over D_1 in ranking. The click probability ratio has to be even higher when $r_2 = 2$ (7/3) and $r_2 = 1$ (3/1).

The following table shows the mean, maximum, minimum, and standard deviation of estimated P_{click} for results in our dataset. According to the table, the largest possible p_2/p_1 between documents with $r = 3$ and those with $r = 4$ is $0.82/0.64 = 1.28$, which is smaller than the minimum required ratio to rank $r = 3$ over $r = 4$ (15/7). Therefore, the metric would not rank $r = 3$ over $r = 4$ in our dataset. Similarly, the metric will not rank $r = 2$ over $r = 3$, because $0.69/0.32 < 7/3$. A small proportion of results with $r = 1$ may be preferred over those with $r = 2$. In order to be ranked over $r = 2$, a document with $r = 1$ requires at least 0.54 click probability. Only 5.8% of all results with $r = 1$ have a click probability higher than that value.

Relevance	Mean P_{click}	Min–Max P_{click}	SD P_{click}
4	0.74	0.64–0.87	0.089
3	0.59	0.32–0.82	0.133
2	0.44	0.18–0.69	0.110
1	0.36	0.11–0.65	0.101
0	0.39	0.08–0.74	0.091

Our analysis suggests that, in most cases, our approach will not conflict with conventional relevance ranking of search results. Of course, the chances vary by the estimated click probability in the dataset. However, considering it requires at least about 2 to 3 times greater click probability to rank a less relevant result over a more relevant one, it seems safe to conclude that in any reasonable dataset, there is only a slight chance that our approach will conflict with existing relevance ranking of results. This quality ensures that after introducing click probability into ranking, our approach will not radically decline relevance of results.

5.3 Incorporating Partly Judged Queries

In a practical search engine, relevance judgments are costly to scale up, but search logs are cheap and increasing all the time. Once we have relevance judgments for a set of queries, we may have more and more incoming queries providing new clicks. We can reuse the old relevance judgments for the new queries with the same topic, but these queries may have unjudged results. It is risky to simply assume that all unjudged results as non-relevant and put them into training. Instead, we simply set Δcs-nDCG to 0 if either or both results of a swapping pair has not been judged—we skip updating model in case of unjudged results. Similarly, when training using regular nDCG as the target metric, we can also incorporate partly judged queries by setting ΔnDCG and λ-gradients to 0 if either or both results of the swapping pair are unjudged.

The TREC 2014 session track dataset offers opportunities for studying whether it is helpful to use partly judged queries for training. Among the 1,257 sessions in the dataset, only results for queries in the first 120 sessions were fully judged. The rest of the sessions have overlap with the fully judged 120 sessions in topics. We reuse the relevance judgments for the 120 fully judged sessions to other sessions with the same topics in the dataset. The other sessions include unjudged results, and we only use these sessions for training.

6. EXPERIMENT

We use our approaches to re-rank the top 10 results for queries in the TREC 2014 session track dataset. We only re-rank the top 10 results because the dataset only provides the top 10 results. We evaluate results by both the relevance of the re-ranked list and the chances of click and skip errors (based on the estimated click probability ground truth).

We select queries which have full relevance judgments and at least one click on the top 10 results. In order to calculate features using past user interaction information, we do not select the first query of each session. This selects 145 queries. These queries are used for evaluation. To make use of partly judged queries, we further select other queries (excluding the first query of each session) with at least 2 results being judged and at least one click in the top 10 results. This selects 222 partly judged queries. We use 10-fold cross validation in all experiments: each fold uses 70% of the 145 fully judged queries for training, 20% for validation, and 10% for testing. We also use the 222 partly judged queries for training using the approach described in Section 5.3.

We do not evaluate results using cs-nDCG since this metric has not yet been validated as an evaluation metric. Instead, we evaluate results using regular nDCG@10 and metrics measuring click and skip errors in search result ranking.

One of the metric is discounted cumulated click & skip errors at rank k (DCE@k). Similar to discounted cumulated gain (DCG), we can measure the cumulated number of click and skip errors at rank k with a position-based discount, as in Equation 7 and 8. $[r_i = 0]$ and $[r_i = 1]$ are two binary variables that take the value 1 if their statements are true, or the value 0 otherwise. The overall DCE is defined as the sum of DCE$_{click}$ and DCE$_{skip}$.

$$\text{DCE}_{click}@k = \sum_{i=1}^{k} \frac{p_i[r_i = 0]}{\log_2(i+1)} \tag{7}$$

$$\text{DCE}_{skip}@k = \sum_{i=1}^{k} \frac{(1-p_i)[r_i = 1]}{\log_2(i+1)} \tag{8}$$

Another metric is the number of pairwise ranking errors. We count the following types of pairwise ranking errors in the re-ranked results:

- $R_{skip>click}$ is the cases that, for two equally relevant results, the one with a lower click probability (a higher chance of skip error) is at a higher rank. We separately count $R_{skip>click}$ for different levels of relevance as well ($r = 1, 2, 3, 4$).
- $NR_{click>skip}$ is the cases that, for two non-relevant results, the one with a higher click probability (a higher chance of click error) is at a higher rank.
- $NR{>}R$ is the cases that a non-relevant result is at a higher rank than a relevant one.
- $LowR{>}HighR$ is the cases that a less relevant result is at a higher rank than a more relevant one (but both results are relevant with $r > 0$). Note that this is not necessarily an error, but it can indicate how well search result ranking conforms to the conventional relevance ranking of search results (which requires $HighR{>}LowR$).

We compare our approach (the LambdaMART ranking model trained for cs-nDCG) with a few baselines. Little previous work targets this problem. Therefore, we compare with the LambdaMART ranking model trained for the regular nDCG. We also compare with the state-of-the-art ad hoc search models and contextual search approaches in a session with strong reported performance on the TREC session track datasets, including:

- Query likelihood with Dirichlet smoothing (QL) [35].
- Context sensitive relevance feedback (CSRF) [26]. CSRF is a relevance feedback approach considering past search queries and contents of clicked results within the same session. Variants of CSRF were ranked at the top in the TREC 2011 and 2012 session tracks [22].
- We also compare with the original ranking of results in the search log, and the perfect and worst possible rankings of results by nDCG and by cs-nDCG.

We report mean values of the evaluation metrics on the 145 fully judged queries in following sections. We test significant difference of results using a paired t-test.

7. EVALUATION

7.1 Click-Sensitive Ranking Models

Table 6 shows re-ranking effectiveness of our approaches and the baseline approaches using both fully judged and partly judged queries for training in each fold. We set $g_p = 1$ when using cs-nDCG for training.

Results show that our approach can significantly reduce the chances of click and skip errors with a slight decline in the relevance of the ranked list. As Table 6 shows, our ranking model trained for cs-nDCG@10 (“All Features (cs-nDCG)”) significantly reduced the overall DCE@5 by 9.6% (1.388 vs. 1.535, smaller values are better) comparing to the baseline model using the same features but trained for the regular nDCG@10 (“All Features (nDCG)”). DCE@5 for the click error alone fell by 19.1% (0.241 vs. 0.298), and that for the skip error alone was reduced by 7.1% (1.148 vs. 1.236). All the differences are significant at $p < 0.01$.

The effectiveness of our approach for reducing click and skip errors can also be verified by looking into $R_{skip>click}$ pairwise errors. This refers to the cases where results with higher chance of skip errors are ranked over equally relevant results with lower chance of skip errors. The total number of $R_{skip>click}$ pairwise errors fell by 37% (1.95 vs. 3.10). Such pairwise error is not common among $r = 4$ and $r = 3$, because only a very small fraction of relevant results have relevance grades 3 or 4 in the dataset. Most relevant results have relevance grades 2 or 1, for which the chances of $R_{skip>click}$ errors decreased by 45% and 35%, respectively. In addition, $NR_{click>skip}$ error also fell by 43%. This is the cases that non-relevant results with greater risks of click errors are ranked over those with fewer risks. Thus, both

Table 6: Comparison of ranking models (using all features) optimized for nDCG and cs-nDCG, and other baseline approaches.

Runs	**nDCG @10**	**DCE@5**			$R_{skip>click}$					***NR click>skip***	***NR>R***	***LowR >HighR***
		All	**click**	**skip**	**All**	$r = 4$	$r = 3$	$r = 2$	$r = 1$			
Original	0.210	1.284	0.399	0.885	2.77	0.01	0.08	0.43	2.26	8.83	6.32	1.96
Perfect (Relevance)	0.296	-	-	-	-	-	-	-	-	-	0	0
Worst (Relevance)	0.144	-	-	-	-	-	-	-	-	-	14.77	3.90
Perfect (cs-nDCG)	0.295	1.418	0.132	1.286	0	0	0	0	0	0	0	0.03
Worst (cs-nDCG)	0.144	1.078	0.752	0.326	6.35	0.02	0.32	0.94	5.07	19.86	14.77	3.87
QL (nDCG)	0.250	1.445	0.353	1.092	3.14	0.01	0.10	0.44	2.55	11.01	3.74	1.79
QL (cs-nDCG)	0.249	1.440	0.356	**1.084**	3.18	0.01	0.08	0.46	2.62	11.21	3.63	1.79
CSRF (nDCG)	0.255	1.454	0.364	1.089	2.98	0.01	0.04	0.43	2.48	11.07	3.53	1.61
CSRF (cs-nDCG)	0.254	1.451	0.364	1.087	2.98	0.01	0.08	0.46	2.27	10.11	4.33	1.68
All Features (nDCG)	**0.284**	1.535	0.298	1.236	3.10	0	0.08	0.53	2.50	11.48	**1.61**	**0.13**
All Features (cs-nDCG)	0.278	**1.388**	**0.241**	1.148	**1.95**	0	0.03	**0.29**	**1.63**	**6.51**	2.33	0.50

Bold font indicates the best results in its column (excluding the oracle runs). Dark and light shadings stand for $p < 0.01$ and $p < 0.05$.

metrics agree that our approach can successfully reduce click and skip errors in search result ranking. Note that the number of pairwise ranking errors is an $O(n^2)$ measure regarding the number of results n. This is why the improvements seem greater than those for DCE@5.

While reducing click and skip errors by about 10%–20%, the relevance of the ranked lists only slightly declined—nDCG@10 fell by 2.1% (0.278 vs. 0.284, $p < 0.01$). The number of pairwise ranking errors between relevant and non-relevant results ($NR>R$) increased by 45% (2.33 vs. 1.61), and that between more relevant and less relevant results increased by 284% (0.50 vs 0.13). It should be noted that the large relative scale of the differences come from the small number of these errors in the baseline approach ("All Features (nDCG)"). The actual increase in the number of pairwise ranking errors is small, as suggested by the small differences in nDCG@10 between the two approaches.

Comparing to other ad hoc search and session search baselines (e.g., QL and CSRF), our approach consistently outperforms all of them in both relevance of results and the overall chances of click and skip errors. All the ad hoc search and session search baselines have smaller DCE@5 for skip errors, but this is in fact because these approaches achieved limited relevance. Since less relevant results were ranked at the top, the chances of having skip errors is naturally smaller (since non-relevant results always have zero skip error). The advantage of our approach over these baselines is not surprising considering much more information is adopted into our ranking features comparing to the baselines. This also suggests our feature set is very effective for solving both relevance-based ranking and click-sensitive ranking.

"Worst (Relevance)" and "Worst (cs-nDCG)" provide upper bounds for different types of errors in the dataset. $NR>R$ and $LowR>HighR$ are pairwise ranking errors related to the relevance of search results, which sum up to a total of 18.87 in our dataset. $R_{skip>click}$ and $NR_{click>skip}$ are those related to click and skip errors, which sum up to 26.21. Our approach "All Features (nDCG)" reduced the relevance-based pairwise ranking errors to 1.74 (1.61 + 0.13), about 10% of the total possible errors. In contrast, "All Features (cs-nDCG)" reduced the click and skip error related ranking errors to 8.46 (1.95 + 6.51), about 1/3 of the total possible errors. This suggests that existing approaches can already solve over 90% of the ranking errors related to result relevance, but they can handle only about 2/3 of all possible click and skip errors, leaving large room for improvements.

7.2 Effectiveness of Features

Table 7: Ranking models using different features.

Runs	**nDCG@10**	**DCE@5**		
		All	**click**	**skip**
All Features	0.278	1.388	0.241	1.148
Web page	0.279	1.404	0.248	1.156
Title	0.272	1.369	0.285	1.083
Snippet	0.265	1.334	0.310	1.024
URL	0.255	1.340	0.342	0.998
Current Query	0.275	1.373	0.299	1.074
Past Queries	0.238	1.368	0.371	0.997
Q Independent	0.273	1.395	0.244	1.151

Dark and light shadings indicate significant differences with "All Features" at $p < 0.01$ and $p < 0.05$, respectively.

We further analyze the effectiveness of different feature sets in this section. Table 7 shows results for models trained using cs-nDCG ($g_p = 1$) with different sets of features.

We first compare features using web page full content ("Web page") with those using different elements of result summaries ("Title", "Snippet", and "URL"). As Table 7 shows, models using web page content can achieve better relevance of results, but significantly more skip errors ("DCE@5 skip"). Actually, features using only result snippet information can achieve the best overall DCE@5—even less errors than using the combination of all other features, but it achieved limited relevance of results. This indicates that result summary characteristics are important features for reducing click and skip errors in search result ranking, which is not surprising because searchers mainly made their click and skip decisions based on web page summaries. This result also suggests that it is important to include information of result summaries into ranking. Conventional approaches only use result web pages for ranking, while our experiments show it is beneficial to further incorporate information from result summaries to reduce the chances of click and skip errors. As Table 7 also suggests, combining title, snippet, and URL features with web page content features can lead to a slight decline of overall DCE@5, but no significant change in nDCG@10.

Note that our experiment setting inflated the effectiveness of result summaries in ranking relevant and non-relevant results. This is because all ranking candidates come from the top 10 results of a moderately effective system (the original system has 0.210 nDCG@10). Therefore, results in Table 7 do not mean that, by sorely using information from result summaries, we can achieve nDCG@10 close to those using full web page content.

Comparing features using different query information, we found that features using past query information ("Past Queries")

Table 8: Ranking models using different training data.

Runs	nDCG@10	DCE@5		
		All	click	skip
nDCG both	0.284	1.535	0.298	1.236
cs-nDCG both	0.278	1.388	0.241	1.148
nDCG full	0.280	1.521	0.307	1.213
cs-nDCG full	0.276	1.429	0.256	1.173

Dark and light indicate differences between "nDCG both" and "nDCG full", and between "cs-nDCG both" and "cs-nDCG full" are significant at 0.01 and 0.05 levels.

is the least useful. In contrast, features using the current query information alone ("Current Query") achieved similar performance to the full model ("All Features"), with only a slightly worse nDCG. This suggests that the inclusion of within-session user interaction features provide limited contribution to the problem in our dataset. It also suggests that our approach can be effectively generalized to cases where the local session context information is not available (e.g., the first query of a session).

7.3 Training Using Partly Judged Queries

We further evaluate whether incorporating partly judged queries into training can improve the effectiveness of the ranking models. We compare models using both fully judged and partly judged queries for training (labeled with "both") with those using only fully judged queries (labeled with "full")—each fold only used 70% of the 145 fully judged queries for training. We train ranking models using all features. Table 8 shows the results.

After incorporating partly judged queries ("cs-nDCG both"), we observed significant improvements in both relevance of results and declines in the chances of click and skip errors. Comparing to models trained using only fully judged queries ("cs-nDCG full"), we observed considerable declines in overall DCE@5 (1.388 vs. 1.429, $p < 0.01$) and click errors (0.241 vs. 0.256, $p < 0.05$). This suggests that using partly judged queries can significantly improve the effectiveness of search result ranking, achieving both higher relevance of results and fewer click and skip errors. After using partly judged queries, we also observed similar improvements for models trained using regular nDCG.

Results in this section suggest that, in a practical situation (e.g., a large search engine with abundant search logs), the effectiveness of our approach will not be restricted by the size of relevance judgments (which is usually small in scale comparing to query logs). Our study suggests that we can complement limited editorial relevance judgments using new click and skip observations on queries of similar topics, even when these new queries do not have full relevance judgments.

To conclude, results in section 7 demonstrate that the proposed approach can effectively reduce the chances of click and skip errors with a small decline in result relevance. Our analysis also suggests that existing approaches work relatively well enough in ranking relevant and non-relevant results, but have limited performance in ranking results with risks of click and skip errors. This leaves opportunity for future work.

8. DISCUSSION AND CONCLUSION

In this paper, we explored the instantiation of a new ranking paradigm of search systems—ranking results by not only relevance, but also how likely searchers may commit an error when viewing result summaries displayed on the SERP, i.e., clicking on a non-relevant result and/or skipping a relevant one. This ranking paradigm provides a more practical view of search engine result ranking comparing to existing approaches—purely ranking relevant results over non-relevant ones.

We instantiate the new ranking paradigm based on two parts: a set of features predictive of both result relevance and searchers' click and skip behavior; a metric measuring the quality of a ranked list considering both factors, which can be used to optimize ranking models. Results in section 7 suggest that using the cs-nDCG metric, we can train ranking models that can effectively reduce the chances of click and skip errors comparing to existing methods (the same ranker with same features optimized for regular nDCG). The cost for reducing click and skip errors is about a 2.1% decline in nDCG@10. It remains unclear whether or not it is worthwhile to trade 2.1% of nDCG@10 for optimized click and skip interaction. We leave this for future work. But we believe it is at least reasonable to question whether nDCG (as well as other metrics that purely consider relevance of the ranked list) is comprehensive enough to give insights to potential users' experience after they interact with the ranked list, because it requires both relevant results and correct interaction to deliver the useful information to the searchers. Failing to achieve either may fail to satisfy searchers.

Our study also stands for a new interactive search mode in which the system ranks results by not only results themselves, but also the possible ways of presenting results to the searchers (e.g., summaries). This is essentially a key advantage of our approach. Results in Section 7.2 also demonstrate improvements after incorporating result summary information into ranking. It may potentially be generalized to the form of search result ranking in which we have multiple candidate summaries for each result, and the task is to select not only the best ranking of results, but also the best presentations of results to be displayed on the SERP. The best ranking of results may also depend on the appropriate presentations of results as well. This stands for an end-to-end task of solving search result ranking and result snippet generation at the same time, while currently the two tasks are processed by separate procedures. Here we did not explore this possibility because our dataset provides only one summary for each result.

In addition, our approach does not rely on past click information from search logs or extensive contextual features to achieve its effectiveness. As Section 7.2 suggests, using only features related to the current search query, the ranking model can achieve effectiveness that is comparable to the full model (using all features). This shows that the approach can be generalized to cases where users' search history are not available.

Admittedly, there are also a few limitations in our study. Among which, the representativeness of the dataset is probably the most apparent and arguable one. The TREC session track dealt with relatively more complex search tasks [23] than those normally submitted to the search engines (e.g., navigational search). In addition, the search log provides limited impressions for each query-URL pair comparing to real web search logs, which makes the estimated click probability ground truth less reliable. Therefore, we believe it is necessary to re-evaluate this technique using a large scale dataset and real search logs.

To conclude, despite a few limitations, our study demonstrated the possibility and benefits of ranking search results considering both relevance of results and searcher click and skip errors. As Section 7 shows, existing IR approaches can already achieve strong performance when ranking relevant and non-relevant results, but they are not equally effective in ranking results with different risks of click and skip errors. This leaves us great opportunity and large room for improvements in the future.

9. ACKNOWLEDGMENT

This work was supported in part by the Center for Intelligent Information Retrieval and in part by NSF grant #IIS-0910884. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsor.

10. REFERENCES

[1] Overview of the special issue on contextual search and recommendation. *ACM Trans. Inf. Syst.*, 33(1), 2015.

[2] E. Adar, J. Teevan, and S. T. Dumais. Large scale analysis of web revisitation patterns. In *CHI '08*, 2008.

[3] E. Agichtein, E. Brill, and S. Dumais. Improving web search ranking by incorporating user behavior information. In *SIGIR '06*, 2006.

[4] E. Agichtein, E. Brill, S. Dumais, and R. Ragno. Learning user interaction models for predicting web search result preferences. In *SIGIR '06*, 2006.

[5] O. Arkhipova and L. Grauer. Evaluating mobile web search performance by taking good abandonment into account. In *SIGIR '14*, 2014.

[6] M. S. Bernstein, J. Teevan, S. Dumais, D. Liebling, and E. Horvitz. Direct answers for search queries in the long tail. In *CHI '12*, 2012.

[7] C. J. Burges. From RankNet to LambdaRank to LambdaMART: An overview. Technical Report MSR-TR-2010-82, Microsoft Research, 2010.

[8] B. Carterette, E. Kanoulas, M. Hall, and P. Clough. Overview of the TREC 2014 session track. In *TREC 2014*, 2014.

[9] O. Chapelle and Y. Zhang. A dynamic bayesian network click model for web search ranking. In *WWW '09*, 2009.

[10] L. B. Chilton and J. Teevan. Addressing people's information needs directly in a web search result page. In *WWW '11*, 2011.

[11] A. Chuklin and P. Serdyukov. Good abandonments in factoid queries. In *WWW '12 Companion*, 2012.

[12] C. L. A. Clarke, E. Agichtein, S. Dumais, and R. W. White. The influence of caption features on clickthrough patterns in web search. In *SIGIR '07*.

[13] G. V. Cormack, M. D. Smucker, and C. L. Clarke. Efficient and effective spam filtering and re-ranking for large web datasets. *Inf. Retr.*, 14(5):441–465, 2011.

[14] N. Craswell, O. Zoeter, M. Taylor, and B. Ramsey. An experimental comparison of click position-bias models. In *WSDM '08*, 2008.

[15] E. Cutrell and Z. Guan. What are you looking for?: An eye-tracking study of information usage in web search. In *CHI '07*, 2007.

[16] G. Dupret and C. Liao. A model to estimate intrinsic document relevance from the clickthrough logs of a web search engine. In *WSDM '10*, 2010.

[17] G. E. Dupret and B. Piwowarski. A user browsing model to predict search engine click data from past observations. In *SIGIR '08*, 2008.

[18] D. Guan, S. Zhang, and H. Yang. Utilizing query change for session search. In *SIGIR '13*, 2013.

[19] F. Guo, C. Liu, A. Kannan, T. Minka, M. Taylor, Y.-M. Wang, and C. Faloutsos. Click chain model in web search. In *WWW '09*, 2009.

[20] K. Hofmann, F. Behr, and F. Radlinski. On caption bias in interleaving experiments. In *CIKM '12*, 2012.

[21] J. Jiang, D. He, and J. Allan. Searching, browsing, and clicking in a search session: Changes in user behavior by task and over time. In *SIGIR '14*, 2014.

[22] J. Jiang, D. He, and S. Han. On duplicate results in a search session. In *TREC 2012*, 2012.

[23] J. Li, S. Huffman, and A. Tokuda. Good abandonment in mobile and PC internet search. In *SIGIR '09*, 2009.

[24] J. Luo, S. Zhang, and H. Yang. Win-win search: Dual-agent stochastic game in session search. In *SIGIR '14*, 2014.

[25] M. Richardson, E. Dominowska, and R. Ragno. Predicting clicks: Estimating the click-through rate for new ads. In *WWW '07*, 2007.

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

[27] M. Shokouhi, R. W. White, P. Bennett, and F. Radlinski. Fighting search engine amnesia: Reranking repeated results. In *SIGIR '13*, 2013.

[28] J. Teevan, E. Adar, R. Jones, and M. A. S. Potts. Information re-retrieval: Repeat queries in yahoo's logs. In *SIGIR '07*, 2007.

[29] A. Tombros and M. Sanderson. Advantages of query biased summaries in information retrieval. In *SIGIR '98*, 1998.

[30] C. Wang, Y. Liu, M. Wang, K. Zhou, J.-y. Nie, and S. Ma. Incorporating non-sequential behavior into click models. In *SIGIR '15*, 2015.

[31] R. W. White and E. Horvitz. Captions and biases in diagnostic search. *ACM Trans. Web*, 7(4), 2013.

[32] R. W. White, J. M. Jose, and I. Ruthven. A task-oriented study on the influencing effects of query-biased summarisation in web searching. *Inf. Process. Manage.*, 39(5), 2003.

[33] E. Yilmaz, M. Shokouhi, N. Craswell, and S. Robertson. Expected browsing utility for web search evaluation. In *CIKM '10*, 2010.

[34] Y. Yue, R. Patel, and H. Roehrig. Beyond position bias: Examining result attractiveness as a source of presentation bias in clickthrough data. In *WWW '10*, 2010.

[35] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *SIGIR '01*, 2001.

[36] F. Zhong, D. Wang, G. Wang, W. Chen, Y. Zhang, Z. Chen, and H. Wang. Incorporating post-click behaviors into a click model. In *SIGIR '10*, 2010.