Evaluating New Search Engine Configurations with Pre-existing Judgments and Clicks

Umut Ozertem
Yahoo! Labs
umut@yahoo-inc.com

Rosie Jones*
Akamai Technologies
rosie.jones@acm.org

Benoit Dumoulin†
Microsoft
bedumoul@microsoft.com

ABSTRACT

We provide a novel method of evaluating search results, which allows us to combine existing editorial judgments with the relevance estimates generated by click-based user browsing models. There are evaluation methods in the literature that use clicks and editorial judgments together, but our approach is novel in the sense that it allows us to predict the impact of *unseen* search models *without* online tests to collect clicks and without requesting new editorial data, since we are only re-using *existing* editorial data, and clicks observed for previous result set configurations. Since the user browsing model and the pre-existing editorial data cannot provide relevance estimates for all documents for the selected set of queries, one important challenge is to obtain this performance estimation where there are a lot of ranked documents with missing relevance values. We introduce a query and rank based smoothing to overcome this problem. We show that a hybrid of these smoothing techniques performs better than both query and position based smoothing, and despite the high percentage of missing judgments, the resulting method is significantly correlated (0.74) with *DCG* values evaluated using fully judged datasets, and approaches inter-annotator agreement. We show that previously published techniques, applicable to frequent queries, degrade when applied to a random sample of queries, with a correlation of only 0.29. While our experiments focus on evaluation using *DCG*, our method is also applicable to other commonly used metrics.

Categories and Subject Descriptors

H.3.3 [**Information Systems**]: Information Storage and Retrieval

General Terms

Experimentation, Measurement

Keywords

Relevance Estimation, Relevance Evaluation, User Browsing Models

*Work done while at Yahoo! Labs
†Work done while at Yahoo! Labs

WWW 2011, March 28–April 1, 2011, Hyderabad, India.
ACM 978-1-4503-0632-4/11/03.

1. INTRODUCTION

Any time we modify a web search or information retrieval algorithm, we would like to compare it to previous or competing algorithms. Traditionally this is done with metrics, such as discounted cumulative gain (DCG) [9], Bpref [2], expected reciprocal rank (ERR) [5], rank biased precision (RBP) [12], which are computed from human relevance judgments, which can be costly. More recently, quality metrics based on user interaction including click-through distributions have been introduced; however using these online metrics directly to test new search models can be risky, because unsuccessful configurations may hurt user experience while being tested to be able to collect the required user interaction data. In addition, state-of-the-art click models can make only predictions for high frequency query-URL pairs, and so can be over-confident in their estimates. Therefore, an inexpensive evaluation method reusing the existing judgments, and session logs from the previous search models is highly desirable.

We provide a method of reusing existing relevance judgments and combining them with the relevance estimates obtained by a user browsing model. Our experiments focus specifically on estimating the difference in Discounted Cumulative Gain (DCG) [9] between two search models (we will refer to this as ΔDCG), since this metric is widely used in the community. This approach is applicable to any evaluation metric for which we can calculate the mean and variance from random variables (such as Average Precision [3]), and we will elaborate on general applicability of our method.

In the literature, there are many user browsing models to estimate the relevance of documents from the user session logs. Some of these methods are based on the idea of providing pairwise preferences for pairs of documents. Joachims and Radlinski's methods are perhaps the most well-known of those belonging to this family [11, 14, 10]. They use heuristics like *skip-above* or *skip-next* to provide pairwise preference judgments for pairs of documents, which are not corrupted by *presentation bias*. Presentation bias is based on the fact that the users do not examine all the documents presented to them, and the click-through distributions are biased towards the higher ranked documents. There are also models that aim to provide the absolute relevance values [7, 8], a Bayesian Network by Piwowarski et. al [13] and a Dynamic Bayesian Network (DBN) based model by Chapelle and Zhang [6]. In this work, to evaluate the relevance estimates from session logs we use the Chapelle-Zhang DBN model. The next section gives a brief overview of this model,

as well as the reasons why this is the user browsing model of our choice.

In summary, we propose an evaluation metric for an *unseen result set configuration*, using the editorial data and clicks from previous configurations. Using this, we estimate the ΔDCG between two result sets, namely between the baseline model, and the unseen configuration. This paper is structured as follows. The next section gives a brief overview of the user browsing model used. In Section 3, we give a mapping from the outcome of the click model to editorial labels, and a method to do query and position based smoothing to substitute for missing relevance values. Then we present our experimental setup and provide details about the dataset used in the experiments. Afterwards, we show evaluations of the proposed smoothed-DBN model showing that it overwhelmingly outperforms (correlation of 0.74 versus 0.29) previously published methods when used over a random sample of queries, and approaches inter-annotator agreement. Finally we conclude the paper with a brief discussion of directions for future work.

2. DBN USER BROWSING MODEL

User browsing models use web search interactions including searches and clicks to estimate the relevance of documents to search queries. Some consider pair-wise interactions [11, 14, 10] to estimate relative relevance. Others use click-through patterns on one or more of the results to estimate relevance that is comparable across queries and URLs.

The DBN model by Chapelle and Zhang [6] deals with the result set as a whole. As well as the order of the results, it also takes into account the effects of other URLs, similar to the approach in the cascade model [7, 8]. The reasoning is as follows: a relevant result presented along with very relevant results gets few clicks, whereas the same document would have gotten many more clicks if it had been presented along with less relevant ones.

The DBN models the examination probability of a particular document as a function of its rank in the result set, as well as the quality of the other documents in the set. The model does not require any relevance judgments for training, and it can assign relevance scores to query-URL pairs where the URL has been seen several times for the given query, and the motivation is to infer two latent variables, defined as *attractiveness* and *satisfaction*.

Let e, c and s be binary random variables, indicating whether the user has examined, clicked to, or satisfied by a given search result, respectively. Chapelle & Zhang define *attractiveness* as the perceived relevance of a search result, formally $p(c = 1|e = 1)$, the probability that the user would click on the search result given that she examined it. Further, they define *satisfaction* as the landing page relevance, $p(s = 1|c = 1)$, the probability that the user would be satisfied with the result given that she clicked on it. After inferring these two latent variables, they model the relevance of a search result as *attractiveness*$\times$*satisfaction*.

The derivation and the complete details of the DBN model is out of the scope of this paper, but to provide motivation to the readers we would like to present the simplified version of it. In fact, it is notable is that this very basic simplified-DBN model that does not require any optimization performs only slightly worse than the DBN model optimized via expectation maximization.

2.1 Simplified DBN Model

Given a ordered result set, assume that the user examines the results from top to bottom, one by one, without skipping any of them. Also assume that the user keeps going deeper into the result set until she is satisfied, and stops afterwards. These two assumptions imply the following, $e_1 = \ldots = e_l = 1$, and $e_{l+1} = e_{l+2} = \ldots = 0$, in words, all the results until the last clicked one are examined, and the rest, the ones after the last clicked result are not examined. Then the latent variables attractiveness and satisfaction can simply be obtained by counting the following three values for each URL; N^u_{view}, N^u_{click} and $N^u_{last-click}$, namely the number of times the URL is examined, clicked, and is the last click of a search query[1]. Using these three values, the latent variables attractiveness (a_u) and satisfaction (s_u) for each URL are defined as

$$a_u = \frac{N^u_{click}}{N^u_{view}}, \quad s_u = \frac{N^u_{last-click}}{N^u_{click}} \tag{1}$$

Intuitively, a_u is the ratio of times that it is clicked versus the times that it gets clicked or skipped[2]. Similarly, s_u of a URL is the ratio of times that the users end their search (hence, do not go back to the search result page and click to another result) after clicking this particular result.

Overall, the DBN model not only provides better estimation accuracy than earlier models, it is also by nature well connected to earlier models. The examination model, and the cascade model are special cases of this model, and with the way the examination likelihood (or N^u_{view} in the simplified model) is defined, DBN inherently implements the skip-above pairs type of ideas like Joachims's work. Also note that in the following there is nothing specifically adjusted or fine-tuned for this particular model; hence, one can plug-in any other user browsing model that gives absolute relevance values. In the experimental results section, we will show that similar results can be obtained with ordinal regression as well.

3. EVALUATION MODEL

We would like to evaluate a new ranking model by comparing with a baseline, and looking at the difference in the chosen metric. The steps consist of (1) express the change in the metric in terms of a function of the means and variance of a probability density function over the metric (2) mapping the estimates from the click-based model to judgments for the metric by fitting a distribution to data in the *intersection* (3) computing estimates for the remaining missing values using query and position based smoothing. The intersection is the portion of the query-URL pairs that we have both editorial judgments and the user browsing model estimates. Note that, although derivation of the performance evaluation method in the beginning of this section is given for DCG [9] only, the techniques in the rest of the paper apply to any other performance metrics based on human

[1]They do not model query chains and reformulations jointly, so there is no definition of a *session*. Therefore, the queries are handled independently, and the last click is considered per query basis.

[2]Skipped means the result is examined but not clicked. Note that for a URL the view count is not increased unless a lower ranking result is clicked.

relevance judgments like MAP, ERR, or RBP as well. The derivation of MAP as a random variable has been given by Carterette et al. [3].

3.1 Delta DCG

For each pair of new and baseline model pairs, we will evaluate the ΔDCG_n for n=5 and DCG formula is given by

$$DCG_n = \sum_{i=1}^{n} \frac{rel_i}{\log_2(i+1)} \tag{2}$$

where normally rel_i values are the relevance scores by human judges in a 5-scale grades. These grades later translated into a numeric scale to be used in the DCG formulation. At this point, following the derivation in [4], we introduce a random variable g_i representing the relevance of the document i, and rewrite the DCG formula as

$$DCG_n = \sum_{i=1}^{n} \frac{g_i}{\log_2(i+1)} \tag{3}$$

We would like to introduce this random variable because estimating a particular grade from user browsing model based estimates is quite hard, whereas estimating the probability density function (PDF) of a predefined parametric form is relatively more reliable. This will be clear in the next section. Given g_i, the grade probability of document i, the probability distribution $p(g_i)$ is multinomial. Let $\{a_1, \ldots, a_5\}$ be the set of numeric scale values to be used in DCG estimation that correspond the 5-scale grades, such that $a_1 > a_2 > a_3 > a_4 > a_5$. We define p_{ij} such that $p_{ij} = p(g_i = a_j)$ for $1 \leq j \leq 5$ and $\sum_{j=1}^{5} p_{ij} = 1$. The expected grade $E(g_i)$, and its variance $Var(g_i)$ are given by

$$\begin{aligned} E\{g_i\} &= \textstyle\sum_{j=1}^{5} p_{ij} a_j \\ Var\{g_i\} &= \textstyle\sum_{j=1}^{5} p_{ij} a_j^2 - E\{g_i\}^2 \end{aligned} \tag{4}$$

Using $E\{g_i\}$ and $Var\{g_i\}$, we write the mean and variance of the DCG. The mean is given by

$$E\{DCG_n\} = \sum_{i=1}^{n} \frac{E\{g_i\}}{\log_2(i+1)} \tag{5}$$

Assuming the relevance of each document is independent of each other (hence, $Cov(g_i, g_j) = 0$), the variance of DCG is given by

$$Var\{DCG_n\} = \sum_{i=1}^{n} \frac{Var\{g_i\}}{(\log_2(i+1))^2} - E\{DCG_n\}^2 \tag{6}$$

To evaluate the new model, we sample from the queries that this new model affects (which, in general, may or may not be the all queries). For a particular query in this set, let D and D' be the set of results obtained with the search engine when the model to be tested is turned on (new configuration) and off (the baseline model), respectively. We want to evaluate $E\{\Delta DCG_n^{DD'}\}$ and $Var\{\Delta DCG_n^{DD'}\}$. However, simply evaluating the DCG for these two sets independently and looking into the difference (that is $\Delta DCG_n^{DD'} = DCG_n^{D} - DCG_n^{D'}$) is not optimal here, and one should compute the $E\{\Delta DCG_n^{DD'}\}$ directly. (For example, if a particular document i is at the same rank both in D and D', the variance of g_i for this document should cancel out and not affect the $Var\{\Delta DCG_n^{DD'}\}$.)

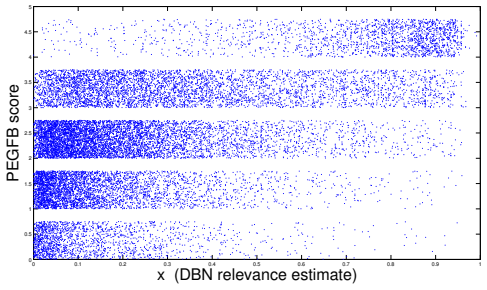

Figure 1: DBN relevance estimates vs. editorial scores in PEGFB scale (Perfect - Excellent - Good - Fair - Bad)

Given r_i^D and $r_i^{D'}$ the ranks in each result set, we define the discounts for all documents conditioned on both result sets D and D':

$$d_i^D = \begin{cases} \frac{1}{log_2(r_i^D+1)} & \text{if } 1 \leq r_i^D \leq n \\ 0 & \text{if } D \text{ does not have document } i \end{cases} \tag{7}$$

Finally using the discounts for each result set, one can write the ΔDCG over the union of the documents in both result sets

$$E(\Delta DCG_n^{DD'}) = \sum_{i \in D \cup D'} E(g_i)\ (d_i^D - d_i^{D'}) \tag{8}$$

In the next two sections, we will focus on the estimation of $E(g_i)$ for all the query-URL pairs from different sources; editorial data, user browsing model, and query-position based smoothing.

3.2 DBN Relevance Estimates vs. Editorial Judgments

To understand the relationship between DBN relevance estimates and editorial judgments, we inspect the intersection of the editorial and the DBN relevance data, namely the query-URL pairs for which we have both an editorial judgment and a DBN relevance estimate. This is shown in Figure 1, where each point represents a query-URL and the vertical and horizontal axes are the DBN relevance estimate x and the editorial scores, respectively. A noise jitter is added to the vertical axis for visual purposes. Note that DBN scores tend to correlate with editorial scores, so these quantities are predictive of each other. Since they are not perfectly predictive, we will relate them to each other in terms of probability estimates, as will given next.

3.3 Mapping the Browsing Model Estimates to Editorial Judgments

Although it is possible to use the relevance estimates from the browsing model to estimate the ΔDCG using only session logs, incorporating the existing editorial data into the evaluation process is of great interest because click models can estimate relevance only for queries that appear several times in the query logs (at least 10 times in the DBN model). Therefore they tend to have a bias towards more frequent queries. Since the infrequent queries are still a significant fraction of traffic to the search engine, an evaluation method based only on more frequent queries might be overconfident

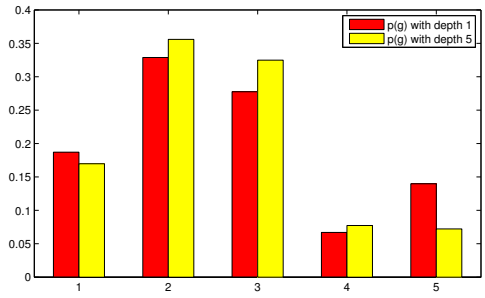

Figure 3: The prior grade distribution $p(g)$

of the estimates provided. One can solve this problem by combining editorial judgments from infrequent queries into the evaluation process.

We denote the relevance estimates from the DBN browsing model as x_i. Note that since the DBN is a probabilistic model, and the relevance estimates that it provides are in the range [0,1]. Therefore, if one needs to combine the DBN estimates with some existing editorial data, one needs to compute a mapping between the relevance estimates of the DBN and the editorial grade scale $\{a_1, \ldots, a_5\}$.

Note that we could map the DBN scores to the editorial grade scale, or the reverse. In this case we chose to map to the editorial grade scale for ease of understanding for those familiar with working with DCG based evaluation. Note that there is no loss of information in mapping in either direction, since we can work in probability density functions. To obtain this mapping, we inspect the intersection of the editorial and the DBN relevance data, the query-URL pairs that we have both editorial grades and DBN estimates, shown in Figure 1 and described above.

Given its DBN relevance estimate and the numeric scale values of the editorial grades $\{a_1, \ldots, a_5\}$, our aim is to estimate the grade probability distribution of any query-URL pair. To estimate the multinomial grade distribution $p(g_i)$ for each x_i, hence to obtain p_{ij} values, one should first estimate the prior distribution of grades $p(g)$ and the grade conditional DBN estimates $p(x|g)$, and employ Bayes rule. Since g is a discrete valued random variable, obtaining $p(x|g)$ is quite straightforward by fitting individual models for each grade separately. We fit beta distributions to the DBN results of each grade, as shown in Figure 2.

The prior grade distribution $p(g)$ is the marginal of the $p(g_i)$. This multinomial distribution can simply be obtained by counting the samples for each grade. However, note that this should be done for the same ranking function in a position aware manner. For example, if $p(g)$ is going to used in the estimation of DCG_1, one should use the histogram of the top ranked judged documents. Similarly for the estimation of DCG_5, one should use the histogram of the documents in top 5. For the judged dataset we have, both of these are shown in Figure 3.

Using Bayes rule, one can estimate the posterior to be $p(g|x) \sim p(x|g)\ p(g)$ with a missing normalization constant. Figure 4 shows $p(g|x)$ for both prior distributions. The posterior probability $p(g|x)$ with the grade prior probabilities obtained from the whole editorial data is given in Figure 5, evaluated at 10 points uniformly sampled along x.

Table 1: Inter-editor agreement matrix

	P	E	G	F	B
P	183	82	24	7	1
E	82	338	724	117	22
G	24	724	1395	844	92
F	7	117	844	658	293
B	1	22	92	293	230

Table 2: Mean and variance of each grade distribution estimated from the inter-editor agreement matrix

	P	E	G	F	B
$E(g)$	3.48	2.27	1.91	1.42	0.86
$Var(g)$	0.59	0.61	0.65	0.69	0.64

On the other hand, estimating the grade probability from editorial data is relatively simpler. The most straightforward approach is, for a particular query-URL, to select $p_{ij} = 1$ for the given grade. Note that this approach assumes that the grades are *exact*, that is the grade pdf $p(g)$ has zero variance. In fact, that is the assumption by Carterette and Jones [4], when they combine editorial data and click-based relevance estimates. However, it is well known that this assumption is not valid, since the judges hardly ever fully agree with each other [1, 16, 15]. To be able to relax this assumption, we measured the inter-editor agreement for a query-URL data set that has three judgments given by three different editors independently, to account for the variance of each grade[3]. The query-URL dataset consists of 5 URL per query for 334 random queries, resulting in 1670 unique query-URLs and 5010 judgments in total. The three judges for each query-URL are randomly assigned from a pool of 42 individuals. Cross-comparing the grades for the same query-URL pairs given by different editors, one ends up with a 5×5 matrix as shown in Table 1, where the disagreement can be measured by the non-diagonal entries.

Given the grade, one can use the rows (or columns) of the inter-editor agreement matrix to estimate the underlying multinomial grade probability distribution by normalizing by the sum of each row. $E(g)$ and $Var(g)$ values of these multinomial distributions for each grade in the DCG estimation is summarized in Table 2, and these are the values we will use in our experiments.

3.4 Query and Position Based Smoothing

In general, the user browsing model and the pre-existing editorial data cannot provide relevance estimates for all documents for the selected set of queries. To estimate the missing judgments, we use a query and position based smoothing over the existing relevance values, obtained either through editorial data or the user browsing model.

In Table 3 we see an illustration of the grades of top five documents, where the relevance judgments are given in a five scale of Perfect Excellent Good Fair and Bad, and × rep-

[3]This data set with multiple judgments per query-URL is different than the one we use for the evaluation of the method in the Experimental Results section.

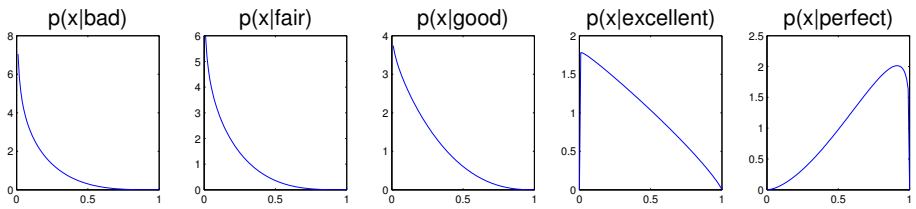

Figure 2: beta distributions for $p(x|g)$

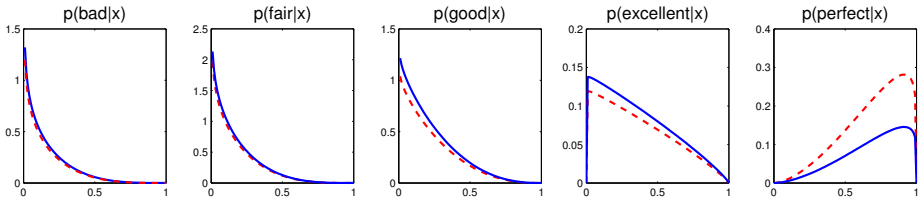

Figure 4: $p(g|x)$ evaluated with respect to the $p(g)$ of depth 1 (dashed) and 5 (solid) as shown in Figure 3.

Table 3: An illustration of grades of top 5 documents for different queries, where missing values are shown with $\times$. For q_1 query-based smoothing seems the most natural, and we might guess that the missing grade is Bad. For q_5 a position-based smoothing is more natural - assuming the document at rank 1 has typical relevance for a rank 1 document, and so on.

q_1	q_2	q_3	q_4	q_5
B	$\times$	P	G	$\times$
B	$\times$	E	$\times$	$\times$
$\times$	F	$\times$	$\times$	$\times$
B	B	$\times$	$\times$	$\times$
B	B	B	$\times$	$\times$

resents missing judgments. For the cases where we have *a sufficient number of grades* for the query and the available grades are close to each other (like in q_1 and q_2), query-based smoothing is more efficient. Where there are a few -or perhaps no- grades (like in q_4 and q_5) and/or the available grades have a *large variance* (like in q_3), position-based smoothing becomes more efficient. Therefore, we want a hybrid smoothing model that combines these two models in a query-dependent manner. For simplicity, the derivation here is given for only one of the result sets, D. One should repeat the same for D', and for the documents that both appear in D and D', use the average of these two estimates.

Let us define two basic smoothing methods.

Position-based smoothing: Define $p_r^D(g)$ as the average grade pdf at rank r, averaged over all queries and document set D, which is given by

$$p_r^D(g) = n_r^{-1} \sum_{d \ @ \text{ rank } r \text{ in } D} p(g_d) \tag{9}$$

Query-based smoothing: Define $p_q^D(g)$ as the average grade pdf in D for the query q, given by

$$p_q^D(g) = n_q^{-1} \sum_{d \text{ in } D \text{ for } q} p(g_d) \tag{10}$$

Here n_r and n_q denote the number of available relevance judgments for rank r or query q, respectively.

Query and position based smoothing: One important observation is that the query based smoothing performs much better when (1) there are more than a couple grades for that query (2) the grades are similar or the same. On the other hand, although it is more robust and performs equally well regardless of the number of available grades per query, position based smoothing performs worse than query based smoothing on average. Over a random set of queries, the percentage of missing judgments per query is rarely uniform, tail queries may have only a few judged or click estimated documents, whereas more frequent queries have much more available click estimated relevance values due to much bigger user interaction data.

Since the performance of these two methods vary from query to query, we define a hybrid smoothing model that combines these two smoothing methods in a query-dependent manner to get the best of the both. The hybrid method is a weighted combination of the two, and adapts the weights depending on the number (and distribution) of the available relevance values. For this we define a meta-parameter σ that gives what exactly the *sufficient number of grades* and *large variance* mean. In Section 5.1 we give a leave-one-out cross-validation method to learn the optimum value of σ from the data directly. Consider the following formulation

$$p_s(g) = w_q \; p_r^D(g) + (1 - w_q) p_q^D(g) \tag{11}$$

where w_q are query-dependent weights. Intuitively, we want w_q to increase (hence bias towards query-smoothing) when, the number of available grades for that query increases and the available grades have a lower variance.

For a particular query q, we define the mean relevance as the average of the available (graded or click-estimated) relevance values

$$\mu_q^D(g) = \frac{1}{N^D} \sum_{d \text{ in } D \text{ for } q} E(g_d) \tag{12}$$

where N^D is the number of documents in D. Hence, we select the weight as a function of the variance and the number of the available relevance values for this query

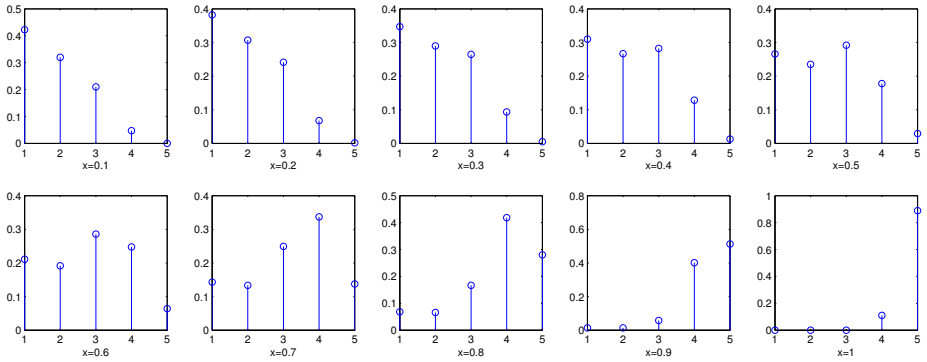

Figure 5: $p(g|x)$ at 10 uniformly sampled points along x (for depth = 1)

$$d_q = \frac{1}{N^2}\left(\sum_{d\in D}(E(g_d) - \mu_q^D)^2 + Var(g_d)\right)$$
$$w_q = e^{-d_q/\sigma^2} \quad (13)$$

Here we have a parameter σ, and for $\sigma = 0$ and $\sigma \to \infty$, (10) converges to position- and query-based smoothing, respectively. The optimal value of σ can be selected by using leave-one-out cross validation over the existing relevance values and using Fibonacci search (or similar). Note that for each novel-baseline search configuration model pair, the optimal value of σ varies, depending on the percentage of the missing judgments in the whole dataset as well as how the missing judgments are distributed over queries. If the missing judgments are somewhat evenly distributed over the query set, the optimal σ turns out to be smaller. Conversely, if the missing judgments are concentrated for a portion of the dataset, that is many queries have very few or no missing judgments whereas some other have a lot, the optimal σ value turns out to be higher. We will show experiments and estimation error results regarding the proposed smoothing model.

So far we have presented three ways of relevance information: the editorial data, relevance estimates of the user browsing model, and missing relevance values estimated by query-position smoothing. To combine these we use the following order:

1. For each available grade, get the corresponding row of Table 1, normalize and use as the editorial grade pdf $p(g)$.
2. If the editorial data is not available, employ the user browsing model and use $p(x|g)p(g)$
3. If the user browsing model also doesn't have a relevance estimate due to insufficient session logs for this query use the query-position smoothing given in Equation (11).

4. EXPERIMENTAL SETUP

Before presenting the experimental results we start with providing the details of the data sets used in the experiments and the overall experimental setup.

For this experiment we define 5 *tests*. Each test is a baseline-novel search engine pair, and each test will be conducted on the set of queries that the corresponding novel model affects -hence, the query sets in each test are not random splits. In our experiments, all tests are for evaluating new version of a query rewriter (QRW). In these tests, the novel search engine configuration (which returns the document set D) is basically the search engine with new QRW model that has yet to be tested online, and the baseline configuration (that returns the document set D') is the search engine with the existing QRW model. Although we design our tests to evaluate QRW methods, the model can be used for other binary query classifiers designed for specialized experience, for example, the classifiers that are used for triggering the shopping, local or news modules. A positive $E(\Delta DCG^{DD'})$ means that the new QRW model is more successful than the baseline.

The query set consists of 1785 uniformly sampled queries from the *affected* queries of each QRW model. The sizes of the query sets are different for each test, proportional to the coverage of the affected queries by each QRW model. Hence, for all the queries in the dataset, the test configuration is likely to (and most likely does) give search results different from the baseline system with the old QRW model.

The number of test queries affected by the test configurations, and the total number of unique query-URL pairs (at top 5 ranks) for each model as well as number of available editorial judgments are given in Table 4. Note that the number of unique query-URL pairs is not simply 5 × number of queries, since the total number of documents needed per query is the union of URLs returned for the novel configuration and the baseline configuration. The more aggressive the model is, the more new URLs it will introduce, and therefore, the bigger the average number of URLs per query ($D \cup D'$) becomes.

The judged query-URL pairs constitute to 34.5% of the unique query-URL pairs needed for the 5 tests. These are editorial judgments in the PEGFB scale that have been collected independently for previous experiments within the last three months. The DBN user browsing model is trained on three months of user click logs of a major search engine, and it produces an output for the queries-URL pairs that have been examined at least 10 times ($N^u_{view} \geq 10$ for the simplified DBN model presented in Section 2.1). For the query-URL pairs without an editorial judgment, the percentage of query-URLs with a DBN relevance estimate is around 30.5% of all the required ones in our dataset.

We use the data described above and given in the 3^{rd} and

Table 4: Statistics of the training data, from a total test set of 1785 unique queries

	Test1	Test2	Test3	Test4	Test5
unique queries	447	707	911	293	55
unique query-URL	2803	4878	6404	2016	382
judged query-URL	879	1738	2304	684	130
DBN query-URL	733	1596	1949	622	124

4^{th} row of Table 4 as the training data of our evaluation model and estimate the ΔDCG for the five tests. To measure the accuracy of our evaluation method, we compare it to fully editorially calculated DCG values. For this we collected editorial grades on the same PEGFB scale that covers all the unique query-URLs needed for all tests (2^{nd} row in Table 4), and use it as our evaluation data. Furthermore, to underline the fact that the judges do not agree with each other as also mentioned in Section 3.3, we get this full query-URL set judged three times with random judge assignments from a pool of 49 individuals. We will use the multiple fully judged query-URL datasets to measure how well a fully editorial judgment based ΔDCG value correlates to another independent fully editorial judgment based ΔDCG value, and we will show that our method significantly approaches this inter-annotator correlation value.

5. EXPERIMENTAL RESULTS

The accuracy of the smoothing technique and the performance of the proposed method and comparisons with an earlier method as well as correlations with fully editorial judgment based ΔDCG values will be presented here. First we show a comparison of the three smoothing methods presented in Section 3.2, showing that all improve over Zhang and Chapelle's unsmoothed DBN models, and then showing how to select an optimal interpolation weight. Since the derivation also builds on the formulation by Carterette and Jones [4], we also provide some comparisons with this method to underline the differences of our approach. We show our method significantly outperforms theirs on a random sample of queries in all frequency ranges. Finally we show the performance of our evaluation method for five different search engine tests and compare the results with fully editorially judged ΔDCG.

5.1 Comparison of the Smoothing Methods

In this section we evaluate the performance of query-based smoothing, position-based smoothing, and query/position based smoothing, by looking at their accuracy at predicting the relevance judgements scores for individual query-URL pairs. Recall from Section 3.3 that the ordinal relevance judgments are mapped to the scores $\{1, \ldots, 5\}$. We can make a prediction of each query-URL pair, and compare to the true relevance grade score, then calculate the mean-squared error (MSE) over all predictions. We compare the mean-squared error (MSE) and error probability densities of these three smoothing models in leave-one-out cross-validation sense. The leave-one-out procedure is as follows:

1. Take all the judged documents, and remove one judged document from the dataset (denote this as g_j).

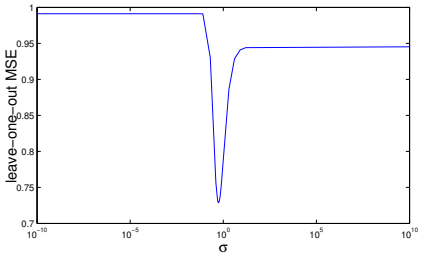

Figure 6: Leave-one-out MSE for the range of σ values (horizontal axis in the logscale)

2. Plug in all the remaining judged documents into the estimator (8), (9) or (10), to estimate the grade of the left out document d_j (denote the estimate as $\widehat{g}_j$).
3. Estimation error for the j^{th} document is given by $e_j = g_j - \widehat{g}_j$
4. Repeat the above for all judged documents in the set, report average squared error $E\{e\}$ over all samples.

One can use the leave-one-out procedure to evaluate the smoothing method, as well as finding the optimal granularity parameter σ. Figure 6 shows the leave-one-out MSE for varying σ values. Note that for $\sigma = 0$, Equation (11) converges to the position based smoothing, which gives 0.9911 MSE. Similarly, for $\sigma \to \infty$, Equation (11) converges to query based smoothing with a MSE of 0.9452. At $\sigma = 0.55$ the hybrid smoothing method in (11) gives a MSE of 0.7286, leading to a significant $23-26\%$ improvement in estimation error with respect to the other two methods.

Again for the same leave-one-out evaluation, Figure 7 compares the error pdf of three methods, query smoothing, position smoothing and query-position smoothing with $\sigma = 0.55$. Investigating the error pdf's one can see that both query and position smoothing lead to *systematic errors*, a number of peaks in the pdf that are not centered at zero. The error pdf's are obtained by standard kernel density estimation with Gaussian kernel function. The variance of the Gaussian kernel is selected to be 0.1 for all three methods. One can argue that it is possible to get rid of these peaks by using a wider Gaussian kernel, but the error samples themselves clearly show the systematic error behavior as well. On the other hand, the hybrid query-position smoothing method leads to a more Gaussian-like error distribution with a single peak at zero, resulting in a much lower error entropy in smoothing.

In Table 5 we see a summary of the MSE under different smoothing methods. Best results are obtained by query/position based smoothing, the interpolation of the other two. The best results are with the parameter σ set to 0.55. We use this smoothing method (with the optimum parameter) for the remainder of our experiments, and refer to it as *smoothed-DBN*.

5.2 Evaluation Accuracy

In this section we show that our smoothed DBN model significantly outperforms models from previous work, and approaches inter-annotator agreement.

Table 5: Leave-one-out MSE for smoothing. Lowest error gives best results. Best results are with interpolated smoothing.

Query-smoothed	Position-smoothed	Query/Position-smoothed ($\sigma = 0.55$)
0.9452	0.9911	**0.7286**

5.2.1 Comparison with Carterette&Jones Method

Since the formulation in Section 3 builds on the evaluation method by Carterette and Jones [4], we would like to point out the novelties of our approach and provide some comparisons. To model the relevance using clicks, they use ordinal regression between the grades (same five-level scheme that we use) and the clickthrough rates of the document and the clickthrough rates of the documents presented above and below. As they show in their results this model works very well; however, the shortcoming of this approach is that it requires a lot of user interactions, therefore, they limit their training set to queries that have at least 200 impressions in their session log data. Similarly, they use queries with at least 500 impressions as their test data.

Evaluation based on only frequent queries may tend to be overconfident of the underlying metric, since the infrequent queries that are neglected in the evaluation constitute a significant portion of the overall query volume. Therefore, while collecting editorial judgments for evaluation, its a common practice in the community to evaluate the ranking models with a uniformly sampled random set of queries, and that is what we use as the test data of our method in our experiments. In our uniformly random query sample of 1785 queries, the number of queries that have more than 200 impressions is 122, and for 500 impressions 52. Hence, the training and testing cases in Carterette and Jones paper only constitute for a tiny portion of the uniformly sampled query set we use. On the other hand with the DBN model, it is possible to obtain reliable relevance estimates with as few as 10 impressions, and for the queries with less than 10 impressions we rely on the available editorial data and the smoothing model. With fewer impressions, of course the statistical variance of these DBN relevance estimates are higher than the ones obtained via more impressions, but the estimates from 10 impressions prove to be useful since they correlate well with the fully editorially evaluated ΔDCG as we will show in the next section.

One other significant difference is that Carterette and Jones use the clicks that are collected over the ranking models that are being compared; on the other hand, we only use the clicks collected over the earlier ranking models to avoid the cost of running an online test. Here, we rely on the fact that the previous ranking models also return some documents in common to get some portion of the click interaction data for free, however comes in with the unavoidable cost of some missing data. Another important point here is that since we only use pre-existing judgments for evaluation, and do not request any additional judgments. When they combine the click-based estimates with editorial judgments, Carterette and Jones select the query-URL samples to be judged after observing the results retrieved by the two ranking models to be compared, and they give an algorithm to get judgments for the most informative query-URL samples in the data set. On the other hand, the editorial data that we use for the same purpose is only some pre-existing

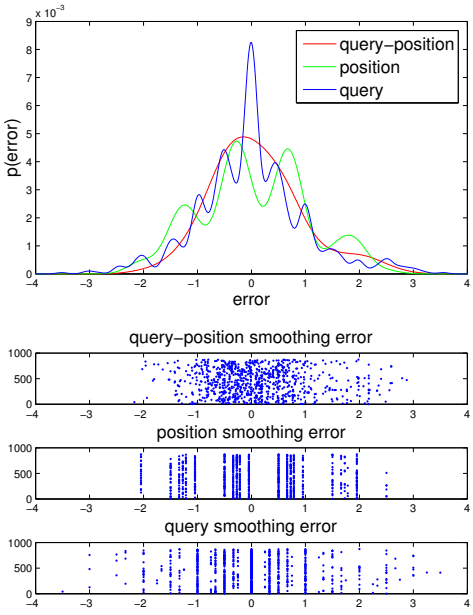

Figure 7: Leave-one-out estimation error probability densities for three smoothing methods (top), and the error samples of the leave-one-out estimation.

judgments that are collection of independently conducted editorial tests as the validation or testing sets of other research projects, and our aim is to reutilize these existing judgments. Therefore, the judgments that we use are not tuned for the most informative samples (which differentiate the two compared configurations the most) by any means. In fact, on average 23% of the pre-existing editorial and click-estimated relevance data in our training set belongs to documents that are *ranked at the same position in D and D'*, which do not even differentiate the two compared configurations at all. Although these samples are useful at the smoothing step to estimate the missing relevance judgments, they do not directly affect ΔDCG. In summary, our approach is significantly different in the sense that we require no online test, and do not request any editorial data and work with only pre-existing click and editorial data.

We experimented with the ordinal regression method by Carterette and Jones to provide comparisons. Since their method is particularly designed to work with queries with large numbers of user interactions, we varied the comparisons to the subsets with more than k impressions in the dataset for $k = 200, 50, 10, 0$. We present the average cross-correlation with fully editorially estimated ΔDCG results over these subsets of the query data for 5 test configurations in the in Table 6. The ordinal regression method is slightly more accurate than our smoothed-DBN for $k \geq 200$, presumably because both ordinal regression and DBN models have enough samples to converge and there is very little or no need for smoothing. As less and less frequent queries are included into the query set, our model is significantly superior, since the DBN model converges a lot faster and requires a lot less impressions, and the smoothing model is helping to fill in the missing relevance values. Over a random sample of queries ($k \geq 0$) Carterette and Jones has a cross-correlation of only 0.29 with the fully labeled data, while the smoothed-

Table 6: Average cross-correlation with fully editorially estimated data for different query sets. Carterette and Jones' method degrades when we include queries with fewer impressions, while our smoothed method is robust to samples including rare queries. All difference in cross-correlation are statistically significant.

Query Set	Carterette & Jones	Smoothed-DBN
$\geq$ 200 impressions	0.79	0.78
$\geq$ 50 impressions	0.63	0.77
$\geq$ 10 impressions	0.37	0.74
$\geq$ 0 impressions	0.29	0.74

DBN maintains a cross-correlation of 0.74. The performance for our method is also decreasing as the query set consists of less and less frequent queries, which is of course expected, but the magnitude of the decrease is much less, from 0.78 to 0.74.

5.2.2 Comparison Against Fully-Judged Editorial Data

Comparing the estimated performance figures against a fully editorially evaluated values is the simplest way to measure the accuracy of the evaluation method -which perhaps is the most commonly used approach in most click-based user modeling or click-based evaluation papers. Here we take one more step by also comparing how well two fully editorially evaluated ΔDCG correlate on the same dataset. As clearly seen from Table 1, the editorial relevance judgments are far from being exact, the editors show some significant disagreement. This disagreement induces a significant variance on the DCG and ΔDCG values evaluated from them, and therefore these are far from being an exact ground truth for evaluation. In this view, the correlation between two ΔDCG values evaluated independently collected relevance judgments from a random set of individuals can be regarded as an upper bound for any click-based estimation model can reach. Therefore in the following, we compare the correlation between our method and the editorially obtained values to editorial-to-editorial correlation.

To evaluate the accuracy of the overall method, we compare ΔDCG values we obtain against ΔDCG values evaluated from the fully judged validation set for five different test configurations. Table 7 presents the ΔDCG values given by the three fully judged sets and the smoothed-DBN, the bolded results show statistically significant ones with p-value less than 0.05. We also compare the results of the smoothed-DBN against two baseline methods that use the pre-existing click-estimated relevance values and the grades, individually. In these baselines, we use the assumption that all unjudged documents are bad ($p_{i1} = 1$) and all unavailable click models have 0 clickthrough rate, hence bad relevance.

We summarize the cross-correlation between delta-DCG predictions and the fully-judged editorial data in Table 8. For these 5 test configurations, the average pairwise correlation of fully judged ΔDCG results is 0.81, and the average pairwise correlation between our method and fully judged results is 0.74. In some cases the final goal is not to measure the actual value of the ΔDCG, but to choose the better of two test search ranking configurations. In this case, the actual DCG values are not as important as the sign of ΔDCG. We repeat the above analysis for the estimation of the random variable $sign(\Delta DCG)$. The average correlation between all the fully judged results is 0.73. The same value for all fully judged results and our method is 0.69. In summary, our model reaches similar/same level of correlation almost as much as two fully judged sets correlate to each other with about 65% missing judgments.

After all, the correlation with fully judged sets is the accuracy merit for the smoothed-DBN; however, considering in terms of the individual tests, we can say that both editors and smoothed-DBN agrees that the QRW model in Test 4 does not give any statistically significant difference against its predecessor, and the QRW model in Test 3 increases and the one in Test 5 decreases the overall performance. Although no full agreement, in Test 1 and Test 2, the smoothed-DBN is correlated with the overall editorial assessment as well.

6. CONCLUSIONS

We propose an inexpensive offline evaluation method that uses the pre-existing click logs and judged data collected from earlier models. Despite the fact that we use user interactions to model the relevance, we call this an offline method since no online tests are conducted, no click logs are collected for the tested configurations. Instead we only use the clicks from earlier models, and exploit the fact that the models have some shared results in common, and interpolate for the missing data using smoothing.

We use Chapelle & Zhang DBN model to obtain click-based relevance estimates. Comparisons with ordinal regression based method by Carterette & Jones show that DBN model provides similar/same accuracy for frequent queries, and it provides a much better estimates for infrequent queries. Since we only work with click data collected from earlier models and editorial data collected independently for other studies, ending up with significant amount of missing data is unavoidable. To fill in for the missing relevance judgments we provide a smoothing technique that combines position and query based smoothing depending on the available grades on any particular query. We show that this hybrid smoothing technique performs better than both query and position based smoothing, and despite the high percentage of missing judgments, the resulting method is significantly correlated with DCG values evaluated using fully judged datasets.

In this paper, we use the editorial and click data that has been collected within the last three months and assume that there are no significant time varying patterns within this time period. In general, reusability of pre-existing judgments and past session logs for evaluation is questionable since they may lead to bad results for many cases when there are time varying patterns in the data. For example, on the new data one can see many new queries (for example "iphone 4"), and radically different intents for some old queries, such as the query "iphone" meaning "iphone 4" not "iphone 3" anymore, or the query "ipad" being a typo a year ago, but not anymore. The tested configurations in this paper are QRW models (query stemming or query segmenting etc.), and the underlying characteristics of these language models do not exhibit significant time varying patterns. However in general, there should be an expiration date for the reusability of the pre-existing judgments and clicks.

Table 7: Evaluation comparison between fully judged sets and the smoothed-DBN. "+" indicates that the model detects an improvement in the Test, "-" indicates that the model detects a degradation, and "?" indicates that the model cannot detect a significant difference. ΔDCG estimates are shown in brackets.

	Test1	Test2	Test3	Test4	Test5
Fully judged set 1	+ **(0.0487)**	? (0.0159)	+ **(0.0789)**	? (0.0057)	- **(-0.1627)**
Fully judged set 2	? (-0.0290)	+ **(0.0753)**	+ **(0.1745)**	? (0.0588)	- **(-0.1108)**
Fully judged set 3	? (-0.0174)	+ **(0.0778)**	+ **(0.1152)**	? (0.1236)	- **(-0.1511)**
Click data only	? (0.0062)	? (0.0196)	? (-0.0575)	? (0.0098)	? (0.1055)
Editorial data only	? (0.0909)	+ **(0.2425)**	? (0.5711)	? (0.3579)	- **(-0.0809)**
Smoothed-DBN	? (0.0955)	+ **(0.1564)**	+ **(0.2371)**	? (0.2277)	- **(-0.0749)**

Table 8: Cross-correlation of delta-DCG predictions

Query Set	clicks-only	Judgments-only	Smoothed-DBN	Inter-annotator
$\geq$ 200 impressions	0.61	0.57	0.78	0.83
$\geq$ 50 impressions	0.52	0.55	0.77	0.81
$\geq$ 10 impressions	0.34	0.54	0.74	0.81
$\geq$ 0 impressions	0.18	0.51	0.74	0.81

7. REFERENCES

[1] P. Bailey, N. Craswell, I. Soboroff, P. Thomas, A. P. de Vries, and E. Yilmaz. Relevance assessment: are judges exchangeable and does it matter. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 667–674, 2008.

[2] C. Buckley and E. M. Voorhees. Retrieval evaluation with incomplete information. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 25–32, 2004.

[3] B. Carterette, J. Allan, and R. K. Sitaraman. Minimal test collections for retrieval evaluation. In E. N. Efthimiadis, S. T. Dumais, D. Hawking, and K. Järvelin, editors, *SIGIR*, pages 268–275, 2006.

[4] B. Carterette and R. Jones. Evaluating search engines by modeling the relationship between relevance and clicks. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 217–224. MIT Press, Cambridge, MA, 2008.

[5] O. Chapelle, D. Metlzer, Y. Zhang, and P. Grinspan. Expected reciprocal rank for graded relevance. In *CIKM '09: Proceeding of the 18th ACM conference on Information and knowledge management*, pages 621–630, 2009.

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

[7] N. Craswell, O. Zoeter, M. J. Taylor, and B. Ramsey. An experimental comparison of click position-bias models. In M. Najork, A. Z. Broder, and S. Chakrabarti, editors, *WSDM*, pages 87–94, 2008.

[8] G. E. Dupret and B. Piwowarski. A user browsing model to predict search engine click data from past observations. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 331–338, New York, NY, USA, 2008. ACM.

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

[10] T. Joachims, L. Granka, B. Pan, H. Hembrooke, and G. Gay. Accurately interpreting clickthrough data as implicit feedback. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 154–161, New York, NY, USA, 2005. ACM Press.

[11] T. Joachims and F. Radlinski. Search engines that learn from implicit feedback. *IEEE Computer*, 40(8):34–40, August 2007.

[12] A. Moffat and J. Zobel. Rank-biased precision for measurement of retrieval effectiveness. *ACM Trans. Inf. Syst.*, 27(1):1–27, 2008.

[13] B. Piwowarski, G. Dupret, and R. Jones. Mining user web search activity with layered bayesian networks or how to capture a click in its context. In *WSDM '09: Proceedings of the Second ACM International Conference on Web Search and Data Mining*, pages 162–171, New York, NY, USA, 2009. ACM.

[14] F. Radlinski and T. Joachims. Active exploration for learning rankings from clickthrough data. In *ACM SIGKDD International Conference On Knowledge Discovery and Data Mining (KDD)*, 2007.

[15] I. Soboroff. A comparison of pooled and sampled relevance judgments. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 785–786, New York, NY, USA, 2007. ACM.

[16] A. Trotman, N. Phrao, and D. Jenkinson. Can we at least agree on something? In *Proc. SIGIR Workshop on Focused Retrieval*. 2007.