Are Click-through Data Adequate for Learning Web Search Rankings?

Zhicheng Dou[1,2], Ruihua Song[1], Xiaojie Yuan[3], and Ji-Rong Wen[1]
[1]Microsoft Research Asia, No. 49 Zhichun Road, Beijing, China, 100190
[2,3]Nankai University, No. 94 Weijin Road, Tianjin, China, 300071
[1]{zhichdou,rsong,jrwen}@microsoft.com, [3]yuanxj@nankai.edu.cn

ABSTRACT

Learning-to-rank algorithms, which can automatically adapt ranking functions in web search, require a large volume of training data. A traditional way of generating training examples is to employ human experts to judge the relevance of documents. Unfortunately, it is difficult, time-consuming and costly. In this paper, we study the problem of exploiting click-through data for learning web search rankings that can be collected at much lower cost. We extract pairwise relevance preferences from a large-scale aggregated click-through dataset, compare these preferences with explicit human judgments, and use them as training examples to learn ranking functions. We find click-through data are useful and effective in learning ranking functions. A straightforward use of aggregated click-through data can outperform human judgments. We demonstrate that the strategies are only slightly affected by fraudulent clicks. We also reveal that the pairs which are very reliable, e.g., the pairs consisting of documents with large click frequency differences, are not sufficient for learning.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Search process, Relevance feedback*

General Terms

Algorithms, Experimentation, Measurement, Performance

Keywords

Click-through Data, Web Search Rankings, Learning to Rank, Implicit Feedback, Relevance Judgments

1. INTRODUCTION

Ranking function is one of the most important components of a search engine. There may be hundreds of features that can affect ranking accuracy in a search engine. It is usually difficult and even impractical to adapt a ranking function manually. For this reason, automatically learning ranking functions using machine learning techniques has been receiving much attention in recent years, and several learning-to-rank algorithms have been proposed [11, 9, 4, 5, 20, 6].

To achieve a reasonable ranking, many machine learning algorithms require a large volume of training data. A traditional way of generating training examples is to employ human annotators to explicitly judge the relevance of documents. Unfortunately, using human judgments in learning web search rankings has several disadvantages. First, it is expensive and time-consuming because it usually needs thousands of training queries to learn a reasonable ranking. Second, human judgments are given by annotators instead of real-world users. Annotators may fail to guess what real-world users are seeking for in some cases. Human judgments made under such situations may be unreliable. Third, many documents are hard to judge for a single annotator even if the annotator is an expert. Human judgments for these documents may be less reliable if subtle ratings are required. A typical labeling strategy in learning to rank is to assign a *n-grade* relevance rating to each document. To increase the reliability of human judgments, a small value of n is usually used (for example, 5). Under such settings, many subtle differences between documents (for example, the differences between the documents in same relevance levels) are excluded and this may lose much useful information.

A promising solution to this problem is to automatically extract training examples from implicit behavior of normal users [12, 18, 13, 2, 1]. In contrast to human judgments, such implicit feedback can be collected at a much lower cost, and can reflect the judgments of a large number of real-world users rather than a small number of selected annotators who may have different knowledge background and concept of relevance. In this paper, we focus on click-through data, which represent typical and simple implicit feedback logged by search engines. Click-through data contain the queries submitted by users, followed by the URLs of documents clicked by users for these queries. A common doubt about click-through data is that they may contain noise and are less reliable than human judgments. Joachims et al.[13] found that individual user clicks included bias and could not be used as absolute relevance judgments directly. They have made great efforts on extracting reliable relevance preferences from individual queries [12, 13] and query chains [18, 13].

In this paper, we use click-through in another way. We

CIKM'08, October 26–30, 2008, Napa Valley, California, USA.

find that although some individual user clicks are unreliable, the aggregation of a large number of user clicks provides a valuable indicator of relevance preference. We do not try to extract reliable relevance preferences from individual query sessions. Instead, we aggregate large numbers of user clicks for each query-document pair, and extract pairwise preferences based on the aggregated click frequencies of documents. Simply stated, given a query, a pairwise training example is generated if one document receives more clicks than the other. We evaluate our methods using a large-scale dataset comprised of 12,000 human labeled queries and 46-day click-through logs.

Different from the common perspective that implicit click-through data are less reliable, our experimental results show that click-through data are very effective for learning. A straightforward use of aggregated click frequencies can achieve reasonable rankings, and can be even better than using human judgments. We also reveal that click-through data can be more reliable and informative than human judgments since they include decisions of large numbers of real-world users. We also show that the approach of using click-through data for learning is not very sensitive to fraudulent clicks. Another interesting thing we found in this paper is that reliable pairs, e.g., the pairs consisted of documents with larger click frequency difference, are not sufficient for learning.

2. RELATED WORK

In recent years, learning to rank has been receiving much attention in the information retrieval area. Several learning-to-rank algorithms that use relative pairwise preferences have been proposed and applied [11, 9, 4, 5, 20, 6]. For example, Joachims et al.[11] applied the Ranking SVM algorithm based on linear SVM to information retrieval. Cao et al.[6] adapted the Ranking SVM to document retrieval by modifying the Loss function. Burges et al.[4] developed the RankNet algorithm, which used neural networks for learning the ranking functions. They then developed the LambdaRank [5] for speeding up the RankNet training. Freund et al.[9] proposed the RankBoost algorithm, which uses ideas of Adaboost for learning ranking functions. Zheng et al.[20] proposed a ranking algorithm Gbrank based on the regression method of using gradient boosting trees. In this paper, we utilize the RankNet algorithm, and the details of the algorithm are beyond the scope of this paper.

This paper focuses on the problem of extracting training data for these learning-to-rank algorithms from implicit user feedback. There has been much work on analyzing the relationship between implicit feedback and user interests. A comprehensive overview of studies of implicit measures can be found in [14]. In this paper, we focus on web search scenarios. Joachims et al.[12, 18, 13] analyzed users' decision processes in web search using eye-tracking and compared implicit feedback against manual relevance judgments. They found that user behavior did depend on the quality of the presented ranking and click-through data contained much useful information that could be used for improving ranking functions. They also found that click-through data contained much noise and bias. Users' decisions were affected by "trust bias" and "quality of context bias," and user clicks could not be used as absolute relevance judgments directly. For this reason, they made great efforts on extraction of reliable relative preference [11, 17, 19, 13]. They explored and evaluated several strategies to automatically generate relative relevance judgments for learning retrieval function from individual user queries or query chains [12, 18, 13]. A typical approach was to assume that a clicked result was more relevant than an unclicked higher-ranked result (click > non-click above). This strategy compensates for presentation bias by considering that users scan results from top to bottom. However, as pointed out by Radlinski and Joachims [19], preferences extracted by this strategy always oppose the presented ordering. In particular, such relevance judgments are all satisfied if the ranking is reversed, making the preferences difficult to use as training data, especially when the ranking is already reasonable. Furthermore, since preferences are extracted from an individual query or a query chain, they may be still noisy and biased due to user needs diversity and query ambiguity. Radlinski and Joachims [19] introduced the FairPairs method to modify the presentation of search results to collect more reliable preferences. Because the modification of search results may affect user satisfcation (especially when the first and second result are swapped), this method cannot be easily adopted in search engines. Different from above work, we do not extract preferences from individual queries and query chains. We aggregate large numbers of user clicks for each query-document pair, and extract pairwise preferences based on the aggregated click frequencies of documents.

Fox et al.[8] also explored the relationship between implicit and explicit measures in web search. They developed Bayesian models to correlate implicit measures and explicit relevance judgments. Similar to Joachims et al.'s work [13], their research was based on individual queries and search sessions. The models were used to predict user satisfaction but not to learn rankings. Furthermore, they considered a wide range of user behaviors (e.g., dwelling time, scrolling time, reformulation patterns) in addition to click-through behavior. In this paper, we only focus on click-through data.

Part of our work is close to the research done by Agichtein, Brill, and Dumais [2]. They found that the preferences extracted by the strategies proposed by Joachims et al.[13] were still noisy. They proposed to use aggregated click frequencies to filter out noisy clicks. Similar to our work, they compared the preferences extracted from click-through data with the preferences derived from human relevance judgments. In this paper, we further evaluate the effectiveness of these preferences by using them to learn ranking functions. In [1], Agichtein, Brill, and Dumais explored several approaches that incorporated implicit feedback features directly into the trained ranking function. Experimental results showed that using these additional user behavior features could improve web search performance. Since implicit user feedback was used as features, these approaches still need human judgments. In this paper, training examples can be extracted directly from click-through data and thus no human labeled training data are needed.

Zheng et al.[20] also used preferences extracted from click-through data for learning web search rankings. They used the likelihood ratio test (LRT) to extract significant pairs and applied the strategies proposed by Joachims et al.[13] to extract preferences among the significant pairs. They extracted only 20,948 preferences, a finding much smaller than ours, and they did not compare their effectiveness with human judgments. In this paper, we compared the learning ability of human judgments and click-through data using a large number of queries.

3. METHODOLOGY

Several learning-to-rank algorithms, including the RankNet [4] and the Gbrank[20], can accept explicit pairwise training examples. Given a query q, assume that d_i and d_j are two returned documents. $rating(q, d_i)$ and $rating(q, d_j)$ are corresponding relevance judgments made by human experts. Previous work [4, 20] usually uses the following strategy to extract pairwise preferences from explicit n-grade human judgments:

STRATEGY 1. **(Label)** *If document d_i is given a higher relevance rating than d_j, i.e., $rating(q, d_i) > rating(q, d_j)$, then a relevance preference $rel(q, d_i) >_{lbl} rel(q, d_j)$ is extracted for learning.*

The main work of this paper is to extract training examples from implicit user feedback. As we described in Section 2, Joachims et al.[13] have found that implicit feedback can be used as relative relevance judgments. In this paper, we also interpret click-through data as relative relevance judgments. Different from previous approaches, we do not try to extract reliable preferences from individual queries. We aggregate large numbers of user clicks for each query-document pair, and extract pairwise preferences based on the aggregated click frequencies of documents. Assume that $click(q, d_i)$ and $click(q, d_j)$ are corresponding aggregated click frequencies of documents d_i and d_j. We propose to use the following strategy to generate relevance preferences from click-through data:

STRATEGY 2. **(CT)** *Let $cdif(q, d_i, d_j) = click(q, d_i) - click(q, d_j)$. $cdif(q, d_i, d_j)$ is click frequency difference of two documents d_i and d_j for query q. If $cdif(q, d_i, d_j) > 0$, i.e., document d_i is clicked more often than document d_j, a relevance preference example $rel(q, d_i) >_{ct} rel(q, d_j)$ is extracted for learning.*

CT strategy is proposed based upon a simple notion: if a document is favored by more users, it will be more relevant. Certainly, preferences generated by this strategy may include bias. The position of a result document actually influences users' decisions and a highly-ranked result is likely to be clicked more frequently than a result ranked lower. We also don't do any normalization or smoothing on the aggregated click frequency, but as we will introduce in Section 6, using the preferences generated by such a straightforward strategy can achieve accurate rankings already. We will investigate new pair extraction strategies to extract preferences with higher reliability and coverage in future work.

To evaluate reliability and effectiveness of click-through data, we conduct several experiments to compare them with human judgments. These experiments are divided into two phases. In the first phase, we analyze the correlation between human judgments and click-through data. We analyze whether the preferences extracted from click-through are concordant with those generated based upon human judgments. Several previous works [13] have used the same way to analyze the reliability of click-through data. We think that this is not enough for evaluating the learning effectiveness of click-through data. In the second phase, we further use these preferences as training data to learn ranking functions and evaluate the accuracy of generated rankings. If the rankings are as good as or even better than those generated based upon human judgments, we can conclude that these preferences are effective for learning web search rankings.

Table 1: Basic statistics of dataset

	Training	Validation	Test
#Queries	10,000	1,000	1,000
#Documents	584,322	325,514	324,782
#Judged Documents	313,316	28,820	29,788
#Clicked Documents	71,170	6,301	6,880

4. DATA COLLECTION

We use a dataset from a commercial search engine which comprises 12,000 randomly sampled queries and a certain number of returned documents for the English/U.S. market. On average, about 30 documents per query are manually judged by human experts. A five-grade (0 to 4, bad match to perfect match) rating is assigned for each judged document. Unlabeled documents are given the rating 0.

We collect 46-day click-through logs from the search engine for these queries (from July 9, 2006 to August 23, 2006). Clicks for these queries are aggregated and a click frequency is generated for each query-document pair. Click frequency 0 is assigned for the documents that are not clicked by users.

For each query-document pair, hundreds of features are generated (most of them are similar to the features in the TREC collection in LETOR [16]). These features will be used for learning-to-rank documents in Section 6.

In brief, each record of the dataset is mainly comprised of a query identity, a document identity, a human rating, a click frequency, and hundreds of numeric features. The records are then shuffled and split into three subsets: 10,000 queries for training, 1,000 queries for validation, and 1,000 queries for testing. Table 1 summarizes statistics of the dataset. Please note that for validation/test queries, more documents are used.

5. CORRELATION BETWEEN CLICKTHROUGH AND HUMAN JUDGMENTS

In this section, we will analyze the correlation between click-through data and human judgments. The Kendall tau coefficient [15] is usually used to measure the degree of correlation between two rankings. In this paper, we select to use the Kendall tau-b (τ_b) because it uses a correction for ties. τ_b is a nonparametric measure of association based on the number of concordances and discordances in paired observations. Given a query q and two returned documents d_i and d_j:

- d_i and d_j are concordant if $rating(q, d_i) > rating(q, d_j)$ and $click(q, d_i) > click(q, d_j)$, or if $rating(q, d_i) < rating(q, d_j)$ and $click(q, d_i) < click(q, d_j)$.
- d_i and d_j are discordant if $rating(q, d_i) > rating(q, d_j)$ and $click(q, d_i) < click(q, d_j)$, or if $rating(q, d_i) < rating(q, d_j)$ and $click(q, d_i) > click(q, d_j)$.
- d_i and d_j are tied if $rating(q, d_i) = rating(q, d_j)$ and/or $click(q, d_i) = click(q, d_j)$.

The total number of pairs that can be constructed for a query with n documents is $N = n(n-1)/2$. N can be decomposed into these five quantities:

$$N = P + Q + X_0 + Y_0 + (XY)_0$$

P is the number of concordant pairs, Q is the number of discordant pairs, X_0 is the number of pairs tied only on the

Table 2: Overall correlation between click-through data and human judgments (Kendall tau-b)

	Training	Validation	Test
BothClicked	0.201274	0.163600	0.194758
AtLeastOneClicked	0.345716	0.300375	0.363094

human judgments, Y_0 is the number of pairs tied only on the click frequencies, and $(XY)_0$ is the number of pairs tied on both human judgments and click frequencies.

The τ_b is calculated by the following formula:

$$\tau_b = \frac{P - Q}{\sqrt{(P + Q + X_0)(P + Q + Y_0)}} \quad (1)$$

The τ_b has the range $-1 \leq \tau_b \leq +1$, with -1 standing for 100% negative association, or perfect inversion, and +1 standing for 100% positive association, or perfect agreement. A value of zero indicates the absence of association.

Average Kendall tau-b for a dataset is calculated by averaging the Kendall tau-b values for all queries in the dataset.

5.1 Overall Correlation

In this section, we analyze the correlation between the preferences extracted based upon human judgments and those extracted based upon click frequencies. Please note that unjudged documents are skipped in the experiments. Table 2 shows experimental results. The first row in this table shows that Kendall tau-b values computed based upon the pairs in which both documents are clicked, and the second row shows Kendall tau-b values computed based upon the pairs in which at least one document is clicked (the other one is either clicked or unclicked). The click frequency of unclicked document is set to 0. Please note that the pairs in which both documents are not clicked are skipped to avoid too many ties. Experimental results show that human judgments and click frequencies are only weakly correlated. The Kendall tau-b correlation coefficient on training set is only about 0.20 when considering only clicked documents. It reaches about 0.35 when unclicked documents are included.

After comparing the two rows of Table 2, we find click-through and human judgments correlate better when unclicked documents are included. By observing the data, we find most unclicked documents have lower ratings than clicked documents. When using unclicked documents, more accordant preferences are generated generally.

5.2 Click Frequency Differences

In the previous section, a preference is extracted simply when two documents are clicked with different frequencies, even if a document is clicked only one more time than the other. We think that the preference may be more reliable when click frequency difference is larger. We propose to use the following strategy to extract only pairs with click frequency being greater than n:

STRATEGY 3. **(CT_Gn)** *A relevance preference example $rel(q, d_i) >_{ct} rel(q, d_j)$ is extracted only if click frequency difference of d_i and d_j is greater than n, i.e., $cdif(q, d_i, d_j) > n$.*

We plot correlation coefficients between preferences generated by Label and CT_Gn with different settings of n on training set in Figure 1. Figure 1 shows that correlation ratio increases along with increase of n. It means that preferences with larger click frequency difference correlate more to human judgments indeed. For example, when selecting only pairs with click frequency difference > 500, the Kendall tau-b correlation coefficient is about 0.7. Please note correlation ratio for CT_G0 is larger than the overall correlation given in Table 2. This is because the overall correlation counts in document pairs in which click frequencies of two documents are equal and human ratings are different (part $Y0$ in Equation (1)), while CT_Go excludes these pairs.

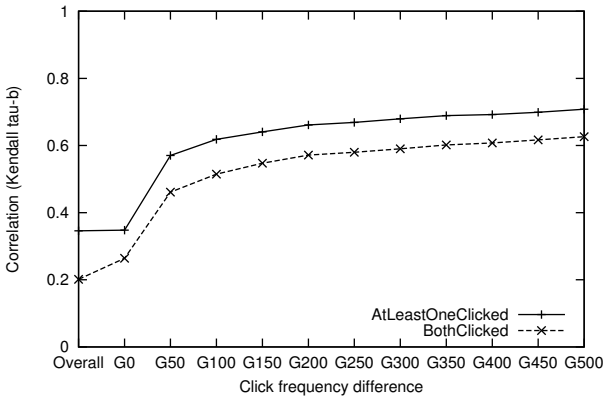


Figure 1: Correlation between human ratings and click-through pairs with click frequency difference $> n$ (CT_Gn) on training set.

5.3 Click Entropy

In this section, we propose to use the click entropy [7] to classify queries, and analyze the correlation between click-through data and human judgments for queries with different click entropies. The click entropy can be used as a simple evidence to identify click diversity. Click entropy of a query q is defined as Equation 2:

$$ClickEntropy(q) = \sum_{d \in \mathcal{D}(q)} -P(d|q) \log_2 P(d|q) \quad (2)$$

Here $ClickEntropy(q)$ is the click entropy of query q. $\mathcal{D}(q)$ is collection of documents clicked for query q. $P(d|q)$ is the percentage of clicks on document d among all clicks on q.

A query with a small click entropy value is more likely to be a navigational [3] or clear query because users consistently click a few results for this query. Examples of such queries include "yahoo," "myspace," and "youtube." A query with a larger click entropy value is more likely to be an informational [3] or ambiguous query. Users tend to select multiple documents to fulfill their information needs and different users may have their own preferences for these queries. The queries "photos," "information retrieval," and "jobs" are with large click entropies.

To make the click entropy more reliable, we eliminate the queries with total click times < 25. Figure 2 shows the correlation between the human ratings and the click frequencies for the left training queries with varied click entropies. We proportionally divide the left 7,646 training queries into four bins by their click entropies. Each bin contains about 1,911 queries. Bin-1 contains the queries with smallest click entropies (0 to 1.71, average 1.03), and bin-4 contains the

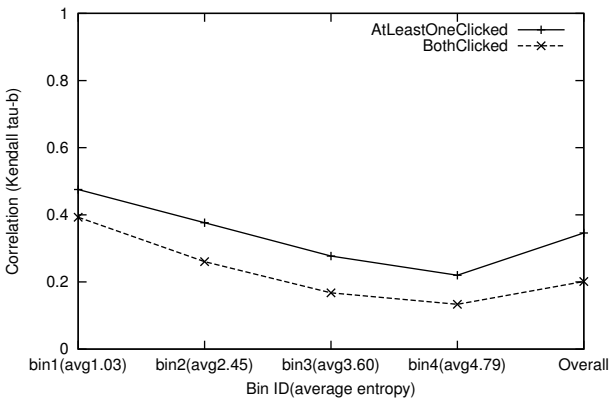


Figure 2: Correlation between human judgments and click-through data with varied click entropies.

queries with largest click entropies (4.07 to 9.88, average 4.79).

Figure 2 shows that the pairs contained in the queries with small click entropies correlate more to human judgments. We give the following reasons. First, for navigational and/or clear queries, there are usually only one or a few perfect documents. Users can easily judge which document is the one they want to find. Second, compared with ambiguous queries, current search engines can rank results better on these queries. Clicks based upon such ranking are relatively more reliable. Third, since queries are less ambiguous, user clicks are more consistent. It is also easier for annotators to judge the relevance of documents. All these factors cause aggregated user clicks for these queries to correlate more with human judgments. In contrast to this, informational and/or ambiguous queries are more difficult to judge, both for real-world users and annotators. Furthermore, different users may have different preferences on these queries. Annotators may find it difficult to understand real-world information needs on these queries and their judgments may be biased and incorrect.

6. USING CLICK-THROUGH DATA TO LEARN WEB SEARCH RANKINGS

In this section, we use the preferences extracted by previous strategies as training examples to learn ranking functions. We employ the RankNet, a neural net tuning algorithm which optimizes feature weights to best match explicitly provided pairwise preferences, to learn ranking functions. Rank-Net has demonstrated excellent performance in learning to rank. Specific training algorithms used by the RankNet are beyond the scope of this paper. Detailed description can be found in [4].

We use a 2-layer implementation of the RankNet in our experiments. For each experiment, RankNet is trained for 100 rounds. The best model is selected by testing all 100 models on the 1,000-query validation set, and is then used to be tested on the test set.

6.1 Ranking Accuracy Evaluation Metrics

We use two different ranking evaluation methods in this paper: NDCG@K based upon human ratings and Kendall tau-b based upon click frequencies.

6.1.1 NDCG@K based upon human ratings

We first evaluate ranking accuracy by using a normalized discounted cumulative gain measure (NDCG) [10] based upon human judgments on test documents. For a given query q, the NDCG@K is computed as:

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

M_q is a normalization constant calculated so that a perfect ordering would obtain NDCG of 1; and each $r(j)$ is a human rating of the result returned at position j.

NDCG is well suited to web search evaluation, as it rewards relevant documents in the top-ranked results more heavily than those ranked lower. We report NDCG@5, i.e., $K = 5$. We also experiment with other settings of K, and they show similar results with $K = 5$.

6.1.2 Kendall tau-b based upon click frequencies

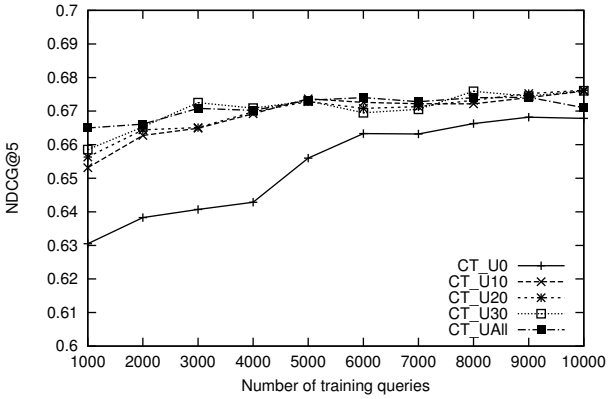
We also use click-through data to evaluate ranking accuracy. For each test query, we sort the documents by their click frequencies and get a rank list. This rank list is purely based upon click frequencies so we call it click rank list. Please note that the documents with same click frequencies are given a same rank in the click rank list. We then use Kendall tau-b introduced in Section 5 to measure the correlation between the click rank list and the rank list generated by the ranking function being evaluated.

6.2 Overall Performance

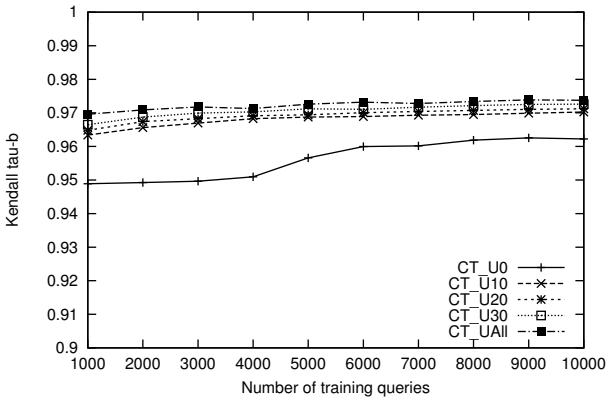
In this section, we investigate the performance of CT strategy. Figure 3 shows the RankNet performance of CT when using different numbers of unclicked documents. In this figure, series CT_Un means using maximum n unclicked documents, series CT_UAll means using all unclicked documents, and CT_U0 means using only clicked documents. The click frequency of each unclicked document is set to 0. This figure shows that using some unclicked documents can help improve learning performance. In Section 5.1, we found that pairs including unclicked documents correlate more to human ratings. Including these pairs can improve the accuracy of training examples and thus can help achieve better rankings. In the remaining part of this paper, we will use all unlabeled documents for CT strategy.

For Label strategy, Burges et al.[4] have found that using some unlabeled documents can improve ranking performance. We experiment with 10, 20, 30, 40, and all unlabeled documents as extra examples of low relevance documents, and we find that using 20 unlabeled documents gets approximately optimal results. In the remaining part of this paper, we will use 20 unlabeled documents for Label strategy.

To compare CT and Label, we experiment with different sizes of training queries (from 100 to 10,000) and report experimental results in Figure 4. This figure shows that CT strategy outperforms Label strategy with varied sizes of training set, both when being evaluated by human judgments with NDCG metric, and when being evaluated by click frequencies with Kendall tau-b metric. Please note that all the improvements are significant ($p<0.05$). One consideration is that there may be different numbers of preferences for these two strategies. After counting the pairs generated by each strategy, we find Label generates much more preferences/document pairs (about twice many) than CT. For ex-

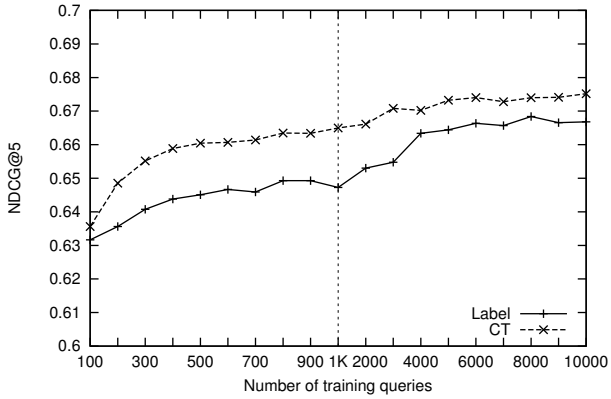


(a) NDCG@5

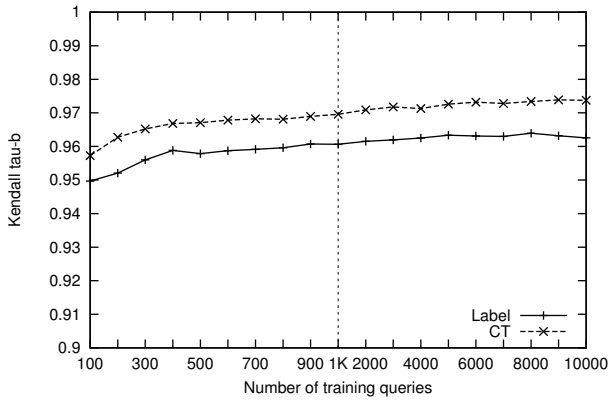


(b) Kendall tau-b

Figure 3: RankNet performance when using different amounts of unclicked documents. CT_Un means using up to n unclicked documents per query.



(a) NDCG@5



(b) Kendall tau-b

Figure 4: RankNet performance comparisions of strategies Label and CT

ample, Label generates about 9.5 millions preferences from all 10,000 training queries, while CT generates only about 5 millions. The reason is that there are about 30 documents judged by human experts, usually more than the number of documents clicked by users. This result tells us that CT strategy outperforms Label strategy, even if it generates less training preferences than Label. This means that the preferences generated from click-through data are more effective than those extracted from human ratings.

In brief, the above experimental results demonstrate that click-through data are very useful and effective in learning ranking functions. A straightforward exploitation of same amount of aggregated click-through data can achieve better rankings than using human judgments. This conclusion deviates from some original notions that click-through cannot be directly used for learning to rank. This is a promising start point of using click-through data to learn Web search rankings. Since click-through data can be collected with much lower cost, we can utilize more click-through data. We will investigate whether higher ranking accuracy will be achieved if more click-through data are used in future work.

6.3 Click Frequency Difference

Section 5.2 showed that pairs with larger click frequency differences correlate more to human judgments. Are these pairs more useful and more effective in learning ranking functions? We will investigate this problem in this section.

We use preferences extracted from all 10,000 training queries by CT_Gn strategy to train the RankNet. Figure 5 reports ranking accuracy when n is from 0 to 500 with an increase step 50. Experimental results show that the larger the n is, the worse the performance is. By counting the numbers of pairs for these strategies, we find the number of pairs decreases when click frequency increases, which may cause a decrease in learning performance. For example, CT_G0 generates 4,890,820 pairs, while CT_G500 contains only 252,753.

To reduce the impact of amount of training data and investigate how click frequency difference could affect learning performance with same amount of training data, we propose to generate equivalent numbers of pairs with varied click frequency difference ranges. Specifically, we use the pairs with click frequency difference between 10 and 25 (CT_10to25), between 26 and 99 (CT_26to99), and greater or equal than 100 (CT_GE100) to train the RankNet separately. Figure 6 shows that these three strategies generate equivalent numbers of pairs.

We analyze the correlations between pairs generated by these strategies and pairs generated based upon human judgments. We show experimental results in Table 3. Please note that we use all unclicked documents in these experiments.

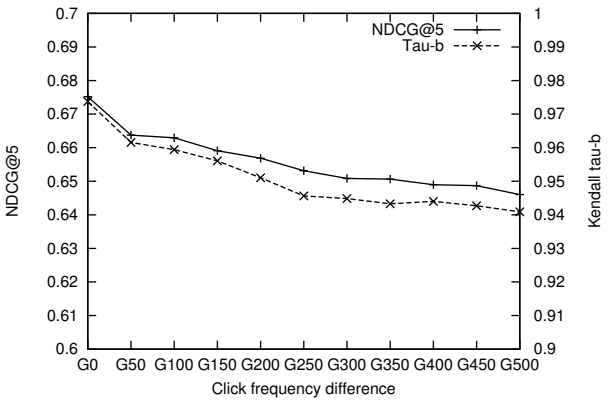


Figure 5: RankNet performance when using pairs with variant click frequency differences

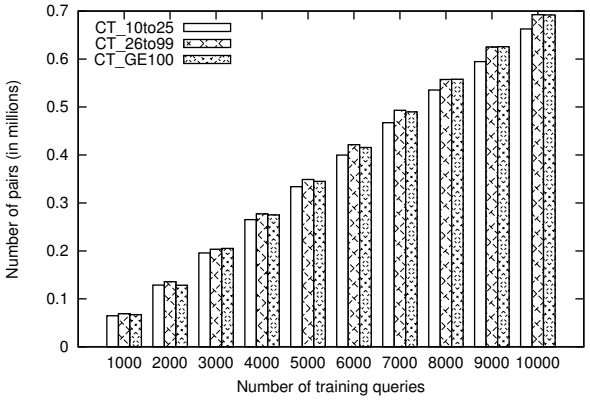


Figure 6: Numbers of pairs generated by three different pair selection strategies

Table 3: Correlation between human judgments and click-through data under three different pair selection strategies

	Training	Validation	Test
CT_10To25	0.305743	0.270589	0.306940
CT_26To99	0.390308	0.361930	0.337831
CT_GE100	0.617736	0.628011	0.605718

Table 3 also demonstrates that correlation between these two types of pairs is stronger when click frequency difference is larger.

Figure 7 shows the RankNet performance when using the three pair generation strategies. The results indicate that when being trained using pairs with larger click frequency differences, the RankNet does not generate better rankings. CT_GE100 performs worst though it has highest correlation with human judgments. Here we try to give a possible reason. Although the preferences with larger click frequency are more reliable, they are obvious and simple. They lack enough information for learning more accurate weights of features. The learner will be biased if using only such training examples and fails to rank documents in complex cases (for example when relevance difference between two documents is not very significant). On the contrary, the pairs in which the two documents are slightly different may contain much subtle and diverse information which are more important for generating a stable learner.

6.4 Click Entropy

We train the RankNet using the preferences generated from training queries with variant click entropies. Please note that all validation and test queries are used in validation and test phase. We plot the NDCG values in Figure 8. We plot the results of using all training queries and denote this approach with "All".

Let's have a look at Label strategy first. We find when using human label data, the RankNet performs better when using queries in bin-2 and bin-3 than using queries in bin-1 and bin-4. For the queries with smallest click entropies (in bin-1), pairs contained in these queries are obviously simple on average. These simple training examples are less useful for learning a robust ranking function. The ranking function learned based upon only these pairs may be highly biased. For the queries in bin-4, it may be because human judgments for these queries are less reliable than queries with small entropies. It is usually hard for an annotator or even an expert to precisely decide which document is more relevant than another for these queries. Rankings generated base upon such training examples are generally less accurate. Compared with bin-1 and bin-4, since bin-2 and bin-3 include both "easy" and "difficult" training examples, they can achieve better ranking accuracy. Certainly, they still performace worse than using all training queries.

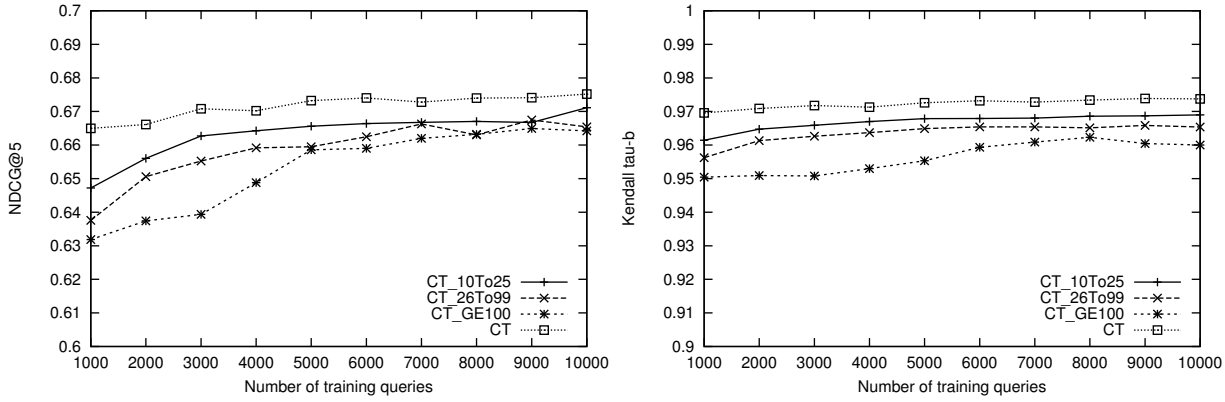
Compared with Label, we find CT strategy is more robust. CT outperforms Label especially for queries in bin-1 and bin-4. Because click-through data include the decisions of large numbers of real-world users, they can be more reliable than human judgments made by a small number of annotators. Furthermore, since click-through data aggregate diverse user selections, they can be more diverse and informative than human judgments. Many subtle differences between documents are included in miscellaneous user clicks and these are very useful for learning.

6.5 Stability of Click-through-based Strategies

A potential concern of using click-through data for learning is that it may be easily affected by fraudulent clicks. To evaluate the stability of our proposed strategies, we manually generate some fraudulent clicks in training data and analyze how ranking accuracy is affected.

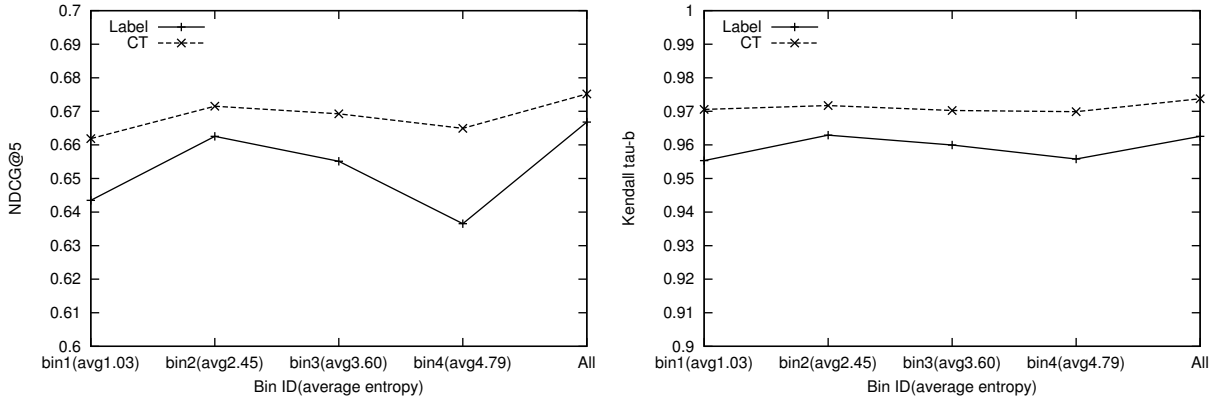
We use all 10,000 training queries in this experiment. We randomly "spam" several queries by generating some fraudulent clicks for them. For each query to be spammed, we randomly select one or more documents without clicks, and update their click frequencies to 100,000,000 which is larger than all real click frequencies. The spammed documents become the "best" documents for this query in click-through data. We experiment with spamming 1 to 5 documents per query. Note that there are only 7 clicked documents per query in training set on average and there may be some spammed documents already.

Figure 9 demonstrates that ranking accuracy decreases indeed when more documents are spammed, but the decrease is within a small range. When only a small number of documents are spammed per query, ranking accuracy is only slightly affected even if a large number of queries are spammed. For example, when one document is spammed



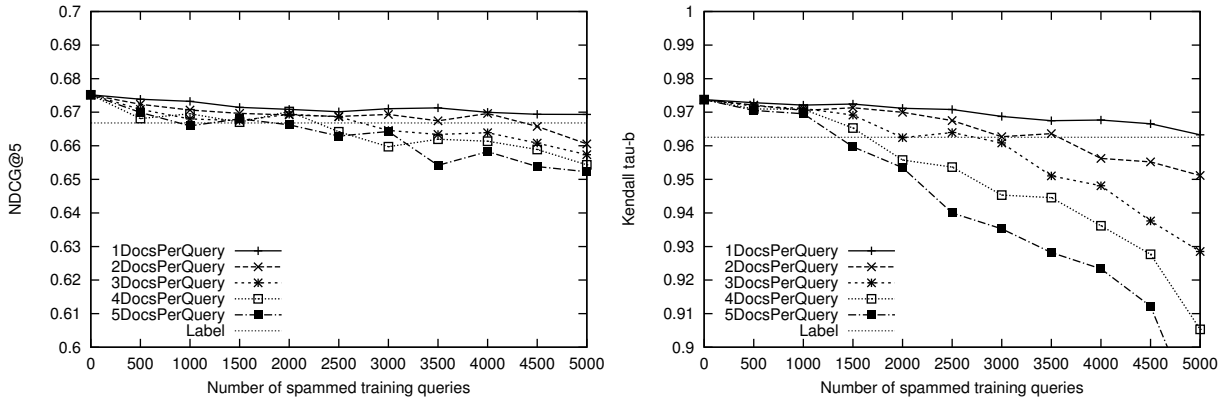
(a) NDCG@5 (b) Kendall tau-b

Figure 7: RankNet performance of three different pair selection strategies



(a) NDCG@5 (b) Kendall tau-b

Figure 8: RankNet performance when using queries with different click entropy ranges



(a) NDCG@5 (b) Kendall tau-b

Figure 9: RankNet performance when variant numbers of queries are spammed.

per query and half of training queries are spammed (5,000 queries), ranking accuracy only reduces 0.005 and is still better than that achieved by Label strategy. If a large number of documents (for example, 5 documents) are spammed per query and only a small number of queries are spammed (for example, less than 1,500 queries), ranking accuracy is still comparable with that achieved by Label strategy. This means that click spam has only limited impact on our clickthrough-based methods.

In fact, since real-world queries are abundant and we only use a small portion of them, it is really hard for spammers to spam such large percents of queries and documents as we experimented above. For this reason, our click-through-based methods are stable and applicable in real-world use.

7. CONCLUSIONS

In this paper, we studied the problem of using aggregated click-through log to learn web search rankings. We used a large-scale dataset comprised of 12,000 human labeled queries and 46-day click-through logs. We showed that click-through data weakly correlate to human judgments on average. Despite this, we further revealed that click-through data are useful and effective in learning web search rankings. A straightforward use of aggregated click-through data can achieve a better ranking than using human judgments. We revealed that click-through data have inimitable advantages to human judgments. First, they can be collected with much lower cost, and their amount is virtually unlimited. Second, they contain decisions of large numbers of real-world users, so they can be more reliable than human judgments made by a small number of annotators. For example, for informational or ambiguous queries, click-through data are more reliable than explicit human judgments. Third, compared with limited numbers of relevance levels in human ratings, click-through data contain many subtle differences between documents which are very useful for learning an accurate and stable ranking. We also demonstrated that our strategies are stable and are only slightly affected by fraudulent clicks. These results tell us that using click-through data in learning web search rankings are applicable.

Another interesting thing we revealed in this paper is that the pairs which are very reliable, e.g., the pairs consist of documents with large click frequency differences or the pairs contained in queries with small click entropies, are not sufficient for learning. The rankings generated based upon only such pairs may be biased. This tells us that we should not overemphasize reliability and ignore the coverage of training examples. This is an important conclusion we should consider when adapting click-through data for learning.

8. REFERENCES

[1] E. Agichtein, E. Brill, and S. Dumais. Improving web search ranking by incorporating user behavior information. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 19–26, New York, NY, USA, 2006. ACM Press.

[2] E. Agichtein, E. Brill, S. Dumais, and R. Ragno. Learning user interaction models for predicting web search result preferences. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 3–10, New York, NY, USA, 2006. ACM Press.

[3] A. Broder. A taxonomy of web search. *SIGIR Forum*, 36(2):3–10, 2002.

[4] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender. Learning to rank using gradient descent. In *ICML '05: Proceedings of the 22nd international conference on Machine learning*, pages 89–96, New York, NY, USA, 2005. ACM Press.

[5] C. J. C. Burges, R. Ragno, and Q. V. Le. Learning to rank with nonsmooth cost functions. In *Advances in Neural Information Processing Systems 18*, pages 395–402, Cambridge, MA, 2006. MIT Press.

[6] Y. Cao, J. Xu, T.-Y. Liu, H. Li, Y. Huang, and H.-W. Hon. Adapting ranking svm to document retrieval. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 186–193, New York, NY, USA, 2006. ACM Press.

[7] Z. Dou, R. Song, and J.-R. Wen. A large-scale evaluation and analysis of personalized search strategies. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 581–590, New York, NY, USA, 2007. ACM Press.

[8] S. Fox, K. Karnawat, M. Mydland, S. Dumais, and T. White. Evaluating implicit measures to improve web search. *ACM Trans. Inf. Syst.*, 23(2):147–168, 2005.

[9] Y. Freund, R. Iyer, R. E. Schapire, and Y. Singer. An efficient boosting algorithm for combining preferences. *J. Mach. Learn. Res.*, 4:933–969, 2003.

[10] K. Järvelin and J. Kekäläinen. Ir evaluation methods for retrieving highly relevant documents. In *SIGIR '00: Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 41–48, New York, NY, USA, 2000. ACM Press.

[11] T. Joachims. Optimizing search engines using clickthrough data. In *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 133–142, New York, NY, USA, 2002. ACM Press.

[12] 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.

[13] T. Joachims, L. Granka, B. Pan, H. Hembrooke, F. Radlinski, and G. Gay. Evaluating the accuracy of implicit feedback from clicks and query reformulations in web search. *ACM Trans. Inf. Syst.*, 25(2):7, 2007.

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

[15] M. Kendall and B. B. Smith. Randomness and random sampling numbers. *Journal of the Royal Statistical Society*, 101(1):147–166, 1938.

[16] T.-Y. Liu, T. Qin, J. Xu, W. Xiong, and H. Li. Letor: Benchmark dataset for research on learning to rank for information retrieval. In *LR4IR 2007 in conjunction with SIGIR 2007*, 2007.

[17] F. Radlinski and T. Joachims. Evaluating the robustness of learning from implicit feedback. In *Proceedings of the 22nd ICML Workshop on Learning in Web Search*, 2005.

[18] F. Radlinski and T. Joachims. Query chains: learning to rank from implicit feedback. In *KDD '05: Proceeding of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 239–248, New York, NY, USA, 2005. ACM Press.

[19] F. Radlinski and T. Joachims. Minimally invasive randomization for collecting unbiased preferences from clickthrough logs. In *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI)*, 2006.

[20] Z. Zheng, K. Chen, G. Sun, and H. Zha. A regression framework for learning ranking functions using relative relevance judgments. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 287–294, New York, NY, USA, 2007. ACM Press.