Evaluating Retrieval Performance using Clickthrough Data

Thorsten Joachims
Cornell University
Department of Computer Science
Ithaca, NY 14853 USA
tj@cs.cornell.edu

Abstract

This paper proposes a new method for evaluating the quality of retrieval functions. Unlike traditional methods that require relevance judgments by experts or explicit user feedback, it is based entirely on clickthrough data. This is a key advantage, since clickthrough data can be collected at very low cost and without overhead for the user. Taking an approach from experiment design, the paper proposes an experiment setup that generates unbiased feedback about the relative quality of two search results without explicit user feedback. A theoretical analysis shows that the method gives the same results as evaluation with traditional relevance judgments under mild assumptions. An empirical analysis verifies that the assumptions are indeed justified and that the new method leads to conclusive results in a WWW retrieval study.

1 Introduction

User feedback can provide powerful information for analyzing and optimizing the performance of information retrieval systems. Unfortunately, experience shows that users are only rarely willing to give explicit feedback (e. g. [10]). To overcome this problem, this paper explores an approach to extracting information from implicit feedback. The user is not required to answer questions, but the system observes the user's behavior and infers implicit preference information automatically.

The particular retrieval setting studied in this paper is web search engines. In this setting, it seems out of question to ask users for relevance judgments about the documents returned. However, it is easy to observe the links the user clicked on. With search engines that receive millions of queries per day, the available quantity of such clickthrough data is virtually unlimited. This paper shows how it is possible to tap this information source to compare different search engines according to their effectiveness. The approach is based on the idea of designing a series of experiments (i.e. blind tests) for which clickthrough data provides an unbiased assessment under plausible assumptions.

2 Previous Work

Most evaluation in information retrieval is based on precision and recall using manual relevance judgments by experts [1]. However, especially for large and dynamic document collections, it becomes intractable to get accurate recall estimates, since they require relevance judgments for the full document collection. To some extend, focused sampling like in the pooling method [11] as used in TREC [21] can reduce assessment cost. The idea is to focus manual assessment on the top documents from several retrieval systems, since those are likely to contain most relevant documents. While some attempts have been made to evaluate retrieval functions without any human judgments using only statistics about the document collection itself [20][8][14], such evaluation schemes can only give approximate solutions and may fail to capture the users' preferences.

Retrieval systems for the WWW are typically not evaluated using recall. Instead, only their precision at N is measured [12][7]. This does not only decrease the amount of manual relevance assessment, but also – like the method presented in this paper – focuses the evaluation on those documents actually observed by the user [19]. However, the need for manual relevance judgments by experts still limits the scale and the frequency of evaluations.

The usefulness measure of Frei and Schäuble [5] uses a different form of human relevance assessment. With respect to being a relative performance criterion, it is similar to the method proposed in this paper. The usefulness measure is designed to compare two retrieval strategies without absolute relevance judgments. Referring to empirical studies [17][13], Frei and Schäuble argue that humans are more consistent at giving relative relevance statements. Furthermore, they recognize that relevance assessments are user and context dependent, so that relevance judgments by experts are not necessarily a good standard to compare against. Therefore, their method relies on relative preference statements from users. Given two sets of retrieved documents for the same query, the user is asked to judge the relative usefulness for all/some pairs of documents. These user preferences are then compared against the orderings imposed by the two retrieval functions and the respective number of violation is used as a score. While this technique eliminates the need for relevance judgments on the whole document collection, it still relies on manual relevance feedback from the user.

Some attempts have been made towards using implicit feedback by observing clicking behavior. For example, the search engine DirectHit uses clickthrough as a measure of popularity. Other search engines appear to record clickthrough, but do not state what use they make of it. Published results on using clickthrough data exist for experimental retrieval systems [3] and browsing assistants [15]. However, the semantics of such data is unclear as argued in the following.

	f used for evaluation		
↓ f used for presentation	bxx	tfc	hand-tuned
bxx	**6.26**± 1.14	46.94±9.80	28.87± 7.39
tfc	54.02±10.63	**6.18**±1.33	13.76± 3.33
hand-tuned	48.52± 6.32	24.61±4.65	**6.04**± 0.92

Table 1: Average clickrank for three retrieval functions ("bxx", "tfc" [16] , and a "hand-tuned" strategy that uses different weights according to HTML tags) implemented in LASER. Rows correspond to the retrieval method used by LASER at query time; columns hold values from subsequent evaluation with other methods. Figures reported are means and two standard errors. This table is taken from [2] .

3 Presentation Bias in Clickthrough Data

Which search engine provides better results: Google or MSNSearch? Evaluating such hypotheses is a problem of statistical inference. Unfortunately, regular clickthrough data is not suited to answer this question in a principled way. Consider the following setup:

Experiment Setup 1 (Regular Clickthrough Data)
The user types a query into a unified interface and the query is sent to both search engines A and B. One of the returned rankings is selected at random and it is presented to the user. The ranks of the links the user clicked on are recorded.

An example of an observation from this experiment is the following: the user types in the query "support vector machine", receives the ranking from search enging B, and then clicks on the links ranked 1, 5, and 6. Data collected by Boyan et al. shows that this setup leads to a strong "presentation bias" [3], making the results difficult to interpret. Consider the average rank of the clicks as a performance measure (e.g. 4 in the example). What can we conclude from this type of clickthrough data?

Table 1 shows the average clickrank for three retrieval strategies averaged over $\approx$ 1400 queries. Rows correspond to the retrieval method presented to the user, while columns show the average clickrank from subsequent evaluation with all retrieval functions. Looking at the diagonal of the table, the average clickrank is almost equal for all methods. However, according to subjective judgments, the three retrieval functions are substantially different in their ranking quality. The lack of difference in the observed average clickrank can be explained as follows. Since users typically scan only the first l (e.g. $l \approx 10$ [19]) links of the ranking, clicking on a link cannot be interpreted as a relevance judgment on an *absolute* scale. Maybe a document ranked much lower in the list was much more relevant, but the user never saw it. It appears that users click on the *relatively* most promising links in the top l, independent of their absolute relevance. This hypothesis is supported by the off-diagonal entries of

Table 1. However, it is difficult to derive a formal interpretation of this type of data.

Other statistics, like the number of links the user clicked on, are difficult to interpret as well. It is not clear if more clicks indicate a better ranking (i.e. the user found more relevant documents) or a worse ranking (i.e. the user had to look at more documents to fulfill the information need). These problems lead to the conclusion that Experiment Setup 1 leads to clickthrough data that is difficult to analyze in a principled way.

4 Unbiased Clickthrough Data for Comparing Search Engines

While the previous experimental setup leads to biased data, we are free to design other forms of presentation that do not exhibit this property. In this light, designing the user interface becomes a question of experiment design. What are the criteria a user interface should fulfill so that clickthrough data is useful?

Blind Test: The interface should hide the random variables underlying the hypothesis test to avoid biasing the user's response. Like patients in medical trials, the user should not know, which one is the "drug" or the "placebo".

Click $\Rightarrow$ Preference: The interface should be designed so that a click during a natural interaction with the system demonstrates a particular judgment of the user.

Low Usability Impact: The interface should not substantially lower the productivity of the user. The system should still be useful, so that users are not turned away.

While Experiment Setup 1 is a blind test, it is not clear how clickthrough is connected to performance. Furthermore, this experiment can have considerable impact on the productivity of the user, since every second query is answered by an inferior retrieval strategy.

4.1 An Experiment Setup for Eliciting Unbiased Data

The following is a more suitable setup for deciding from clickthrough data whether one retrieval strategy is better than another. Under mild assumptions, it generates unbiased data for a hypothesis test from paired observations.

Experiment Setup 2 (Unbiased Clickthrough Data)
The user types a query into a unified interface. The query is sent to both search engines A and B. The returned rankings are mixed so that at any point the top l links of the combined ranking contain the top k_a and k_b links from rankings A and B, $|k_a - k_b| \leq 1$. The combined ranking is presented to the user and the ranks of the links the user clicked on are recorded.

Google Results:

1. Kernel Machines
   http://svm.first.gmd.de/
2. SVM-Light Support Vector Machine
   http://ais.gmd.de/.../svm_light/
3. Support Vector Machine ... References
   http://svm.....com/SVMrefs.html
4. Lucent Technologies: SVM demo applet
   http://svm.....com/SVT/SVMsvt.html
5. Royal Holloway Support Vector Machine
   http://svm.dcs.rhbnc.ac.uk/
6. Support Vector Machine - The Software
   http://www.support–vector.net/software.html
7. Support Vector Machine - Tutorial
   http://www.support–vector.net/tutorial.html
8. Support Vector Machine
   http://jbolivar.freeservers.com/

MSNSearch Results:

1. Kernel Machines
   http://svm.first.gmd.de/
2. Support Vector Machine
   http://jbolivar.freeservers.com/
3. An Introduction to Support Vector Machines
   http://www.support − vector.net/
4. Archives of SUPPORT-VECTOR- ...
   http://www.jiscmail.ac.uk/lists/...
5. SVM-Light Support Vector Machine
   http://ais.gmd.de/.../svm_light/
6. Support Vector Machine - The Software
   http://www.support–vector.net/software.html
7. Lagrangian Support Vector Machine Home Page
   http://www.cs.wisc.edu/dmi/lsvm
8. A Support ... - Bennett, Blue (ResearchIndex)
   http://citeseer.../bennett97support.html

Combined Results:

1. **Kernel Machines**
   http://svm.first.gmd.de/
2. Support Vector Machine
   http://jbolivar.freeservers.com/
3. **SVM-Light Support Vector Machine**
   http://ais.gmd.de/ ∼ thorsten/svm_light/
4. An Introduction to Support Vector Machines
   http://www.support − vector.net/
5. Support Vector Machine and Kernel Methods References
   http://svm.research.bell − labs.com/SVMrefs.html
6. Archives of SUPPORT-VECTOR-MACHINES@JISCMAIL.AC.UK
   http://www.jiscmail.ac.uk/lists/SUPPORT–VECTOR–MACHINES.html
7. **Lucent Technologies: SVM demo applet**
   http://svm.research.bell − labs.com/SVT/SVMsvt.html
8. Royal Holloway Support Vector Machine
   http://svm.dcs.rhbnc.ac.uk/
9. Support Vector Machine - The Software
   http://www.support − vector.net/software.html
10. Lagrangian Support Vector Machine Home Page
    http://www.cs.wisc.edu/dmi/lsvm

Figure 1: Example for query "support vector machine". The two upper boxes show the rankings returned by Google and MSNSearch. The lower box contains the combined ranking presented to the user. The links the user clicked on are marked in bold.

Section 4.2 shows that such a combined ranking always exists. An example is given in Figure 1. The results of two search engines are combined into one ranking that is presented to the user. Note that the abstracts and all other aspects of the presentation are unified, so that the user cannot tell which retrieval strategy proposed a particular page. In the example, the user clicks on links 1, 3, and 7. What inference can one draw from these clicks?

Before going into a detailed statistical analysis in Section 5, let's first analyze this kind of data on an intuitive basis. If one assumes that a user scans the combined ranking from top to bottom without skipping links, this setup ensures that at any point during the scan the user has observed as many (± 1) links from the top of ranking A as from ranking B. In this way, the combined ranking gives (almost) equal presentation bias to both search engines. If one further assumes that the user is more likely to click on a more relevant link, and that the abstract provides enough information to judge relevance better than random, then the clicks convey information about the relative quality of the top $k_a \approx k_b$ links from both retrieval strategies. If the user clicks more often on links from retrieval strategy A, it is reasonable to conclude that the top $k_a \approx k_b$ links from

A are more relevant than those from B. In the example from Figure 1 the user must have seen the top 4 links from both individual rankings, since he clicked on link 7 in the combined ranking. He decided to click on 3 links in the top 4 returned by Google (namely 1, 2, and 4), but only on 1 link from MSNSearch (namely 1). It is reasonable to conclude, that (with probability larger than random) the top 4 links from Google were judged to be better than those from MSNSearch for this query. A detailed analysis of the statistical properties of this type of data is subject to Section 5.

Summarizing Experiment Setup 2, it is a blind test in which clicks demonstrate the relative user preference in an unbiased way. Furthermore, the usability impact is low. In the worst case the user needs to scan twice as many links as for the better individual ranking. But the user is never stuck with just the worse retrieval strategy.

Before analyzing the statistical properties of the data generated in Experiment Setup 2, let's first consider the question of how a combined ranking can be constructed.

4.2 Computing the Combined Ranking

An algorithm for generating a combined ranking according to Experiment Setup 2 is the following.

Algorithm 1 (Combine Rankings)

Input: ranking $A = (a_1, a_2, \ldots)$, *ranking* $B = (b_1, b_2, \ldots)$
Call: combine(A,B,0,0,$\emptyset$*)*
Output: combined ranking D
combine(A,B,k_a*,*k_b*,D)* {
    if($k_a = k_b$*)* {
        if($A[k_a+1] \notin D$*)* { $D := D + A[k_a+1]$; }
        combine(A,B,k_a+1*,*k_b*,D);*
    }
    else {
        if($B[k_b+1] \notin D$*)* { $D := D + B[k_b+1]$; }
        combine(A,B,k_a*,*k_b+1*,D);*
    }
}

The following theorem shows that the algorithm always constructs a combined ranking with the desired property, even if there are duplicates between the two rankings.

Theorem 1 *Algorithm 1 always produces a combined ranking* $D = (d_1, d_2, \ldots)$ *from* $A = (a_1, a_2, \ldots)$ *and* $B = (b_1, b_2, \ldots)$ *so that for all n*

$$\{d_1, \ldots, d_n\} = \{a_1, \ldots, a_{k_a}\} \cup \{b_1, \ldots, b_{k_b}\} \tag{1}$$

with $k_b \le k_a \le k_b + 1$.

Proof Induction over the recursion depth.

Assumption: combine(A,B,k_a,k_b,D) has already constructed a mixed ranking with $\{d_1, \ldots, d_{n_d}\} = \{a_1, \ldots, a_{k_a}\} \cup \{b_1, \ldots, b_{k_b}\}$ and $k_b \le k_a \le k_b + 1$.

Start: Clearly this is true for the initial call combine(A,B,0,0,()).

Step: Given that the assumption is true, there are four cases to consider in the current iteration:

Case $k_a = k_b$ and $A[k_a + 1] \notin D$: Then $A[k_a + 1]$ is appended to D and it holds that $\{d_1, \ldots, d_{n_d}, d_{n_d+1}\} = \{a_1, \ldots, a_{k_a}, a_{k_a+1}\} \cup \{b_1, \ldots, b_{k_b}\}$ and $k_a = k_b + 1$.

Case $k_a = k_b$ and $A[k_a + 1] \in D$: Then $A[k_a + 1]$ is already in D so that $\{d_1, \ldots, d_{n_d}\} = \{a_1, \ldots, a_{k_a}, a_{k_a+1}\} \cup \{b_1, \ldots, b_{k_b}\}$ and $k_a = k_b + 1$.

Case $k_a > k_b$ and $B[k_b + 1] \notin D$: Then $B[k_b + 1]$ is appended to D and it holds that $\{d_1, \ldots, d_{n_d}, d_{n_d+1}\} = \{a_1, \ldots, a_{k_a}\} \cup \{b_1, \ldots, b_{k_b}, b_{k_b+1}\}$ and $k_b = k_a$, since $k_a - k_b \le 1$ by induction.

Case $k_a > k_b$ and $B[k_b + 1] \in D$: Then $B[k_b + 1]$ is already in D so that $\{d_1, \ldots, d_{n_d}\} = \{a_1, \ldots, a_{k_a}\} \cup \{b_1, \ldots, b_{k_b}, b_{k_b+1}\}$ and $k_b = k_a$, since $k_a - k_b \le 1$ by induction.

∎

Note that Algorithm 1 gives ranking A a slight presentation bias, since it starts the combined ranking with a link from A and adds a link from A, if k_a and k_b are equal. To avoid a systematic bias, the retrieval strategy to start with is selected randomly.

For the simplicity reasons, the following treats k_a and k_b as if they were always equal. This is a relatively weak assumption, since the difference between k_a and k_b should have mean 0 due to randomization.

5 Theoretical Analysis

This section analyzes the statistical properties of the clickthrough data generated according to Experiment Setup 2. It will show how, under mild assumptions, this data is sufficient for statistical inference regarding the quality of rankings.

5.1 Connecting Relevance and Clickthrough

Let's consider the standard model of relevance with only two relevance values. Each document is either relevant for a query and a user in a particular context, or not. The quality of a retrieval function is higher the more relevant and the less non-relevant links it retrieves.

For this binary relevance model, the user's clicking behavior with respect to the different retrieval functions can be described using the following model.

Capital letters stand for random variables, while the corresponding non-capitalized letters stands for a realization of that random variable.

$$\Pr(C_a, C_b, C_r, C_n, R_a, N_a, R_b, N_b, R, N) \tag{2}$$

Denote with l the number of links the user observes in the combined ranking. Determined by l, let $k = k_a = k_b$ be the number of links the user observes from the tops of rankings A and B before stopping. Then C_a is the number of clicks on links in the top k of ranking A, while C_b is the number of clicks on links in the top k of ranking B. C_r (C_n) denotes the number of clicks on relevant (non-relevant) links. Note, that c_a plus c_b does not necessarily sum to $c_r + c_n$, since the same link can be in the top k of both retrieval functions. Similarly, R_a, N_a, R_b, and N_b are the numbers of relevant and non-relevant links in the top k of A and B respectively. R and N are the total number of relevant and non-relevant links in the top l of the combined ranking. Note that $r_a + n_a + r_b + n_b$ is not necessarily equal to $l = r + n$, since both retrieval functions may propose the same links.

Which variables in Equation (2) can be observed? Obviously, we can observe C_a and C_b, as well as the total number of clicks $C_r + C_n$. Furthermore, we can approximate l with the rank of the last link the user clicked on. This makes it possible to compute k. To be precise, let $D = (d_1, d_2, \ldots)$ be the combined ranking of $A = (a_1, a_2, \ldots)$ and $B = (b_1, b_2, \ldots)$. Furthermore, let $u_1, \ldots, u_f$ be the ranks in D of the links the user clicked on sorted by increasing rank. Then compute k as the minimum rank of c_{u_f} in A and B

$$k \quad = \quad min\left\{i : d_{u_f} = a_i \text{ or } d_{u_f} = b_i\right\}. \tag{3}$$

Define $k = 0$ for queries without clicks. Furthermore, define C_a and C_b as

$$c_a \quad = \quad |\{u_i : d_{u_i} \in (a_1, \ldots, a_k)\}|, \tag{4}$$

$$c_b \quad = \quad |\{u_i : d_{u_i} \in (b_1, \ldots, b_k)\}|. \tag{5}$$

The central question is now: under which assumptions do these observed variables allow inference regarding the variables of key interest – namely the numbers of relevant links R_a and R_b retrieved by A and B respectively? Let's first state the assumption that users click more frequently on relevant links than on non-relevant links.

Assumption 1 *Given a ranking in which the user encounters r relevant links and n non-relevant links before he stops browsing. Denote with c the number of links the user clicks on, whereas c_r of these links are relevant and c_n are non-relevant. Further denote with r_a and r_b the number of relevant links in the top k of rankings A and B respectively. It holds that*

$$\mathcal{E}\left(\frac{C_r}{R\,C}|r_a - r_b\right) - \mathcal{E}\left(\frac{C_n}{N\,C}|r_a - r_b\right) = \epsilon > 0 \tag{6}$$

for some $\epsilon > 0$ and all differences between r_a and r_b with non-zero probability. $\mathcal{E}(\cdot)$ denotes the expectation.

Intuitively, this assumption formalizes that users click on a relevant link more frequently than on a non-relevant link by a difference of ϵ. The smaller ϵ, the more does the user treat both relevant and non-relevant links the same. In particular, $\epsilon = 0$ if the user clicks on links uniformly at random.

In one respect, this assumption is very weak. It merely implies that users can judge the relevance of a document given its abstract better than random, and that they behave "rational" in the sense that they tend to explore relevant links more frequently. Empirical results indicate that in the setting of interactive retrieval good abstracts can help users identify relevant documents [4]. However, the amount of information an abstract conveys about the relevance of a document is likely to depend on the type of query (e.g. home-page finding vs. fact finding). The assumption also states that the ϵ is constant over all values of $r_a - r_b$. In how far this is true will be evaluated experimentally in Section 6.3.

Given Experiment Setup 2, the clicks on relevant and non-relevant links can be further split up with respect to the different retrieval functions. Let's denote by C_{ra} the number of clicks on relevant links from A and by C_{na} the number of clicks on non-relevant links from A. The analogous quantities for B are C_{rb} and C_{nb}. Controlling the way of presenting the combined ranking to the users, one can make sure that they cannot tell which search engine retrieved a particular link. In the experimental setup used in this study, the same layout and abstract generator were used to present links. So, it is reasonable to assume that the distribution of clicks is not biased towards one retrieval strategy unless there is a difference in the number of relevant links retrieved. This is formalized by the following assumption.

Assumption 2

$$\mathcal{E}(C_{a,r}|c_r, c_n, r_a, n_a, r_b, n_b, r, n) = c_r \frac{r_a}{r} \tag{7}$$

$$\mathcal{E}(C_{a,n}|c_r, c_n, r_a, n_a, r_b, n_b, r, n) = c_n \frac{n_a}{n} \tag{8}$$

$$\mathcal{E}(C_{b,r}|c_r, c_n, r_a, n_a, r_b, n_b, r, n) = c_r \frac{r_b}{r} \tag{9}$$

$$\mathcal{E}(C_{b,n}|c_r, c_n, r_a, n_a, r_b, n_b, r, n) = c_n \frac{n_b}{n} \tag{10}$$

Intuitively, the assumption states that the only reason for a user clicking on a particular link is due to the relevance of the link, but not due to other influence factors connected with a particular retrieval function. One model that will produce the expected values from above is the following. Among the r relevant links, the user clicks on links uniformly without dependence on the retrieval function. This can be modeled by the hypergeometric distribution, which will produce the expected values from above. Note that – as desired – the distribution is symmetric with respect to swapping the retrieval functions A and B.

With these assumptions, it is possible to prove the following theorem. Intuitively, the theorem states that under Experiment Setup 2, evaluating clickthrough will lead to the same result as evaluating relevance judgments.

Theorem 2 *In Experiment Setup 2 and under Assumption 1 and Assumption 2, A retrieves more relevant links than B iff the clickthrough for A is higher than clickthrough for B (and vice versa).*

$$\mathcal{E}(R_a) > \mathcal{E}(R_b) \Longleftrightarrow \mathcal{E}\left(\frac{C_a}{C}\right) > \mathcal{E}\left(\frac{C_b}{C}\right) \tag{11}$$

$$\mathcal{E}(R_a) < \mathcal{E}(R_b) \Longleftrightarrow \mathcal{E}\left(\frac{C_a}{C}\right) < \mathcal{E}\left(\frac{C_b}{C}\right) \tag{12}$$

Proof Let's start with proving (11). Instead of comparing the expected values, it is equivalent to determine the sign of the expected difference as follows.

$$\mathcal{E}\left(\frac{C_a}{C}\right) \quad > \quad E\left(\frac{C_b}{C}\right) \tag{13}$$

$$\Leftrightarrow \mathcal{E}\left(\frac{C_a - C_b}{C}\right) \quad \geq \quad 0 \tag{14}$$

Using that the number of clicks c equals the sum of c_r and c_n, the expected difference can be decomposed.

$$\begin{aligned}
&\mathcal{E}\left(\frac{C_a - C_b}{C}\right) \\
&= \sum \mathcal{E}\left(\frac{C_a - C_b}{c_r + c_n}|c_r, c_n, r_a, n_a, r_b, n_b, r, n\right) \Pr(c_r, c_n, r_a, n_a, r_b, n_b, r, n) \\
&= \sum \mathcal{E}\left(\frac{(C_{a,r}+C_{a,n})-(C_{b,r}+C_{b_n})}{c_r + c_n}|c_r, c_n, r_a, n_a, r_b, n_b, r, n\right) \Pr(...)
\end{aligned}$$

$C_{a,r}$ ($C_{a,n}$) denotes the number of clicks of relevant (non-relevant) links from ranking A. The respective numbers for ranking B are $C_{b,r}$ and $C_{b,n}$. Using Assumption 2 it is possible to replace $\mathcal{E}\left(\frac{(C_{a,r}+C_{a,n})-(C_{b,r}+C_{b_n})}{c_r+c_n}|c_r, c_n, r_a, n_a, r_b, n_b, r, n\right)$ with a closed form expression.

$$\begin{aligned}
&\sum \mathcal{E}\left(\frac{(C_{a,r}+C_{a,n})-(C_{b,r}+C_{b_n})}{c_r + c_n}|c_r, c_n, r_a, n_a, r_b, n_b, r, n\right) \Pr(c_r, c_n, r_a, n_a, r_b, n_b, r, n) \\
&= \quad \sum \frac{1}{c_r + c_n}\left((c_r\frac{r_a}{r} + c_n\frac{n_a}{n}) - (c_r\frac{r_b}{r} + c_n\frac{n_b}{n})\right) \Pr(c_r, c_n, r_a, n_a, r_b, n_b, r, n) \\
&= \quad \sum \frac{1}{c}\left(c_r\frac{r_a - r_b}{r} + c_n\frac{n_a - n_b}{n}\right) \Pr(c_r, c_n, r_a, n_a, r_b, n_b, r, n) \\
&= \quad \sum \frac{1}{c}\left(c_r\frac{r_a - r_b}{r} + c_n\frac{(k - r_a) - (k - r_b)}{n}\right) \Pr(c_r, c_n, r_a, n_a, r_b, n_b, r, n) \\
&= \quad \sum (r_a - r_b)\left(\frac{c_r}{r\,c} - \frac{c_n}{n\,c}\right) \Pr(c_r, c_n, r_a, n_a, r_b, n_b, r, n) \\
&= \quad \sum (r_a - r_b)\mathcal{E}\left(\frac{C_r}{R\,C} - \frac{C_n}{N\,C}|r_a - r_b\right) \Pr(r_a - r_b)
\end{aligned}$$

Using Assumption 1, the expectation $\mathcal{E}\left(\frac{C_r}{R\,C} - \frac{C_n}{N\,C}|r_a - r_b\right)$ is positive and constant, so that it does not influence the following inequality.

$$\begin{aligned}
\sum(r_a - r_b)\mathcal{E}\left(\frac{C_r}{R\,C} - \frac{C_n}{N\,C}|r_a - r_b\right)\Pr(r_a - r_b) &\geq 0 &(15)\\
\Leftrightarrow \sum(r_a - r_b)\Pr(r_a - r_b) &\geq 0 &(16)\\
\Leftrightarrow \mathcal{E}(R_a - R_b) &\geq 0 &(17)\\
\Leftrightarrow \mathcal{E}(R_a) &\geq \mathcal{E}(R_b) &(18)
\end{aligned}$$

The proof of (12) is analogous. ∎

5.2 Hypothesis Tests

The previous section showed that in order to detect a difference in the expected numbers $\mathcal{E}(R_a)$ and $\mathcal{E}(R_b)$ of relevant links in A and B, it is sufficient to prove that

$$\mathcal{E}\left(\frac{C_a - C_b}{C}\right) \tag{19}$$

is different from zero. Given n paired observations $< \frac{c_{a,i}}{c_i}, \frac{c_{b,i}}{c_i} >$, this question can be addressed using a two-tailed paired t-test (see e.g. [16]). It assumes that the difference $X := \frac{C_a}{C} - \frac{C_b}{C}$ is distributed according to a normal distribution. The H_0 hypothesis is that X has zero mean. The t-test rejects H_0 at a significance level of 95%, if

$$\hat{x} \notin \left[-t_{n-1,97.5}\frac{\hat{\sigma}}{\sqrt{n}}, t_{n-1,97.5}\frac{\hat{\sigma}}{\sqrt{n}}\right] \tag{20}$$

where n is the sample size, $\hat{x} = \frac{1}{n}\sum(\frac{c_{a,i}}{c_i} - \frac{c_{b,i}}{c_i})$ is the sample mean, $\hat{\sigma}^2 = \frac{1}{n-1}\sum(\frac{c_{a,i}}{c_i} - \frac{c_{b,i}}{c_i} - \hat{x})^2$ is the sample variance, and $t_{n-1,97.5}$ is the 97.5% quantile point of the t-distribution with $n-1$ degrees of freedom.

In practice, it is difficult to ensure that the assumption of normal distribution holds for small samples. To make sure that the results are not invalidated by an inappropriate parametric test, let's also consider a nonparametric test. Instead of a testing the mean, such tests typically consider the median. In our case, I will use a binomial sign test (i.e. McNemar's test) (see e.g. [16]) to detect a significant deviation of the median

$$\mathcal{M}\left(\frac{C_a - C_b}{C}\right) \tag{21}$$

from zero. Other test like the Wilcoxon rank test are more powerful, but the binomial sign test requires the least assumptions. The binomial sign test counts how often the difference $\frac{c_a}{c} - \frac{c_b}{c}$ is negative and positive. Let the number of negative differences be d_n and the number of positive differences be d_p. If the distribution has zero median, these variables are binomially distributed with parameter $p = 0.5$. The test rejects the H_0 hypothesis of zero median with confidence greater 95%, if

$$2 \sum_{i=0}^{min\{d_p,d_n\}} \left(\begin{array}{c} d_n + d_p \\ i \end{array} \right) 0.5^{d_n+d_p} < 0.05 \tag{22}$$

Note that the median equals the mean for symmetric distributions. Therefore a significant result from a binomial sign test implies a significant difference of the mean under the Gaussian assumption.

In the following empirical evaluation both the t-test and the binomial sign test will be used in parallel.

6 Experiments

To evaluate the method proposed in this paper, it was applied to pairwise comparisons between Google, MSNSearch, and a default strategy. The default strategy is added as a baseline retrieval strategy and consists of the 50 (or less, if fewer hits were returned) links from MSNSearch in reverse order. One can expect that the default strategy performs substantially worse than both Google and MSNSearch.

6.1 Data

The data was gathered from three users (including myself) during the 25th of September and the 18th of October, 2001, using a simple proxy system. The user types the query into a search form which connects to a CGI script. The script selects two search engines in a randomized way, queries the individual search engines, and composes the combined ranking. For each link the URL and the title of the page are presented to the user. The user does not get any clues about which search engine is responsible for which link in the combined ranking. Each click of the user is routed through a proxy that records the action and uses the HTTP-Location command to forward to the desired page.

Over all, 180 queries and 211 clicks were recorded. The average number of clicks per query is 1.17. Among these are 39 queries without clicks. The average number of words per query is 2.31. This is comparable to the findings in [19] who report 2.35 words per query for an AltaVista query log. Reflecting the distribution of WWW usage by researchers in computer science, many of the queries were for personal home pages and known items. For such queries the title and the URL provide a good summary for judging the relevance of a page.

For evaluating the method proposed in this paper, manual relevance judgments were collected for the whole dataset. For each of the 180 queries, the top k links of both retrieval strategies (with k as defined in Equation (3)) were judged according to binary relevance. Again, the judgments were performed in a blind fashion. When assigning relevance judgments it was not observable how any search engine ranked the link, and whether the user clicked on the link. In

A	B	$c_a > c_b$ (A better)	$c_a < c_b$ (B better)	$c_a = c_b > 0$ (tie)	$c_a = c_b = 0$	total
Google	MSNSearch	34	20	46	23	123
Google	Default	18	1	3	12	34
MSNSearch	Default	17	2	1	4	24

Table 2: Comparison using pairwise clickthrough data. The counts indicate for how many queries a user clicked on more links in the top k of the respective search engine.

A	B	$r_a > r_b$ (A better)	$r_a < r_b$ (B better)	$r_a = r_b > 0$ (tie)	$r_a = r_b = 0$	total
Google	MSNSearch	26	17	51	29	123
Google	Default	19	1	1	13	34
MSNSearch	Default	15	1	0	8	24

Table 3: Comparison using manual relevance judgments. The counts indicate for how many queries there were more relevant links in the top k of the respective search engine.

particular, the order in which the links were presented for relevance judgment was randomized to avoid systematic presentation bias. Overall, 180 links were judged to be relevant[1].

6.2 Does the Clickthrough Evaluation Agree with the Relevance Judgments?

Table 2 shows the clickthrough data. Column 3 and 4 indicate for how many queries the user clicked on more links from A or B respectively. According to the binomial sign test, the differences between Google and Default, as well as between MSNSearch and Default are significant. The difference between Google and MSNSearch has a p-value of around 90%. The t-test delivers a similar p-value. On average, 77% of the clicks per query were on links in the top k of Google vs. 63% on links in the top k of MSNSearch. For Google vs. Random (85% vs. 18%) and MSNSearch vs. Random (91% vs. 12%) the difference is again significant.

How does this result compare to an evaluation with manual relevance judgments? Table 3 has the same form as Table 2, but compares the number of links judged relevant instead of the number of clicks. The conclusions from the manual relevance judgments closely follow those from the clickthrough data. Again, the difference between Google and Default, as well as MSNSearch and Default is significant according to the binomial sign test. The difference between Google

[1]The equality with the number of queries is coincidental. Note that these 180 are not all the existing relevant links, but merely those in the region of the ranking that was explored by the user. In particular, no links were manually judged for queries without clicks.

		$R_a - R_b$	
A	B	-1	+1
Google	MSNSearch	0.73 ± 0.11	0.71 ± 0.09
Google	Default	—	0.76 ± 0.08
MSNSearch	Default	—	0.85 ± 0.07

Table 4: The estimated value of ϵ from Assumption 1 depending on the difference $R_a - R_b$ with one standard error. Only such estimates are shown, for which there are more than two observations.

and MSNSearch achieves a p-value of approximately 80%.

For all three comparisons, the result from Theorem 2 holds. The average number of relevant links is higher for Google (0.81) than for MSNSearch (0.72) in their pairwise comparison. For Google vs. Random the averages are 0.65 vs. 0.09, and for MSNSearch vs. Random the averages are 0.71 vs. 0.04. This shows that the difference in clickthrough data from Experiment Setup 2 does not only predict whether one retrieval strategy is better than another, but that it also indicates the quantity of the difference.

While this validates that the model makes reasonable predictions, an analysis of the individual assumptions can provide further prove of its adequacy.

6.3 Is Assumption 1 Valid?

Assumption 1 states that the user clicks on more relevant links than non-relevant links on average. In particular, it states that the difference is independent of how many relevant links were suggested by retrieval strategy A compared to B. Given the relevance judgments, this assumption can be tested against data. Let I_d be the set of queries with $r_a - r_b = d$ and $d \neq 0$. Then Table 4 shows the quantity

$$\hat{\epsilon}_d = \frac{1}{I_d} \sum_{I_d} \frac{c_r}{c\,r} - \frac{1}{I_d} \sum_{I_d} \frac{c_n}{c\,n} \tag{23}$$

for the three pairwise comparisons. Only those averages are shown, for which there were more than 2 observations. The first observation is that the value of ϵ is substantially above 0. This means that, in fact, users click much more frequently on relevant links than on non-relevant links. Furthermore, the particular value of ϵ is rather stable independent of $r_a - r_b$. In particular, all values are within errorbars. While this does not prove the validity of the assumption, it does verify that it is not vastly invalid.

6.4 Is Assumption 2 Valid?

Assumption 2 states that users do not click more frequently on links from one retrieval strategy independent of the relevance of the links. While it would take orders of magnitude more data to verify Assumption 2 in detail, the following

A	B	C_{ra} exp	obs	C_{rb} exp	obs	C_{na} exp	obs	C_{nb} exp	obs
Google	MSNSearch	75.9 $\approx$	78	67.8 $\approx$	67	23.0 $\approx$	26	22.8 $\approx$	22
Google	Default	21.0 $\approx$	21	3.0 $\approx$	3	6.7 $\approx$	10	8.9 $\approx$	8
MSNSearch	Default	15.0 $\approx$	15	1.0 $\approx$	1	5.3 $\approx$	9	5.4 $\approx$	3

Table 5: Compares the expected (exp) number of clicks according to Assumption 2 with the observed (obs) number of clicks.

summary already provides an effective check. If Assumption 2 holds, then the following equalities hold.

$$\mathcal{E}(C_r \frac{R_a}{R}) = \mathcal{E}(C_{r,a}) \tag{24}$$

$$\mathcal{E}(C_r \frac{R_b}{R}) = \mathcal{E}(C_{r,b}) \tag{25}$$

$$\mathcal{E}(C_n \frac{N_a}{N}) = \mathcal{E}(C_{n,a}) \tag{26}$$

$$\mathcal{E}(C_n \frac{N_b}{N}) = \mathcal{E}(C_{n,b}) \tag{27}$$

Accordingly, Table 5 compares the expected number of clicks (i.e. left side of equations) with the observed number of clicks (right side of equations). In general, the equalities appear to hold for real data. Only the observed numbers of clicks on non-relevant links for the comparisons against the default strategy are slightly elevated. However, this is not necessarily an inherent problem of Assumption 2, but more likely a problem with the binary relevance scale. Such relevance judgments cannot model small differences in relevance. This becomes particularly obvious in the comparison of MSNSearch and Default (remember that Default is the top 50 links of MSNSearch in reverse). A link in the top 10 of MSNSearch is likely to be more relevant than one ranked 40-50, even if it is not strictly relevant. The slight user preference for "non-relevant" links from MSNSearch (and Google) is likely to be due to this unmeasured difference in relevance. So, this clicking behavior is desirable, since it is likely to be related to relevance in a more fine-grained relevance model.

7 Conclusions and Future Work

This paper presented a new method for evaluating retrieval functions that does not require (expensive and slow) manual relevance judgments. Its key idea is to design the user interface so that the resulting (cheap and timely) clickthrough data conveys meaningful information about the relative quality of two retrieval functions. This makes it possible to evaluate retrieval performance more economically, without delay, and in a more user-centered way. As desired, the measure reflects the preferences of the users in their current context, not that of an expert giving relevance judgments, and it evaluates only that portion of the ranking observed by the user.

The paper introduces a theoretical model and shows under which assumptions clickthrough data will give the same results as an evaluation using optimal relevance judgments. The predictions of the new method, as well as the individual assumptions are evaluated against real data. The results of the evaluation using clickthrough data were found to closely follow the relevance judgments and the assumptions were found to be reasonable.

Open questions include in how far this method can be applied in other domains. In particular, it is not clear whether the method is equally effective also for other types of users with different search interests and behaviors. Furthermore, it might be possible to incorporate other forms of unintrusive feedback, like time spent on a page, scrolling behavior, etc.

Comparing a small set of hypotheses as considered in this paper is the most basic form of learning. The eventual goal of this research is to automatically learn retrieval functions. While previous such learning approaches [6, 2] require explicit feedback data in form of relevance judgments, first results on exploiting clickthrough data for learning a ranking function from relative preference examples are available [9].

8 Acknowledgements

Many thanks to Katharina Morik and the AI unit at the University of Dortmund for providing their help and the resources for the experiments. Thanks also to Christin Schäfer, Norbert Fuhr, and Phoebe Sengers for helpful comments.

References

[1] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley-Longman, Harlow, UK, May 1999.

[2] B. Bartell, G. Cottrell, and R. Belew. Automatic combination of multiple ranked retrieval systems. In *Annual ACM SIGIR Conf. on Research and Development in Information Retrieval (SIGIR)*, pages 173–181, 1994.

[3] J. Boyan, D. Freitag, and T. Joachims. A machine learning architecture for optimizing web search engines. In *AAAI Workshop on Internet Based Information Systems*, pages 1 – 8, August 1996.

[4] D. D'Souza, M. Fuller, J. Thom, P. Vines, J. Zobel, O. de Kretser, R. Wilkinson, and M. Wu. Melbourne trec-9 experiments. In *Text REtrieval Conference (TREC)*, pages 437–462, 2000.

[5] H. Frei and P. Schäuble. Determining the effectiveness of retrieval algorithms. *Information Processing and Management*, 27(2/3):153–164, 1991.

[6] N. Fuhr. Optimum polynomial retrieval functions based on the probability ranking principle. *ACM Transactions on Information Systems*, 7(3):183–204, 1989.

[7] M. Gordon and P. Pathak. Finding information on the world wide web: the retrieval effectiveness of search engines. *Information Processing and Management*, 35:141–180, 1999.

[8] R. Jin, C. Falusos, and A. Hauptmann. Meta-scoring: Automatically evaluating term weighting schemes in ir without precision-recall. In *Annual ACM SIGIR Conf. on Research and Development in Information Retrieval (SIGIR01)*, pages 83–89, 2001.

[9] T. Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the ACM Conference on Knowledge Discovery and Data Mining (KDD)*, 2002.

[10] T. Joachims, D. Freitag, and T. Mitchell. WebWatcher: a tour guide for the world wide web. In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*, volume 1, pages 770 – 777. Morgan Kaufmann, 1997.

[11] K. S. Jones and C. van Rijsbergen. Report on the need for and provision of an "ideal" information retrieval test collection. British Library Research and Development Report 5266, University of Cambridge, Computer Laboratory, 1975.

[12] H. Leighton and J. Srivastava. First 20 precision among world wide web search services. *Journal of the American Society for Information Science*, 50(10):870–881, 1999.

[13] M. Lesk and G. Salton. Relevance assessments and retrieval system evaluation. *Information Storage and Retrieval*, 4(3):343–359, 1968.

[14] L. Li and Y. Shang. A new method for automatic performance comparison of search engines. *World Wide Web*, 3:241–247, 2000.

[15] H. Lieberman. Letizia: An agent that assists Web browsing. In *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence (IJCAI '95)*, pages 924–929, Montreal, Canada, 1995. Morgan Kaufmann.

[16] A. Mood, F. Graybill, and D. Boes. *Introduction to the Theory of Statistics*. McGraw-Hill, 3rd edition, 1974.

[17] A. Rees and D. Schultz. A field experimental approach to the study of relevance assessments in relation to document searching. NSF Report, 1967.

[18] G. Salton and C. Buckley. Term weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–523, 1988.

[19] C. Silverstein, M. Henzinger, H. Marais, and M. Moricz. Analysis of a very large altavista query log. Technical Report SRC 1998-014, Digital Systems Research Center, 1998.

[20] I. Soboroff, C. Nicholas, and P. Cahan. Ranking retrieval systems without relevance judgements. In *Annual ACM SIGIR Conf. on Research and Development in Information Retrieval (SIGIR01)*, pages 66–73, 2001.

[21] E. Voorhees and D. Harman. Overview of the eighth text retrieval conference. In *The Eighth Text REtrieval Conference (TREC 8)*, 1999.