How Does Clickthrough Data Reflect Retrieval Quality?

Filip Radlinski
Dept. of Computer Science
Cornell University
Ithaca, NY, USA
filip@cs.cornell.edu

Madhu Kurup*
Dept. of Computer Science
Cornell University
Ithaca, NY, USA
mmk222@cs.cornell.edu

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

ABSTRACT

Automatically judging the quality of retrieval functions based on observable user behavior holds promise for making retrieval evaluation faster, cheaper, and more user centered. However, the relationship between observable user behavior and retrieval quality is not yet fully understood. We present a sequence of studies investigating this relationship for an operational search engine on the arXiv.org e-print archive. We find that none of the eight absolute usage metrics we explore (e.g., number of clicks, frequency of query reformulations, abandonment) reliably reflect retrieval quality for the sample sizes we consider. However, we find that paired experiment designs adapted from sensory analysis produce accurate and reliable statements about the relative quality of two retrieval functions. In particular, we investigate two paired comparison tests that analyze clickthrough data from an interleaved presentation of ranking pairs, and we find that both give accurate and consistent results. We conclude that both paired comparison tests give substantially more accurate and sensitive evaluation results than absolute usage metrics in our domain.

Categories and Subject Descriptors

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

General Terms: Measurement, Human Factors.

Keywords: Implicit feedback, retrieval evaluation, expert judgments, clickthrough data.

1. INTRODUCTION

While the traditional Cranfield methodology has proven itself effective for evaluating the quality of ranked retrieval functions, its associated cost and turnaround times are economical only in large domains such as non-personalized Web search. Instead, retrieval applications from Desktop Search, to searching Wikipedia, to Intranet Search demand more flexible and efficient evaluation methods. One promising direction is evaluation based on implicit judgments from observable user behavior such as clicks, query reformulations, and response times. The potential advantages are clear. Unlike expert judgments, usage data can be collected at essentially zero cost, it is available in real time, and it reflects the values of the users, not those of judges far removed from the users' context at the time of the information need.

The key problem with retrieval evaluation based on usage data lies in its proper interpretation – in particular, understanding how certain observable statistics relate to retrieval quality. In this paper, we shed light onto this relationship through a user study with an operational search engine we deployed on the arXiv.org e-print archive. The study follows a controlled experiment design that is unlike previous evaluations of implicit feedback, which mostly investigated document-level relationships between (expert or user annotated) relevance and user behavior (e.g. [1, 8, 10]). Instead, we construct multiple retrieval functions for which we know their relative retrieval quality by construction (e.g. a standard retrieval function vs. the same function with some results randomly swapped within the top 5). Fielding these retrieval functions in our search engine, we test how implicit feedback statistics reflect the difference in retrieval quality.

Specifically, we compare two evaluation methodologies, which we term "Absolute Metrics" and "Paired Comparison Tests". Using absolute metrics for evaluation follows the hypothesis that retrieval quality impacts observable user behavior in an absolute sense (e.g. better retrieval leads to higher-ranked clicks, better retrieval leads to faster clicks). We formulate eight such absolute metrics and hypothesize how they will change with improved retrieval quality. We then test whether these hypotheses hold in our search engine. The second evaluation methodology, paired comparison tests, was first proposed for retrieval evaluation in [12, 13]. They follow experiment designs from the field of sensory analysis (see e.g. [17]). When, for example, studying the taste of a new product, subjects are not asked to independently rate the product on an absolute scale, but are instead given a second product and asked to express a preference between the two. Joachims [12, 13] proposed a method for interleaving the rankings from a pair of retrieval functions so that clicks provide a blind preference judgment. We call this method Balanced Interleaving. In this paper, we evaluate the accuracy of Balanced Interleaving on the arXiv, and also propose a new Team-Draft Interleaving method that overcomes potential problems of Balanced Interleaving for rankings that are close to identical.

The findings of our user study can be summarized as follows. None of the eight absolute metrics reflect retrieval performance in a significant, easily interpretable, and reliable way with the sample sizes we consider. In contrast,

*on educational leave from Yahoo! Inc.

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

both interleaving tests accurately reflect the known differences in retrieval quality, inferring consistent and in most cases significant preferences in the correct direction given the same amount of user behavior data.

2. RELATED WORK

The Cranfield evaluation methodology commonly applied in TREC (see e.g. [23]) uses relevance judgments provided manually by trained experts. For each query, a label specifies the relevance of each document on a graded relevance scale. Given a ranking produced in response to a query, the judgments for the top ranked documents can be aggregated to assess the quality of the ranking. Averaging over many queries yields average performance scores such as NDCG, Mean Average Precision and Precision at K (see e.g. [19]).

However, the process of obtaining expert relevance judgments is time consuming [7] and thus expensive. For instance, when designing Web search systems for subgroups of the general population (for example, academic audiences) or specialized document collections (for instance, digital libraries), the cost of obtaining relevance judgments for evaluation can be prohibitive. Moreover, it can be difficult for expert relevance judges to infer the intent of user queries. Consequently, there is a danger that the resulting annotations are not representative of the true distribution of information needs. Finally, even when expert judgments are available for computing standard performance metrics, some of the metrics have been shown to not necessarily correlate with user-centric performance measures [22].

While a number of researchers have considered how to reduce the amount of labeling effort necessary for evaluation (e.g. [21, 5, 3, 6]), or how to obtain evaluation datasets more representative of realistic usage scenarios (e.g. [20]), we follow an alternative evaluation methodology: measuring the quality of ranking functions without expert judgments, but purely by observing natural user interactions with the search engine. This is motivated by the simplicity of recording user behavior such as querying and clicking. We ask whether there are universal properties of user behavior that can be used to evaluate ranking quality.

Numerous proposals for evaluating ranking quality based on user behavior have previously been explored. Kelly & Teevan give an overview [15]. Most of these fall into the category of Absolute Metrics, which we will evaluate in our user study. For instance, Fox et al. [10] learned to predict whether users were satisfied with specific search queries using implicitly collected feedback. They found a number of particularly indicative features, such as time spent on result pages and how the search session was terminated (e.g., by closing the browser window or by typing a new Internet address). However, many of the most informative features they identified cannot be collected unless users are using a modified Web browser. Similarly, Carterette & Jones [8] looked at whether they can identify the better of two ranking functions using clicks. They found that by training a probabilistic click model, they can predict the probability of relevance for each result. Aggregating over entire rankings, they were able to reliably predict the better of two rankings in terms of NDCG. Others who have studied usage-based retrieval evaluation include [4, 13, 1, 2, 9, 11, 18].

3. DESIGN OF THE USER STUDY

To evaluate the relationship between implicit feedback and ranking quality, we implemented a search engine over the arXiv.org e-print archive[1]. This archive consists of a collection of several hundred thousand academic articles. It is used daily by many thousands of users, predominantly scientists from the fields of physics, mathematics and computer science. Hundreds of these users use our search engine on any particular day.

The basic design of our study can be summarized as follows. Starting with an initial (hand-tuned) ranking function f_1, we derive several other retrieval functions by artificially degrading their retrieval quality compared to f_1. In particular, we constructed triplets of ranking functions $f_1 \succ f_2 \succ f_3$, using the notation $f_i \succ f_j$ to indicate that the retrieval quality of ranking function f_i is better than that of f_j. For each such triplet of ranking functions, we know by construction that f_1 outperforms f_2, and that both outperform f_3. We then expose the users of arxiv.org to these three ranking functions as detailed below, and analyze whether, and under which types of exposure, their observable behavior reflects the known differences in retrieval quality.

Over three one-month periods we fielded triplets of ranking functions in the arXiv.org search engine. Our users were unaware of the experiments being conducted. As the users interacted with the search engine, we recorded the queries issued, and the results clicked on. We then performed aggregate analyses of the observed behavior.

3.1 Constructing Comparison Triplets

We start by describing how we created two sets of ranking functions with known relative retrieval performance. Given that our document collection consisted of academic articles with rich meta-data, we started with an original ranking function, called ORIG, that scores each document by computing a sum of the match between the query and the following document fields: authors, title, abstract, full text, arXiv identifier, article category, article area, article submitter, any journal reference and any author-provided comments. The first four fields are usually most important in matching results to queries. Note that this retrieval function weights, for example, words in the title more heavily, since these title words occur in multiple fields (e.g. title and full text). Our search engine was implemented on top of Lucene[2], which implements a standard cosine similarity matching function.

3.1.1 "ORIG≻FLAT≻RAND"-Comparison

To create the first triplet of ranking functions, we first eliminated much of the meta-data available, then randomized the top search results. Specifically, the first degraded ranking function, FLAT, only computes the sum of the matches in the article full text, author list and article identifier. Note that while the abstract and title are included in the full text, by not scoring contributions on each field independently, we reduced the weight placed on those (usually particularly important) fields. Second, ranking function RAND randomized the order of the top 11 results returned by FLAT. The documents below rank 11 were presented unchanged. By construction, we now have a triplet of ranking functions where it is safe to assume that ORIG $\succ$ FLAT $\succ$ RAND. In fact, our subjective impression is that these three ranking functions deliver substantially different retrieval quality – especially ORIG $\succ$ RAND – and any suitable evaluation method should be able to detect this difference.

[1]Operational at `http://search.arxiv.org/`
[2]Available at `http://lucene.apache.org/`

Figure 1: Screenshot of how results are presented.

3.1.2 "Orig≻Swap2≻Swap4"-Comparison

To create a second triplet of ranking functions that shows a more subtle difference in retrieval quality, we degrade performance in a different way. Starting again with our retrieval function Orig, Swap2 randomly selects two documents in the top 5 and swaps them with two random documents from rank 7 through 11. This swapping pattern is then replicated on all later result pages (i.e., swapping two documents between ranks 11 and 15 with two originally ranked between 17 and 21, etc.). Increasing the degradation, Swap4 is identical to Swap2, but it randomly selects four documents to swap. This gives us a second triplet of ranking functions, where by construction we know that Orig ≻ Swap2 ≻ Swap4. Here we believe the differences are smaller. In particular, the top 11 results always contain the same set of documents, just in a different order.

3.2 Users and System Design

Figure 1 illustrates the user interface of the search engine. It takes a set of keywords as a query, and returns a ranking of 10 results per page. For each result, we show authors, title, year, a query-sensitive snippet, and the arXiv ID of the paper. We register a "click" whenever a user follows a hyperlink associated with a result. These clicks lead to a metadata page from where a PDF is available for download.

3.2.1 User Assignment to Experimental Conditions

Given the nature of the document collection, consisting mostly of scientific articles from the fields of Physics, Mathematics, Computer Science, and to a lesser extent Nonlinear Sciences, Quantitative Biology and Statistics, we suspect that many of our users are researchers and students from these disciplines. On average, our search engine received about 700 queries per day from about 300 distinct IP addresses, registering about 600 clicks on results.

We identify users by their IP address. Since this definition of user is primarily used for identifying spammers and bots, we find it sufficient even though in some cases it may conflate multiple people working on the same computer or through a proxy. The IP address is also used to (pseudo) randomly assign users to various experimental conditions in our study (e.g. the condition "users who receive the results from Flat"). In particular, we segment the user population based on an MD5-hash of IP address and user agent reported by the browser. This method of assignment ensures that users consistently receive the same experiment condition. In particular, any time a user repeats a query, he or she will get exactly the same results.

3.2.2 Data Collection

We recorded queries submitted, as well as clicks on search results. Each record included the experimental condition, the time, IP address, browser, a session identifier and a query identifier.

We define a session as a sequence of interactions (clicks or queries) between a user and the search engine where less than 30 minutes passes between subsequent interactions. When attributing clicks to query results, we only counted clicks occurring within the same session as the query. This was necessary to eliminate clicks that appeared to come from saved or cached search results. Note that it is still possible for clicks to occur hours after the query, if the user was continuously interacting with the search engine.

3.2.3 Quality Control and Testing

To test the system and our experiment setup, we conducted a test run between November 3rd and December 5th, 2007. Based on this data, we refined our methods for data cleaning and spam detection (described below), refined the system and experiment design, and validated the correctness of the software. For all crucial parts of data analysis, the first two authors of this paper each independently implemented analysis code then compared their results to detect potential bugs.

4. EXPERIMENT 1: ABSOLUTE METRICS

We can now ask: Do absolute metrics reflect retrieval quality? We define absolute metrics as usage statistics that can be hypothesized to monotonically change with retrieval quality. In this paper, we explore eight such metrics that quantify the clicking and session behavior of users.

4.1 Absolute Metrics and Hypotheses

We measured the following metrics. Many of them were previously suggested in the literature, as they reflect the key actions that users can choose to perform after issuing a query: clicking, reformulating or abandoning the search.

Abandonment Rate	The fraction of queries for which no results were clicked on.
Reformulation Rate	The fraction of queries that were followed by another query during the same session.
Queries per Session	The mean number of queries issued by a user during a session.
Clicks per Query	The mean number of results that are clicked for each query.
Max Reciprocal Rank†	The mean value of $1/r$, where r is the rank of the highest ranked result clicked on.
Mean Reciprocal Rank†	The mean value of $\sum 1/r_i$, summing over the ranks r_i of all clicks for each query.

Table 1: Absolute metrics for the "ORIG≻FLAT≻RAND" and the "ORIG≻SWAP2≻SWAP4" comparison (± two standard errors / 95% confidence intervals). The second column shows the hypothesized change when retrieval quality is degraded.

		ORIG≻FLAT≻RAND			ORIG≻SWAP2≻SWAP4		
	$\mathcal{H}_1$	ORIG	FLAT	RAND	ORIG	SWAP2	SWAP4
Abandonment Rate (Mean)	$<$	0.680 ± 0.021	0.725 ± 0.020	0.726 ± 0.020	0.704 ± 0.021	0.680 ± 0.021	0.698 ± 0.021
Reformulation Rate (Mean)	$<$	0.247 ± 0.021	0.257 ± 0.021	0.260 ± 0.021	0.248 ± 0.021	0.250 ± 0.021	0.248 ± 0.021
Queries per Session (Mean)	$<$	1.925 ± 0.098	1.963 ± 0.100	2.000 ± 0.115	1.971 ± 0.110	1.957 ± 0.099	1.884 ± 0.091
Clicks per Query (Mean)	$>$	0.713 ± 0.091	0.556 ± 0.081	0.533 ± 0.077	0.720 ± 0.098	0.760 ± 0.127	0.734 ± 0.125
Max Reciprocal Rank (Mean)	$>$	0.554 ± 0.029	0.520 ± 0.029	0.518 ± 0.030	0.538 ± 0.029	0.559 ± 0.028	0.488 ± 0.029
Mean Reciprocal Rank (Mean)	$>$	0.458 ± 0.027	0.442 ± 0.027	0.439 ± 0.028	0.444 ± 0.027	0.467 ± 0.027	0.394 ± 0.026
Time (s) to First Click (Median)	$<$	31.0 ± 3.3	30.0 ± 3.3	32.0 ± 4.0	28.0 ± 2.2	28.0 ± 3.0	32.0 ± 3.5
Time (s) to Last Click (Median)	$>$	64.0 ± 19.0	60.0 ± 14.0	62.0 ± 9.0	71.0 ± 19.0	56.0 ± 10.0	66.0 ± 15.0

Time to First Click† The mean time from query being issued until first click on any result.

Time to Last Click† The mean time from query being issued until last click on any result.

When computing the metrics marked with †, we exclude queries with no clicks to avoid conflating this measure with abandonment rate. For each metric, we hypothesize how we expect the metric to change as retrieval quality decreases:

Metric	Change as ranking gets worse
Abandonment rate	Increase (more bad result sets)
Reformulation rate	Increase (more need to reformulate)
Queries per session	Increase (more need to reformulate)
Clicks per query	Decrease (fewer relevant results)
Max recip. rank	Decrease (top results are worse)
Mean recip. rank	Decrease (more need for many clicks)
Time to first click	Increase (good results are lower)
Time to last click	Decrease (fewer relevant results)

Even if the hypothesized directions of change are incorrect, we at least expect these metrics to change monotonically with retrieval quality. We now test these hypotheses for ORIG≻FLAT≻RAND and ORIG≻SWAP2≻SWAP4.

4.2 Experiment Setup

We evaluate the absolute metrics in two phases. Data for the triplet of ranking functions ORIG≻SWAP2≻SWAP4 was collected from December 19th, 2007 to January 25th, 2008 (Phase I); for the ranking functions ORIG≻FLAT≻RAND, it was collected from January 27th to February 25th, 2008 (Phase II). During each phase, each of the three ranking functions were assigned one experimental condition, receiving 1/6th of search engine visitors. This means that in Phase I, 1/6th of the users saw the results from ORIG, another 1/6th saw the results from FLAT, and yet another 1/6th got the results from RAND. In Phase II, the assignment was done analogously for ORIG, SWAP2, and SWAP4. The remaining 50% of the visitors were assigned to paired comparison conditions described in Section 5.

During our test run prior to these evaluations, we noticed that bots and spammers throw off our results. To compute the absolute metrics robustly, we processed the raw logs as follows. First, we eliminated all users who clicked on more than 100 results on any day of our study. This eliminated under 10 users in each condition. We then computed each metric for every user, averaging over all queries submitted by that user. Finally, we computed the median (for the time to click metrics) or mean (for the others) across all users. Even without complicated heuristics for detecting individual spammers or bots, this per-user aggregation is more robust than naive per-query aggregation. For instance, suppose we have 99 users and one spammer (or bot). Suppose the spammer ran 100 queries and always clicked on all top 10 results, while each of the 99 normal users ran just one query and clicked on one result. The average number of clicks per query that we compute is $(1 \times 10 + 99 \times 1)/100 = 1.09$, not $(100 \times 10 + 99 \times 1)/199 = 5.5$ as in query-based averaging.

4.3 Results and Discussion

The measured values (± two standard errors / 95% confidence interval) are reported in Table 1 for each absolute metric and each ranking function. The column labeled $\mathcal{H}_1$ indicates our hypothesized change in the metric if retrieval quality is decreased. Upon inspection, one observes that none of the metrics consistently follows the hypothesized behavior. The number of pairs $A \succ B$ where the observed value follows (✓) or opposes (↯) the hypothesized change is summarized in the "weak" columns of Table 2. It shows that, for example, the abandonment rate agrees with our hypothesis for four pairs of ranking functions (ORIG ≻ FLAT, FLAT ≻ RAND, ORIG ≻ RAND and SWAP2 ≻ SWAP4). However, for the remaining two pairs, it changes in the opposite direction. Even more strongly, none of the absolute metrics even changes strictly monotonically with retrieval quality.

The lack of consistency with the hypothesized change could partly be due to measurement noise, since the elements of Table 2 are estimates of a population mean/median. The column "signif" of Table 2 shows for how many pairs $A \succ B$ we can significantly (95% one-tailed confidence t-test for mean, χ^2-test for median) reject our hypothesis H_1 (↯) or reject its negation (✓). We do not see a significant difference in the hypothesized direction for more than three out of the six pairs $A \succ B$ for any of the absolute metrics. With the exception of Max Reciprocal Rank, not even the "large difference" pairs ORIG ≻ RAND and ORIG ≻ SWAP4 are consistently significant for any of the metrics. This suggests that, at best, we need substantially more data in order to use these absolute metrics reliably, making them unsuitable for low-volume search applications like desktop search, personalized Web search, and intranet search.

Figures 2 and 3 present a more detailed view of these metrics, giving some insight into how the estimates developed over time. The plots show the respective estimate after the first n distinct users (i.e. distinct IP addresses). Each datapoint represents a different cutoff date on which we computed the metric over all prior data. The error bars indicate one standard error / 66% confidence interval. First, many of the curves still cross towards the end, indicating that

Table 2: Comparing the number of correct ("✓") and false ("↯") preferences implied by the absolute metrics, aggregated over the 'ORIG≻FLAT≻RAND" and the "ORIG≻SWAP2≻SWAP4" comparison. A preference is weakly correct/false, if observed value follows/contradicts the hypothesis. A preference is significantly correct/false, if the difference between the observed values is statistically significant (95%) in the respective direction.

	weak		signif	
	✓	↯	✓	↯
Abandonment Rate (Mean)	4	2	2	0
Reformulation Rate (Mean)	4	2	0	0
Queries per Session (Mean)	3	3	0	0
Clicks per Query (Mean)	4	2	2	0
Max Reciprocal Rank (Mean)	5	1	3	0
Mean Reciprocal Rank (Mean)	5	1	2	0
Time (s) to First Click (Median)	4	1	0	0
Time (s) to Last Click (Median)	4	2	1	1

the estimates have indeed not yet converged with sufficient precision. Second, the plots show that the (Gaussian) error bars are reasonable as confidence intervals for the mean, and therefore so is the t-test. In particular, the curves do indeed terminate within the two standard error interval of most prior data-points. This also suggests that there are no substantial temporal changes (e.g. bot or spam attacks that we do not catch in our pre-processing) within each of the experiments. However, note that in Table 1 the Abandonment Rate and the Time to First Click of ORIG are significantly different between the data collected over Christmas and the data collected in February. Our conjecture is that this is due to differences in user population and context (e.g. break vs. semester). It appears that the impact of these population differences on some of the absolute metrics can be of similar magnitude as the desirable differences due to retrieval quality, confirming that only data collected during the same time period can be meaningfully compared.

5. EXPERIMENT 2: PAIRED COMPARISON TESTS

Paired comparison tests are one of the central experiment designs used in sensory analysis. When testing a perceptual quality of an item (e.g. taste, sound), it is recognized that absolute (Likert-scale) evaluations are difficult to make. Instead, subjects are presented with two or more alternatives and are asked to identify a difference or state a preference. In the simplest case, subjects are given two alternatives and are asked which of the two they prefer. For the evaluation of retrieval functions, this experiment design was first explored in [12, 13]. In particular, this work proposed a method for presenting the results from two retrieval functions so that clicks indicate a user's preference between the two. In contrast to the absolute metrics discussed so far, paired comparison tests do not assume that observable user behavior changes with retrieval quality on some absolute scale, but merely that users can identify the preferred alternative in a direct comparison.

5.1 Balanced Interleaving Method

The key design issue for a paired comparison test between two retrieval functions is the method of presentation. As outlined in [13], the design should be (a) blind to the user with respect to the underlying conditions, (b) it should be robust to biases in the user's decision process that do not relate to retrieval quality, (c) it should not substantially alter the search experience, and (d) it should lead to clicks that reflect the user's preference. The naive approach of simply presenting two rankings side by side would clearly violate (c), and it is not clear whether biases in user behavior would actually lead to meaningful clicks.

To overcome these problems, [12, 13] proposed a presentation where two rankings A and B are interleaved into a single ranking I in a balanced way. The interleaved ranking I is then presented to the user. This particular method of interleaving A and B ensures that any top k results in I always contain the top k_a results from A and the top k_b results from B, where k_a and k_b differ by at most 1. Intuitively, a user reading the results in I from top to bottom will have always seen approximately an equal number of results from each of A and B.

It can be shown that such an interleaved ranking always exists for any pair of rankings A and B, and that it is computed by Algorithm 1 [13]. The algorithm constructs this ranking by maintaining two pointers, namely k_a and k_b, and then interleaving greedily. The pointers are set to always point at the highest ranked result in the respective original ranking that is not yet in the combined ranking. To construct I, the lagging pointer among k_a and k_b is used to select the next result to add to I. Ties are broken randomly. An example of such a combined ranking is presented in the column "Balanced" of Figure 4, separate for each outcome of the initial tie-breaking coin toss.

Given an interleaving I of two rankings presented to the user, one can derive a preference statement from user clicks.

Algorithm 1 Balanced Interleaving

Input: Rankings $A = (a_1, a_2, \dots)$ and $B = (b_1, b_2, \dots)$
$I \leftarrow (); k_a \leftarrow 1; k_b \leftarrow 1;$
$AFirst \leftarrow RandBit()$ .... *decide which ranking gets priority*
while $(k_a \leq |A|) \wedge (k_b \leq |B|)$ **do** ... *if not at end of A or B*
  if $(k_a < k_b) \vee ((k_a = k_b) \wedge (AFirst = 1))$ **then**
    if $A[k_a] \notin I$ **then** $I \leftarrow I + A[k_a]$ .. *append next A result*
    $k_a \leftarrow k_a + 1$
  else
    if $B[k_b] \notin I$ **then** $I \leftarrow I + B[k_b]$ .. *append next B result*
    $k_b \leftarrow k_b + 1$
  end if
end while
Output: Interleaved ranking I

	Input Ranking		Interleaved Rankings					
			Balanced		*Team-Draft*			
Rank	A	B	A first	B first	AAA	BAA	ABA	...
1	a	b	a	b	a^A	b^B	a^A	
2	b	e	b	a	b^B	a^A	b^B	
3	c	a	e	e	c^A	c^A	e^B	
4	d	f	c	c	e^B	e^B	c^A	
5	g	g	d	f	d^A	d^A	d^A	
6	h	h	f	d	f^B	f^B	f^B	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	

Figure 4: Examples illustrating how the Balanced and the Team-Draft methods interleave input rankings A and B for different outcomes of the random coin flips. Superscript for the Team-Draft interleavings indicates team membership.

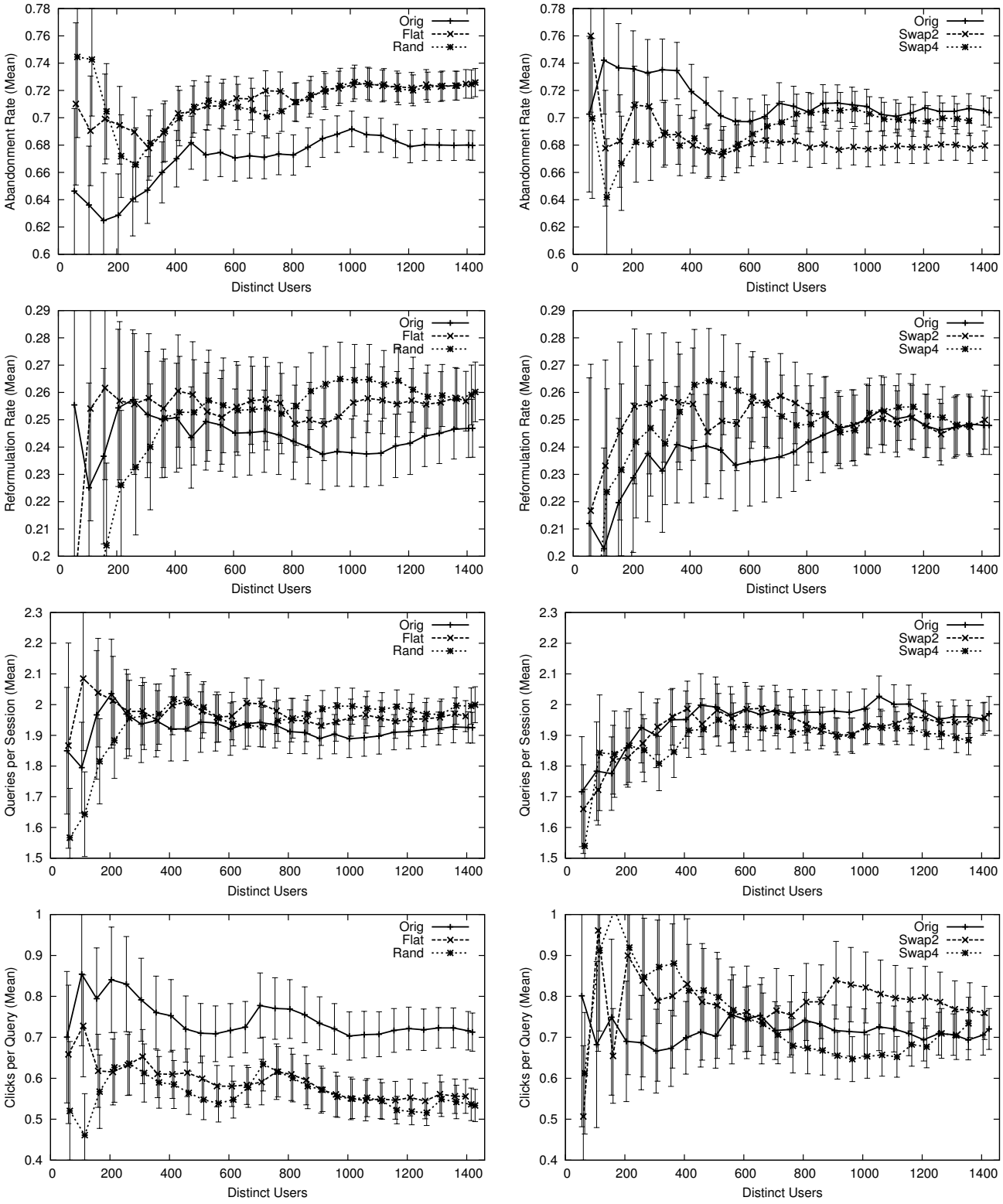

Figure 2: Measurements of the first four absolute performance metrics, for ORIG≻FLAT≻RAND on the left, and ORIG≻SWAP2≻SWAP4 on the right. The error bars indicate one standard error / 66% confidence interval.

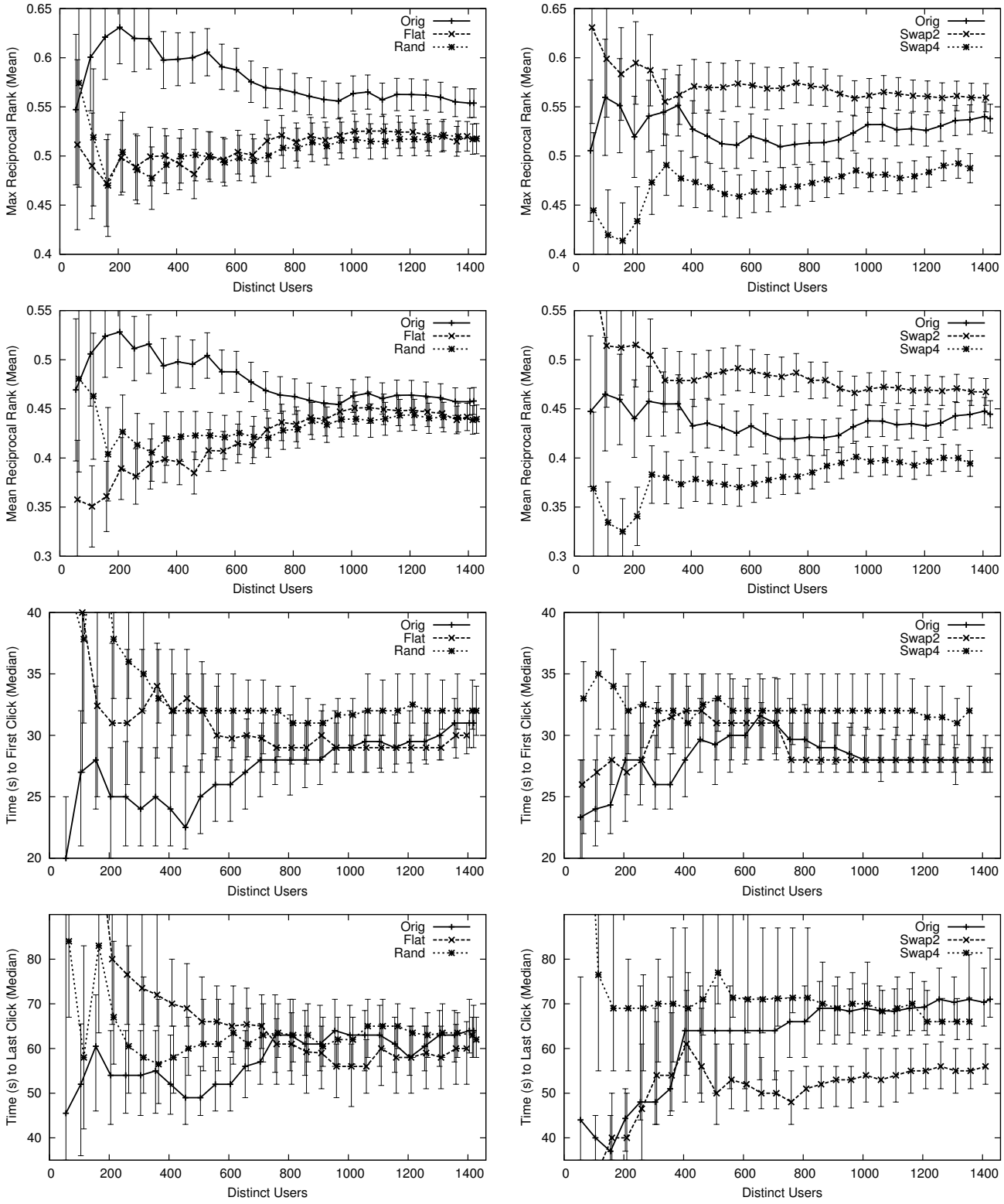

Figure 3: Measurements of the last four absolute performance metrics, for ORIG≻FLAT≻RAND on the left, and ORIG≻SWAP2≻SWAP4 on the right. The error bars indicate one standard error / 66% confidence interval.

In particular, let's assume that the user reads results from top to bottom (as supported by eye-tracking studies [14]), and that the number of links l viewed in I is known and fixed a priori. This means the user has l choices to click on, and an almost equal number came from A and from B. So, a randomly clicking user has approximately an equal chance of clicking on a result from A as from B. If we see significantly more clicks on results from one of the two retrieval functions, we can infer a preference.

More formally, denote $A = (a_1, a_2, ...)$, $B = (b_1, b_2, ...)$, $I = (i_1, i_2, ...)$, and let $c_1, c_2, ...$ be the ranks of the clicks w.r.t. I. To estimate l, [13] proposes to use the lowest ranked click, namely $l \approx c_{max} = \max\{c_1, c_2, ...\}$. Furthermore, to derive a preference between A and B, one compares the number of clicks in the top

$$k = \min\{j : (i_{c_{max}} = a_j) \vee (i_{c_{max}} = b_j)\} \quad (1)$$

results of A and B. In particular, the number h_a of clicks attributed to A and the number h_b of clicks attributed to B is computed as

$$h_a = |\{c_j : i_{c_j} \in (a_1, ..., a_k)\}| \quad (2)$$
$$h_b = |\{c_j : i_{c_j} \in (b_1, ..., b_k)\}|. \quad (3)$$

If $h_a > h_b$ we infer a preference for A, if $h_a < h_b$ we infer a preference for B, and if $h_a = h_b$ we infer no preference.

To further illustrate how preferences are derived from clicks in the interleaved ranking, suppose the user clicked on documents b and e in either of the two balanced interleavings shown in Figure 4. Here, $k = 2$, and the top 3 documents in I were constructed by combining the top 2 results from A and B. Both clicked documents are in the top 2 of ranking B, but only one (b) is in the top 2 or ranking A. Hence the user has expressed a preference for ranking B.

Over a sample of queries and users, denote with $wins(A)$ the number of times A was preferred, and with $wins(B)$ the number of times B was preferred. Using a binomial sign test, we can test whether one ranking function was preferred significantly more often.

5.2 Team-Draft Interleaving Method

Unfortunately, using Eq. 1 to estimate the number of results seen from each ranking can potentially lead to biased results for Balanced Interleaving in some cases, especially when rankings A and B are almost identical up to a small shift or insertion. For example, suppose we have $A = (a, b, c, d)$ and $B = (b, c, d, a)$. Depending on which ranking starts in Alg. 1, interleaving will either produce $I = (a, b, c, d)$ or $I = (b, a, c, d)$. Note that in both cases, a user who clicks uniformly at random on one of the results in I would produce a preference for B more often than for A, which is clearly undesirable. We now describe an interleaving approach that does not suffer from this problem.

The new interleaving algorithm, called Team-Draft Interleaving, follows the analogy of selecting teams for a friendly team-sports match. One common approach is to first select two team captains, who then take turns selecting players for their team. We can use an adapted version of this algorithm for creating interleaved rankings. Suppose each document is a player, and rankings A and B are the preference orders of the two team captains. In each round, captains pick the next player by selecting their most preferred player that is still available, add the player to their team and append the player to the interleaved ranking I. We randomize which captain gets to pick first in each round. The algorithm is summarized in Algorithm 2, and the column "Team-Draft" of Figure 4 gives an illustrative example.

Algorithm 2 Team-Draft Interleaving

Input: Rankings $A = (a_1, a_2, \dots)$ and $B = (b_1, b_2, \dots)$
Init: $I \leftarrow (); TeamA \leftarrow \emptyset; TeamB \leftarrow \emptyset;$
while $(\exists i : A[i] \notin I) \wedge (\exists j : B[j] \notin I)$ **do**
  if $(|TeamA| < |TeamB|) \vee$
    $((|TeamA| = |TeamB|) \wedge (RandBit() = 1))$ **then**
    $k \leftarrow \min_i\{i : A[i] \notin I\}$ ..... *top result in A not yet in I*
    $I \leftarrow I + A[k];$ ..... *append it to I*
    $TeamA \leftarrow TeamA \cup \{A[k]\}$ ..... *clicks credited to A*
  else
    $k \leftarrow \min_i\{i : B[i] \notin I\}$ ..... *top result in B not yet in I*
    $I \leftarrow I + B[k]$ ..... *append it to I*
    $TeamB \leftarrow TeamB \cup \{B[k]\}$ ..... *clicks credited to B*
  end if
end while
Output: Interleaved ranking I, $TeamA$, $TeamB$

To derive a preference between A and B from the observed clicking behavior in I, again denote the ranks of the clicks in the interleaved ranking $I = (i_1, i_2, ...)$ with $c_1, c_2, ...$ We then attribute the clicks to ranking A or B based on which team the clicked result is on. In particular,

$$h_a = |\{c_j : i_{c_j} \in TeamA\}| \quad (4)$$
$$h_b = |\{c_j : i_{c_j} \in TeamB\}|. \quad (5)$$

If $h_a > h_b$ we infer a preference for A, if $h_a < h_b$ we infer a preference for B, and if $h_a = h_b$ we infer no preference. For the example in Figure 4, a user clicking on b and e in the AAA ranking will hit two members of $TeamB$ ($h_b = 2$) and none in $TeamA$ ($h_a = 0$). This generates a preference for B. Note that the randomized alternating assignment of documents to teams and ranks in I ensures that, unlike for Balanced Interleaving, a randomly clicking user will always produce equally many preferences for A as for B in expectation. This avoids the problem of Balanced Interleaving.

5.3 Experiment Setup

We assigned one experimental condition for each pair of retrieval functions within a triplet. To avoid differences due to temporal effects, we conducted the evaluation of the Balanced Interleaving test at the same time as the evaluation of the absolute metrics. This means that data for Balanced Interleaving of Orig≻Swap2≻Swap4 was collected between December 19th, 2007 and January 25th, 2008 (Phase I); data for Balanced Interleaving of Orig≻Flat≻Rand was collected between January 27th and February 25th, 2008 (Phase II). Data for Team-Draft Interleaving was collected between March 15th, 2008, and April 20th, 2008 (Phase III), for both triplets at the same time. In all cases, each experimental condition was assigned 1/6th of the users.

We performed the same data cleaning as for the absolute metrics. However, in addition to user-based aggregation that was essential for estimating the absolute metrics robustly, we also evaluate the paired comparison tests in a query-based fashion. Query-based evaluation simply follows the methods described above, where each query contributes a preference (or tie). So, heavy users provide more preferences. In the user-based evaluation, each user has exactly one "vote" per condition, and the vote is determined by the majority of the individual click preferences of that user.

Table 3: Results of the paired comparison tests for the "ORIG≻FLAT≻RAND" and the "ORIG≻SWAP2≻SWAP4" comparison. Wins and losses are counted on a per-query basis (left) or on a per-user basis (right). We only consider users and queries with at least one click, and their number is given in the table. The remaining percentage of queries/users are ties. Pairs where A (the higher-quality retrieval function) wins significantly (95%) more often than B (the lower-quality retrieval function) are printed in bold.

	Comparison Pair	Query Based			User Based		
	A ≻ B	A wins	B wins	# queries	A wins	B wins	# users
Balanced Interleaving	ORIG ≻ FLAT	**30.6%**	**21.9%**	857	**33.3%**	**23.8%**	538
	FLAT ≻ RAND	**28.0%**	**22.9%**	907	**31.8%**	**23.3%**	529
	ORIG ≻ RAND	**40.9%**	**30.1%**	930	**41.0%**	**27.1%**	553
	ORIG ≻ SWAP2	**18.1%**	**14.6%**	1035	**23.1%**	**17.1%**	589
	SWAP2 ≻ SWAP4	**33.6%**	**27.5%**	1061	35.1%	30.0%	606
	ORIG ≻ SWAP4	**32.1%**	**24.5%**	1173	**37.7%**	**26.7%**	591
Team-Draft Interleaving	ORIG ≻ FLAT	**47.7%**	**37.3%**	1272	**49.6%**	**36.0%**	667
	FLAT ≻ RAND	**46.7%**	**39.7%**	1376	**46.3%**	**36.8%**	646
	ORIG ≻ RAND	**55.6%**	**29.8%**	1095	**58.7%**	**28.6%**	622
	ORIG ≻ SWAP2	44.4%	40.3%	1170	**44.7%**	**37.4%**	693
	SWAP2 ≻ SWAP4	44.2%	40.3%	1202	45.1%	39.8%	703
	ORIG ≻ SWAP4	**47.7%**	**37.8%**	1332	**47.2%**	**35.0%**	697

Table 4: Comparing the number of correct ("✓") and false ("↯") preferences implied by the interleaving methods, analogously to Table 2.

	weak		signif	
	✓	↯	✓	↯
Balanced Interleaving (per query)	6	0	6	0
Balanced Interleaving (per user)	6	0	5	0
Team-Draft Interleaving (per query)	6	0	4	0
Team-Draft Interleaving (per user)	6	0	5	0

5.4 Results and Discussion

Table 3 shows how frequently each ranking functions receives a favorable preference (i.e. "win") in each pairwise comparison for both Balanced Interleaving and Team-Draft Interleaving. For both interleaving methods and also for both query-based and user-based aggregation, the sign of $\Delta_{AB} = wins(A) - wins(B)$ perfectly reflects the true ordering in both ORIG≻FLAT≻RAND and ORIG≻SWAP2≻SWAP4. As summarized in Table 4, in no case do any of the paired tests suggest a preference in the wrong direction. More formally, we statistically test whether the number of wins for the better retrieval function is indeed significantly larger by using a binomial test against $P(A \text{ wins over } B) \leq 0.5$. The significant differences are bolded in Table 3, and 20 out of the 24 pairs are significant. While the remaining four pairs fail the 95% significance level, they are significant at the 90% level. This supports our hypothesis that the paired comparison tests are able to identify a higher-quality retrieval function reliably.

Table 3 does not give substantial evidence that one interleaving or data aggregation method is preferable over another. They all seem to be equally accurate and of roughly equal statistical power. However, note that Team-Draft Interleaving forces a strict preference more often than Balanced Interleaving. For example, any query with a single click always produces a strict preference in Team-Draft Interleaving, even if both rankings are identical. While this does not change the mean, it might lead to larger variability of the results than in Balanced Interleaving, especially for retrieval functions that produce very similar rankings. It appears that the potential problem of Balanced Interleaving identified in Section 5.1 was not an issue in practice.

Interestingly, not only does the sign of Δ_{AB} correspond to the correct ordering by retrieval quality, but the magnitude of this difference appears reasonable as well. In particular, for all tests of a triplet $A \succ B \succ C$, Table 3 shows that $\Delta_{AC} > \max\{\Delta_{AB}, \Delta_{BC}\}$, indicating Strong Stochastic Transitivity [16].

6. DISCUSSION AND LIMITATIONS

As in any controlled experiment, we were able to explore only a few aspects of the problem while keeping many variables in the environment fixed. Most obviously, online retrieval of scientific documents is only one domain for information retrieval and other domains have substantially different properties. In particular, we believe that most of our users were highly educated researchers and students using the system in a research context. Web search, intranet search, personal information search, online purchasing, and mobile search have a much broader and more diverse user base, as well as a different distribution of queries. Since our experiment design is not limited to arXiv.org, it will be interesting to conduct similar studies in those domains as well. The resulting set of studies would give a more complete view of the relationship between user behavior and retrieval quality than the single data point we provide in this paper.

For the sake of simplicity, we focused largely on "raw" clicks as feedback signal. First, this ignores that some clicks may be made in error (e.g. due to a misleading abstract). A more differentiated interpretation of clicks (e.g. based on dwell-time, use of the back button, etc.) may provide a cleaner signal. Second, for some queries the desired information is already presented in the abstract, which obviates the need for a click. Analyzing additional actions such as copy/paste and scan-paths collected via eyetracking may provide additional information that can be incorporated into both the absolute metrics, as well as into the paired comparison tests.

Meaningful abstracts are also essential for collecting meaningful click data. The success of the paired comparison tests suggests that users of arXiv.org were able to make somewhat reliable relevance judgments based on the abstracts. However, generating meaningful abstracts might be more challenging in other domains (e.g. due to spam web pages). Furthermore, one has to be careful that abstract generation is not biased towards any particular retrieval function (e.g. in terms of abstract length or quality).

Apart from a few bots (and possibly a good number of vanity searches), arXiv.org is a domain relatively free of spam. While many domains are similarly free of click-spam (e.g. personal information search, intranet search), it will be interesting to see how the paired comparison tests perform under more substantial click-spam attacks.

While we strove for a set of absolute metrics that covers the majority of observable user behavior, there may be other absolute metrics that are more indicative of ranking quality. For example, there may be sophisticated combinations of various absolute metrics that are more reliable than any single metric [10, 8]. Furthermore, for many of the absolute metrics, the observed differences were not statistically significant given the amount of data we could practically collect. In domains like general Web search, where orders of magnitude more data is available, some of these metrics might indeed make accurate predictions.

Finally, in constructing artificially degraded retrieval functions, we aimed to design both large and small differences. However, further studies are needed to see how fine a difference the paired comparison tests can detect. In particular, it would be interesting to explore whether Strong Stochastic Transitivity holds in other settings, and with even smaller quality differences. If some form of (approximate) stochastic transitivity holds, it is plausible that large numbers of retrieval functions can be reliably evaluated with far less than $O(n^2)$ comparisons using methods from tournament design, which also has implications for automatically learning improved retrieval functions based on paired comparison tests.

7. SUMMARY AND CONCLUSIONS

We explored and contrasted two possible approaches to retrieval evaluation based on implicit feedback, namely absolute metrics and paired comparison tests. In a real-world user study where we know the relative retrieval quality of several ranking functions by construction, we investigated how accurately these two approaches predict retrieval quality. None of the absolute metrics gave reliable results for the sample size collected in our study. In contrast, both paired comparison algorithms, namely Balanced Interleaving as well as the new Team-Draft Interleaving method we propose, gave consistent and mostly significant results. Further studies are needed to extend these results to other search domains beyond the arXiv.org e-print archive.

8. ACKNOWLEDGMENTS

Many thanks to Paul Ginsparg and Simeon Warner for their insightful discussions and their support of the arXiv.org search. The first author was supported by a Microsoft Ph.D. Student Fellowship. This work was also supported by NSF Career Award No. 0237381 and a gift from Google.

9. REFERENCES

[1] E. Agichtein, E. Brill, S. Dumais, and R. Ragno. Learning user interaction models for prediction web search results preferences. In *Proc. of SIGIR*, 2006.

[2] K. Ali and C. Chang. On the relationship between click-rate and relevance for search engines. In *Proc. of Data-Mining and Information Engineering*, 2006.

[3] J. A. Aslam, V. Pavlu, and E. Yilmaz. A sampling technique for efficiently estimating measures of query retrieval performance using incomplete judgments. In *ICML Workshop on Learning with Partially Classified Training Data*, 2005.

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

[5] C. Buckley and E. M. Voorhees. Retrieval evaluation with incomplete information. In *Proc. of SIGIR*, 2004.

[6] B. Carterette, J. Allan, and R. Sitaraman. Minimal test collections for retrieval evaluation. In *Proc. of SIGIR*, 2006.

[7] B. Carterette, P. N. Bennett, D. M. Chickering, and S. T. Dumais. Here or there: Preference judgements for relevance. In *Proc. of ECIR*, 2008.

[8] B. Carterette and R. Jones. Evaluating search engines by modeling the relationship between relevance and clicks. In *Proc. of NIPS*, 2007.

[9] G. Dupret, V. Murdock, and B. Piwowarski. Web search engine evaluation using clickthrough data and a user model. In *WWW Workshop on Query Log Analysis*, 2007.

[10] S. Fox, K. Karnawat, M. Mydland, S. Dumais, and T. White. Evaluating implicit measures to improve web search. *ACM Transactions on Information Science (TOIS)*, 23(2):147–168, April 2005.

[11] S. B. Huffman and M. Hochster. How well does result relevance predict session satisfaction? In *Proc. of SIGIR*, 2007.

[12] T. Joachims. Optimizing search engines using clickthrough data. In *Proc. of KDD*, 2002.

[13] T. Joachims. Evaluating retrieval performance using clickthrough data. In J. Franke, G. Nakhaeizadeh, and I. Renz, editors, *Text Mining*. Physica Verlag, 2003.

[14] 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 Transactions on Information Science (TOIS)*, 25 (2), 2007. Article 7.

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

[16] J. Kozielecki. *Psychological Decision Theory*. Kluwer, 1981.

[17] D. Laming. *Sensory Analysis*. Academic Press, 1986.

[18] Y. Liu, Y. Fu, M. Zhang, S. Ma, and L. Ru. Automatic search engine performance evaluation with click-through data analysis. In *Proc. of WWW*, 2007.

[19] C. D. Manning, P. Raghavan, and H. Schuetze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[20] J. Reid. A task-oriented non-interactive evaluation methodology for information retrieval systems. *Information Retrieval*, 2:115–129, 2000.

[21] I. Soboroff, C. Nicholas, and P. Cahan. Ranking retrieval systems without relevance judgments. In *Proc. of SIGIR*, 2001.

[22] A. Turpin and F. Scholer. User performance versus precision measures for simple search tasks. In *Proc. of SIGIR*, 2006.

[23] E. M. Voorhees and D. K. Harman, editors. *TREC: Experiment and Evaluation in Information Retrieval*. MIT Press, 2005.