Evaluating Search Engine Relevance with Click-Based Metrics*

Filip Radlinski, Madhu Kurup, and Thorsten Joachims

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. In this chapter, we expand upon, Radlinski et al. (How does clickthrough data reflect retrieval quality, In Proceedings of the ACM Conference on Information and Knowledge Management (CIKM), 43–52, 2008), presenting 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 (including the number of clicks observed, the frequency with which users reformulate their queries, and how often result sets are abandoned) 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 find that both give accurate and consistent results. We conclude that both paired comparison tests give substantially more accurate and sensitive evaluation results than the absolute usage metrics in our domain.

*Based on Filip Radlinski, Madhu Kurup, and Thorsten Joachims. How does clickthrough data reflect retrieval quality. In Proceedings of the ACM Conference on Information and Knowledge Management (CIKM), pages 43–52, 2008 ©.

F. Radlinski (✉)
Microsoft Research, Cambridge, UK
e-mail: filiprad@microsoft.com

M. Kurup
Amazon, Seattle, WA, USA
e-mail: madhuk@amazon.com

T. Joachims
Cornell University, Ithaca, NY, USA
e-mail: tj@cs.cornell.edu

J. Fürnkranz and E. Hüllermeier (eds.), *Preference Learning*,
DOI 10.1007/978-3-642-14125-6_16,

1 Introduction

Most search engine evaluation uses the traditional Cranfield methodology, where relevance judgments are provided manually by trained experts. For each query in a test set, the experts provide a label that specifies the relevance of each document in a corpus on a graded relevance scale. Given a ranking produced in response to these queries, the judgments for the top ranked documents can then be aggregated to assess the quality of the ranking. Averaging over many queries yields average performance scores such as Normalized Discounted Cumulative Gain, Mean Average Precision and Precision at K [21].

However, this Cranfield approach presents a number of challenges. First, the process of obtaining expert relevance judgments is time consuming [8] and thus expensive. The associated cost and turnaround times of the Cranfield approach make it economical only in large domains such as non-personalized Web search. Other retrieval applications from Desktop Search, to searching Wikipedia, to Intranet Search usually demand more flexible and efficient evaluation methods.

Second, queries are often ambiguous. The largest Cranfield style test collections generally used to evaluate the performance of ranking algorithms are produced as part of the annual TREC competition [29]. In the TREC setting, each query includes a long description, which is obtained before the judgments are created. When using arbitrary real queries, it can be difficult for expert relevance judges to reliably infer such intents from queries, as they are usually too short to unambiguously identify the users' information needs (e.g., [27]). Consequently, there is a danger that the labels provided may not match the extent to which the documents address the users' actual information needs.

Third, the experts must be knowledgable on the information needs relevant to the collection. When designing Web search systems for specialized subgroups of the general population (for instance, academic audiences) or on specialized document collections (for instance, digital libraries), the cost of obtaining relevance judgments for evaluation can thus become even more prohibitive.

Finally, even when reliable expert judgments are available for computing standard performance metrics, some of the standard metrics have been shown to not always correlate with user-centric performance measures [28].

One promising approach to address these challenges 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, 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. 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., [1,9,12]). Instead, we construct multiple retrieval functions for which we know their relative retrieval quality by construction (e.g., comparing a standard retrieval function versus the same function with some results randomly swapped within the top five positions). Fielding these retrieval functions to real users of our search engine, we test how implicit feedback statistics reflect the difference in retrieval quality. We ask whether there are universal properties of user behavior that can be used to evaluate ranking 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 (such as better retrieval leads to higher-ranked clicks, or 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 by Joachims [14,15]. He follows experiment designs from the field of sensory analysis. When, for example, studying the taste of a new product, subjects are rarely asked to independently rate the product on an absolute scale, but are instead usually given a second product and asked to express a preference between the two (see [19] for a discussion of sensory analysis).

Joachims [14,15] proposed a method for interleaving the rankings from a pair of retrieval functions to mirror such paired comparisons. In this setting, clicks provide a blind preference judgment. We call the algorithm he proposed Balanced Interleaving. In this chapter, we evaluate the accuracy of Balanced Interleaving on arXiv.org, 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, 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

From the perspective of facilitating evaluation of ranking strategies, a number of researchers have considered how to reduce the amount of labeling effort necessary for Cranfield-style evaluation (including [4, 6, 7, 26]), or how to obtain evaluation datasets more representative of realistic usage scenarios (e.g., [25]). However, our focus is on alternative evaluation methodologies, in particular using relevance feedback provided by users.

Two general strategies have been used for obtaining relevance judgements from users: explicitly asking for relevance judgments, or implicitly inferring judgments

from user behavior. Asking users for explicit relevance judgments is onerous, as it essentially requires users to be expert relevance judges. Without an appropriate incentive, users have little motivation to provide high-quality relevance judgments. Hence, evaluating with explicit judgments is generally limited to settings such as movie ranking, where users are more willing to provide reliable judgments in return for personalized movie recommendations (e.g., [10,24]). Evaluations using implicit feedback, based on observing natural user interactions with the search engine, are more practical in general search settings. This is motivated by the simplicity of recording user behavior such as querying and clicking.

Numerous proposals for evaluating ranking quality based on user behavior have previously been explored. Kelly and Teevan give an overview of many previously studied behavioral metrics [17]. Most of these fall into the category of absolute metrics, which we will evaluate in our user study. For instance, Fox et al. [12] learned to predict whether users were satisfied with specific search results 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. In our work, we focus on measures that can be collected from all Web users without requiring additional downloads or browser instrumentation. A number of researchers have studied how to transform clicks into relevance judgments over documents, which could then be aggregated to evaluate ranking performance, including [1, 2, 11, 15, 22]. With a focus on evaluation with just clicks, Carterette and Jones [9] looked at whether they can identify the better of two ranking functions. 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. Using clicks and other implicit feedback directly for evaluation of rankings was also studied by [3,5,13,20]. Most directly related to this work, clicks were first used for evaluation using the Balanced Interleaving paired comparison test described here by Joachims [14,15], although without a controlled study to compare the effectiveness of different evaluation metrics.

In contrast to all the previous studies, we present a controlled real-world experiment evaluating how user behavior given real user-generated queries changes in response to known changes in ranking quality. We measure how user-based evaluation would pick up the differences in quality between five different ranking functions, as measured by eight different absolute click metrics and two pairwise comparison algorithms. A somewhat shorter description of this work was presented in [23].

3 Design of the User Study

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 ranking 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 four 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, leading to the results reported below.

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 metadata, we started with an original ranking function, called ORIG, that scores each document by computing a sum of the match between the query and each of 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 (specifically, in the title field and in the full text field). Our search engine was implemented on top of Lucene,[2] which implements a standard cosine similarity matching function.

[1] Made available at http://search.arxiv.org/

[2] Available at http://lucene.apache.org/

3.1.1 "ORIG≻FLAT≻RAND"-Comparison

To create the first triplet of ranking functions, we first eliminated much of the metadata 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 these (usually particularly important) fields. Second, ranking function RAND reordered the top 11 results returned by FLAT completely at random. Since the nonrandomized ranking was of reasonable quality, randomization reduces the ranking quality. 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.

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 degraded performance in a different way. Starting again with our ranking function ORIG, SWAP2 randomly selects two documents in the top 5 positions and swaps them with two random documents from ranks 7–11. 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.). For instance, if ORIG returned the ten documents $(d_1, d_2, d_3, d_4, d_5, d_6, d_7, d_8, d_9, d_{10})$, SWAP2 might present the user with $(d_1, d_7, d_3, d_9, d_5, d_6, d_2, d_8, d_4, d_{10})$. Increasing the degradation, SWAP4 is constructed identically to SWAP2, except randomly selecting four documents to swap. This gives us a second triplet of ranking functions, where by construction we know that ORIG $\succ$ SWAP2 $\succ$ SWAP4. We believe the quality differences in this triplet are smaller than in the previous triplet. In particular, this is because the top 11 results always contain the same set of documents for all three ranking functions, just in a different order. In contrast, RAND and FLAT return different top 11 documents than ORIG.

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 identifier 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.

Fig. 1 Screenshot of how results are presented

3.2.1 User Characterization

Given the nature of the arXiv 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, as will be described below, we find it sufficient even though in some cases it may conflate multiple people working on the same computer or accessing arXiv.org 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. Moreover, for the rankings that involve randomizing the results returned, the random number generator is seeded with the same information. This method of assignment and seeding ensures that if a user repeats a query within a short time frame, he or she will be shown exactly the same results, making for a consistent user experience.

3.2.2 Data Collection

We recorded queries submitted, as well as clicks on search results for all queries. Each record included the experimental condition, the time, IP address, browser, a unique session identifier, and a unique 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 pass 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 chapter each independently implemented analysis code then compared their results to detect potential errors.

4 Experiment 1: Absolute Metrics

We can now ask: Do absolute metrics reflect retrieval quality? We define absolute metrics as search engine usage statistics that can be hypothesized to monotonically change with retrieval quality. 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 as performance metrics 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.
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 the metrics with abandonment rate. For each metric, we hypothesize how we expect the metric to change as retrieval quality decreases. The explanation for the hypothesized direction of change is noted on the right.

Metric	Hypothesized 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 reciprocal rank	Decrease (top results are worse)
Mean reciprocal 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 $\succ$ FLAT $\succ$ RAND and ORIG $\succ$ SWAP2 $\succ$ SWAP4.

4.2 Experiment Setup

We evaluate the absolute metrics in two phases. Data for the triplet of ranking functions ORIG $\succ$ SWAP2 $\succ$ SWAP4 was collected from December 19th, 2007 to January 25th, 2008 (Phase I); for the ranking functions ORIG $\succ$ FLAT $\succ$ 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 Sect. 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 (IP addresses) 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.[3] This simple per-user aggregation is fairly robust to spammers and bots, much more so 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 $(1 \times 10 + 99 \times 1)/100 = 1.09$, rather than $(100 \times 10 + 99 \times 1)/199 = 5.5$ as it would be with query-based averaging.

4.3 Results and Discussion

The measured values ($\pm$ two standard errors/95% confidence intervals) 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 $\succ$ FLAT, FLAT $\succ$ RAND, ORIG $\succ$ RAND, and SWAP2 $\succ$ 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.

[3] In other words, we report macro-averages rather than micro-averages.

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

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

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 our hypothesized direction of change. A preference is significantly correct/false, if the difference between the observed values is statistically significant (95%) in the respective direction

Absolute metric signals	Weak		Significant	
	✓	↯	✓	↯
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 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 "significant" 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 $\mathcal{H}_1$ (↯) 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 $\succ$ RAND and ORIG $\succ$ SWAP4 are consistently significant for any of the metrics. This suggests that, at best, we need substantially more data to use these absolute metrics reliably, making them unsuitable for low-volume search applications such as 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 as more data was collected. The plots show the respective estimate after the first n distinct users (i.e., distinct IP addresses) were observed. 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. For example, the first point corresponds to roughly the first day of data, after which we had seen 50 distinct users in each experimental condition. The second data point corresponds to taking roughly the first two days, after which we had seen 100 distinct users. As time progressed we saw fewer new distinct IP addresses per day, hence each data point should not be considered as one day. The total experiment duration for each plot was one month.

Consider, as an example absolute metric, the mean abandonment rate for the "ORIG $\succ$ FLAT $\succ$ RAND" and "ORIG $\succ$ SWAP2 $\succ$ SWAP4" experiments as a function of the number of users who have visited the search engine. Note that for "ORIG $\succ$ FLAT $\succ$ RAND", the original (best) ranking function has the lowest abandonment rate, while for "ORIG $\succ$ SWAP2 $\succ$ SWAP4" the original ranking function has the highest abandonment rate.[4] This not only breaks our intuition about abandonment rate, but also indicates that different differences between ranking functions can have different effects on the abandonment rate, making it an unreliable indicator as to the relative quality of ranking functions if our assumed relative ordering of the ranking functions holds.

In general, we see that many of the curves still cross toward the end, indicating that 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 the t-test is also reasonable. In particular, the curves do indeed terminate within the two standard error interval of most prior datapoints. 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 in December/January and the data collected in February. Our conjecture is that this is due to differences in user population and context (e.g., academic 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 differences observed due to retrieval quality, confirming that only data collected during the same time period can be meaningfully compared.

[4] While the two "original ranking function" curves represent the same ranking function, they were collected on two different months thus explaining the variation between them.

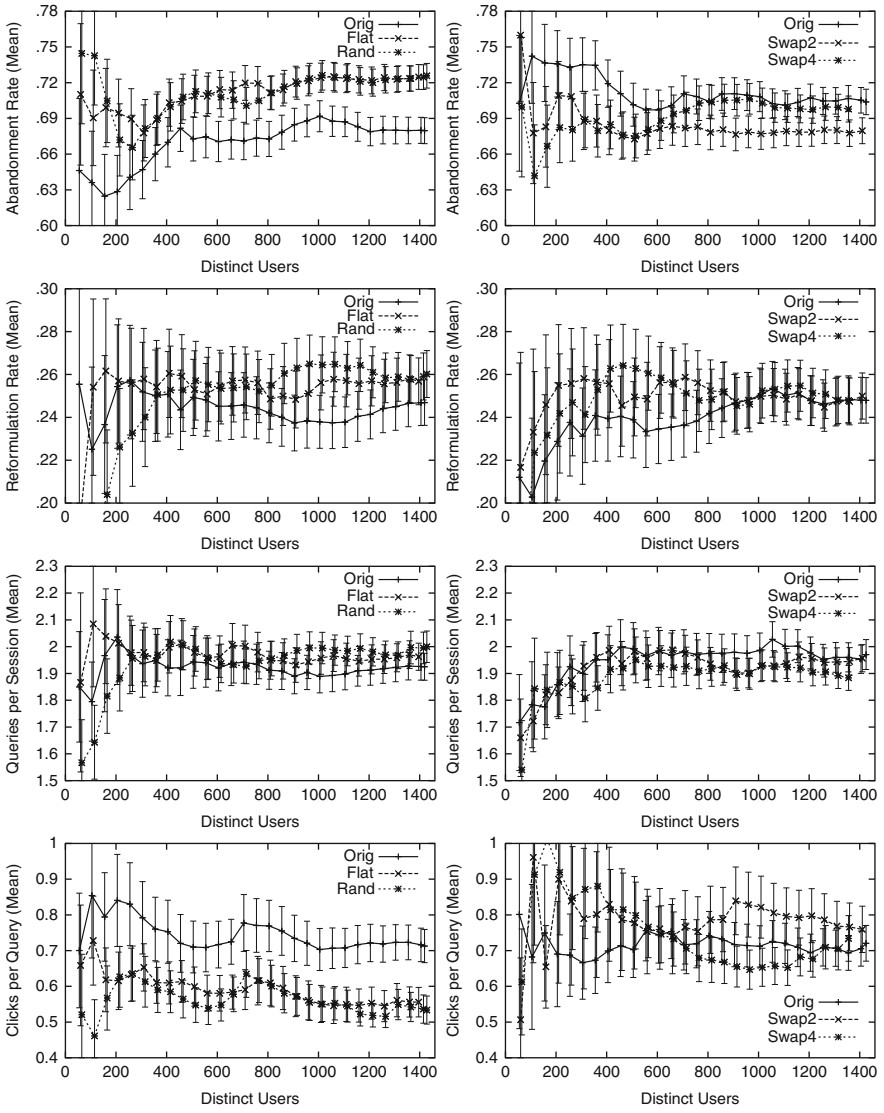

Fig. 2 Measurements of the first four absolute performance metrics, for ORIG ≻ FLAT ≻ RAND on the left, and ORIG ≻ SWAP2 ≻ SWAP4 on the right. The error bars indicate one standard error/66% confidence interval

5 Experiment 2: Paired Comparison Tests

Paired comparison tests are one of the central experiment designs used in sensory analysis [19]. 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,

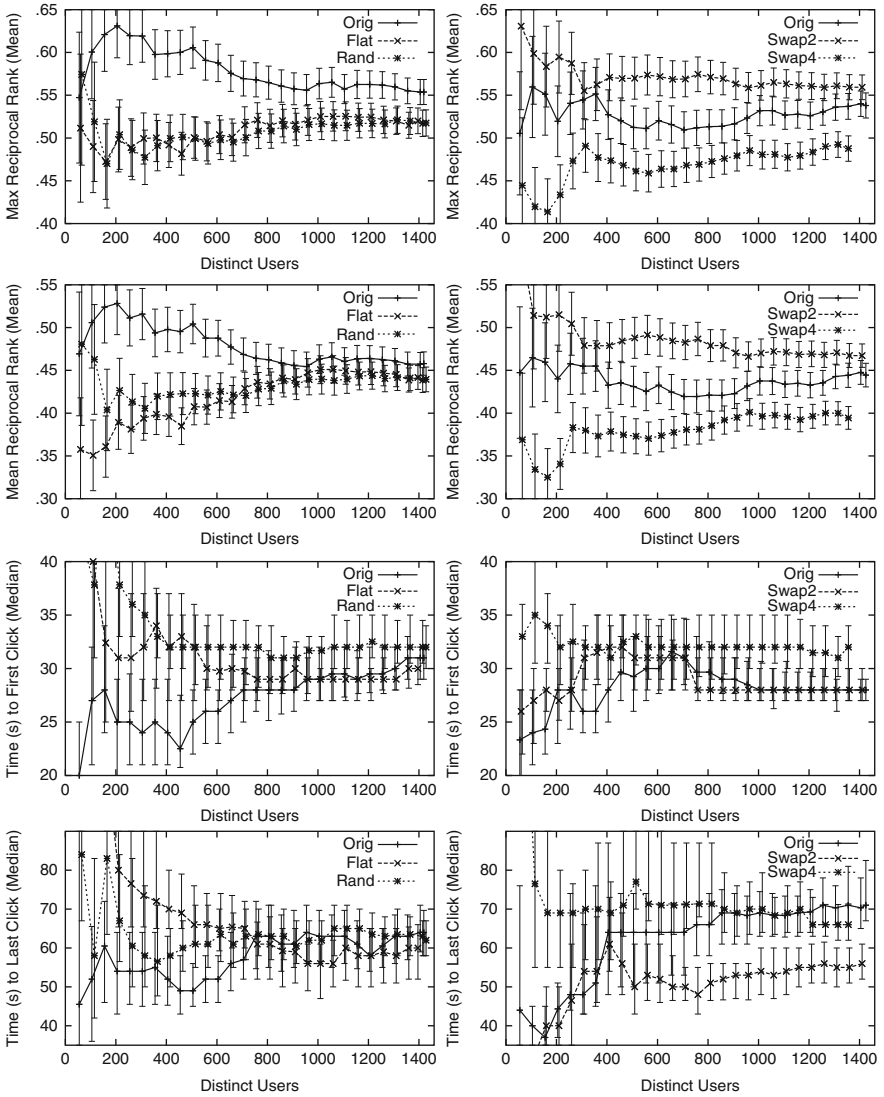

Fig. 3 Measurements of the last four absolute performance metrics, for ORIG ≻ FLAT ≻ RAND on the left, and ORIG ≻ SWAP2 ≻ SWAP4 on the right. The error bars indicate one standard error/66% confidence interval

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 by Joachims [14, 15]. In particular, Joachims proposed a method for presenting the results from two retrieval

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

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 [14], 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 naïve 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, Joachims [14, 15] proposed a presentation where the results present in 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 an approximately 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 [14]. 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.

	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	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	

Fig. 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

Two examples of such combined rankings are presented in the column "Balanced" of Fig. 4. The left column assumes ranking A wins a tie-breaking coin toss, while the right column assumes that ranking B wins the toss.

Given an interleaving I of two rankings presented to the user, one can derive a preference statement from user clicks. In particular, let us assume that the user reads results from top to bottom (as supported by eye-tracking studies [16]), 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 more clicks on results from one of the two retrieval functions, we can infer a preference.

More formally, let $A = (a_1, a_2, \ldots)$ and $B = (b_1, b_2, \ldots)$ be two input rankings we wish to compare. Let $I = (i_1, i_2, \ldots)$ be the combined ranking computed by the Balanced Interleaving algorithm, and let $c_1, c_2, \ldots$ be the ranks of the clicks with respect to I. To estimate l, [14] proposes to use the lowest ranked click, namely $l \approx c_{\max} = \max\{c_1, c_2, \ldots\}$. Furthermore, to derive a preference between A and B, one compares the number of clicks in the top

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

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

$$h_a = |\{c_j : i_{c_j} \in (a_1, \ldots, a_k)\}| \tag{2}$$

$$h_b = |\{c_j : i_{c_j} \in (b_1, \ldots, b_k)\}|. \tag{3}$$

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 Fig. 4. Here, $k = 2$, since 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 (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 two rankings, $A = (a, b, c, d)$ and $B = (b, c, d, a)$. Depending on which ranking wins the tie breaking coin toss in Algorithm 1, interleaving will produce either $I = (a, b, c, d)$ 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. This is because all the documents except a are ranked higher by ranking B, and k is defined as the minimum cutoff that includes all documents. We now describe a new 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 Fig. 4 gives three illustrative examples (e.g., the column "BAA" indicates that captain B picked first in the first round, and that captain A picked first in the second and third rounds).

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 $I = (i_1, i_2, \ldots)$ with $c_1, c_2, \ldots$. We then attribute the clicks to ranking A or B based on which ranking selected the clicked results (or, in the team sport analogy, which team that player was playing for). In particular,

$$h_a = |\{c_j : i_{c_j} \in \textit{Team A}\}| \tag{4}$$

$$h_b = |\{c_j : i_{c_j} \in \textit{Team B}\}|. \tag{5}$$

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 not at end of A or B*
  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, $TeamA$, $TeamB$

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 Fig. 4, a user clicking on b and e in the AAA ranking will click 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 Experimental Evaluation

To compare the effectiveness of absolute metrics with paired comparison evaluation, we assigned one experimental condition to each pair of retrieval functions for each triplet of ranking functions studied. 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 $\succ$ SWAP2 $\succ$ SWAP4 was collected between December 19th, 2007 and January 25th, 2008 (Phase I); data for Balanced Interleaving of ORIG $\succ$ FLAT $\succ$ 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.[5] Unlike for absolute performance metrics, it does not matter if some users run more queries than others: it simply gives heavier users more input into the experiment outcome. Despite this, random spammers or bots will not bias the outcome. For example a user who always clicks on the top result for thousands of queries would reduce the signal in our results, but clicking on an equal number from each "team" would not bias the final outcome. We call this query-based evaluation, and it simply follows the methods described above, where each query contributes a preference (or tie). We will compare the results of this query-based evaluation with user-based evaluation, where each user has exactly one "vote" per condition and that vote is determined by the majority of the individual click preferences of that user.

5.4 Paired Comparison Results

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. We do not count cases when the user did not click at all. For both interleaving methods and also for both query-based and user-based aggregation, the sign of $\Delta_{AB} = (wins(A) - wins(B))/(wins(A) + wins(B))$ perfectly reflects the true ordering in both ORIG ≻ FLAT ≻ RAND and

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

Interleaving	Comparison Pair	Query Based			User Based		
Algorithm	A ≻ B	A wins	B wins	# queries	A wins	B wins	# users
Balanced	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	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

[5] in other words, using micro-averaging.

Table 4 Comparing the number of correct ("✓") and false ("↯") preferences implied by the interleaving methods aggregated over the "ORIG ≻ FLAT ≻ RAND" and the "ORIG ≻ SWAP2 ≻ SWAP4" comparison. A preference is weakly correct/false, if interleaving attributes more wins/losses to the better retrieval functions. A preference is significantly correct/false, if the number of wins is significantly (95%) greater than the number of losses

Paired Comparison Signals	Weak		Significant	
	✓	↯	✓	↯
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

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 the other. They each seem to be equally accurate and of comparable 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 the input rankings are identical. While this does not change the mean, it might lead to larger variance in the individual preference votes than when using Balanced Interleaving, especially for retrieval functions that produce very similar rankings. It appears that the potential problem of Balanced Interleaving identified in Sect. 5.2 was not an issue in this evaluation.

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 [18].

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. In this section, we describe in more detail some of the assumptions that are inherent in the evaluation methods we compare, some of which apply to both absolute and paired comparison evaluation.

6.1 Search Setting

Most obviously, online retrieval of scientific documents is only one domain for information retrieval and other domains may have substantially different properties. In particular, we believe that most of our users were educated researchers and students using the system in a research context. It is possible that our users, for example, consider each result returned more carefully than most Web search users, and delve deeper into the result sets returned. Web search, intranet search, desktop search, online purchasing, and mobile search have a much broader and more diverse user base, as well as a different distribution of queries.

However, as 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 here. From a practical perspective, such an evaluation can be performed in any of these settings without necessitating the creation of a custom search engine as we have implemented for the experiments reported here. In particular, through the simple use of a proxy implemented between users and any general purpose search engine, all of the experiments described here could be performed.

6.2 Click Filtering

For the sake of simplicity, we focused largely on "raw" clicks as feedback signal, with simple heuristics for removing potentially noisy clicks in the case of absolute metrics. This ignores that some clicks may be made in error (e.g., due to a misleading snippet). A more differentiated interpretation of clicks (e.g., based on dwell-time, use of the back button, etc.) may provide a cleaner signal. Additionally, for some queries the desired information is already presented in the snippet, which obviates the need for a click. Analyzing additional actions such as copy/paste and scan-paths collected via eyetracking may provide additional information.

However, it is important to observe that such additional information could be incorporated into *both* absolute metrics as well as paired comparison tests. If clicks followed by a long dwell time on the results are indeed more informative than raw clicks, filtering for such clicks would be expected to improve the strength of the signal observed in all the absolute metrics as well as the strength of the signal observed from interleaved evaluation.

Additionally, if some clicks are malicious, this again may obscure any signal observed. Apart from a few bots (and possibly some vanity searches), arXiv.org is a domain relatively free of click-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.

6.3 Snippets versus Documents

One particular assumption in using raw clicks is that clicks on the short snippets presented to users on results pages tell us about the relevance of the actual documents. The success of the paired comparison tests suggests that users of arXiv.org were able to make somewhat reliable relevance judgments of the articles retrieved based on the snippets generated. To assess the effect of snippets on the experiment results, we also repeated the paired comparison experiment for the ORIG $\succ$ FLAT and ORIG $\succ$ RAND pairs of ranking functions using alternative snippet generation algorithms during a fourth month-long experimental phase. We found that showing normal snippets (about 300 characters), longer snippets (about 450 characters) and simply showing the beginning of the article abstracts resulted in ratios of preference judgments that did not differ in a statistically significant manner from those reported in Table 3. This suggests that the relevance of the articles retrieved can be reliably conveyed in abbreviated form, probably because titles and author names are already very informative in the arXiv.org domain.

However, generating meaningful snippets might be more challenging in other domains (e.g., due to maliciously designed web pages). Furthermore, one has to be careful that snippet generation is not biased toward any particular retrieval function (e.g., in terms of abstract length or quality). In particular, this is one reason why completely independent search engines are difficult to compare with either absolute or interleaving tests. For instance, if results obtained from Web search engine A were simply interleaved with results obtained from Web search engine B, more clicks on the results from A may simply indicate that A produces more misleadingly good snippets, rather than that A is better. Similarly, we could see that A may have a higher abandonment rate in an absolute metric test because the snippets are a little shorter.

6.4 Absolute Metric Choices

While we strove for a set of absolute metrics that covers the majority of easily 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 [9, 12]. 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 absolute metrics might indeed make accurate predictions without the necessity of performing paired comparison tests.

6.5 Scale of Differences

In constructing artificially degraded retrieval functions, we aimed to design both large and small differences in ranking quality. However, further studies are needed to see how fine a difference 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 could be reliably evaluated with far fewer 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 [30, 31].

Additionally, absolute metrics by their nature provide an absolute performance score. This score can then be optimized over time, providing information about long-term improvements to search systems. In contrast, paired comparison tests simply provide information about which ranking is preferred by users without necessarily indicating how much better the preferred ranking function is. It would be interesting to perform studies that measure how the strength of an interleaving signal compares with an absolute measure of ranking performance such as mean average precision or normalized discounted cumulative gain [21].

6.6 Interactive Evaluation Limitations

Finally, interleaved evaluation inherently requires interactive evaluation of ranking functions. This means that a dataset collected for one interleaving evaluation cannot later be reused to evaluate other ranking functions. In particular, as interleaving dynamically creates the ranking presented to users based on two input ranking functions, it would be difficult to infer which results would have been presented and clicked had one of the input functions been different. This differs from some absolute metric evaluations. For instance, if we were to measure mean reciprocal rank, assuming that relevant results tend to be clicked on, a new ranking function that tends to position the previously clicked results closer to the top of the ranking could be assumed to be better than the original one.

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 and the new Team-Draft Interleaving method we proposed, 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.

Acknowledgements 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, NSF Award IIS-0812091 and a gift from Google.

References

1. E. Agichtein, E. Brill, S. Dumais, R. Ragno, Learning user interaction models for prediction web search results preferences, in *Proceedings of the ACM Conference on Research and Development in Information Retrieval (SIGIR)* (2006), pp. 3–10
2. R. Agrawal, A. Halverson, K. Kenthapadi, N. Mishra, P. Tsaparas, Generating labels from clicks, in *Proceedings of ACM International Conference on Web Search and Data Mining (WSDM)* (2009), pp. 172–181
3. K. Ali, C. Chang, On the relationship between click-rate and relevance for search engines, in *Proceedings of Data Mining and Information Engineering* (2006)
4. J.A. Aslam, V. Pavlu, 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)
5. J. Boyan, D. Freitag, T. Joachims, A machine learning architecture for optimizing web search engines, in *AAAI Workshop on Internet Based Information Systems* (1996)
6. C. Buckley, E.M. Voorhees, Retrieval evaluation with incomplete information, in *Proceedings of the ACM Conference on Research and Development in Information Retrieval (SIGIR)* (2004), pp. 25–32
7. B. Carterette, J. Allan, R. Sitaraman, Minimal test collections for retrieval evaluation, in *Proceedings of the ACM Conference on Research and Development in Information Retrieval (SIGIR)* (2006), pp. 268–275
8. B. Carterette, P.N. Bennett, D.M. Chickering, S.T. Dumais, Here or there: Preference judgements for relevance, in *Proceedings of the European Conference on Information Retrieval (ECIR)* (2008), pp. 16–27
9. B. Carterette, R. Jones, Evaluating search engines by modeling the relationship between relevance and clicks, in *Proceedings of the International Conference on Advances in Neural Information Processing Systems (NIPS)* (2007), pp. 217–224
10. K. Crammer, Y. Singer, Pranking with ranking, in *Proceedings of the International Conference on Advances in Neural Information Processing Systems (NIPS)* (2001), pp. 641–647
11. G. Dupret, V. Murdock, B. Piwowarski, Web search engine evaluation using clickthrough data and a user model, in *WWW Workshop on Query Log Analysis* (2007)
12. S. Fox, K. Karnawat, M. Mydland, S. Dumais, T. White, Evaluating implicit measures to improve web search, ACM Trans. Inf. Sci. (TOIS) **23**(2), 147–168 (2005)
13. S.B. Huffman, M. Hochster, How well does result relevance predict session satisfaction? in *Proceedings of the ACM Conference on Research and Development in Information Retrieval (SIGIR)* (2007), pp. 567–573
14. T. Joachims, Evaluating retrieval performance using clickthrough data, in *Text Mining*, ed. by J. Franke, G. Nakhaeizadeh, I. Renz (Physica Verlag, 2003)
15. T. Joachims, Optimizing search engines using clickthrough data, in *Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining (KDD)* (2002), pp. 132–142

16. T. Joachims, L. Granka, B. Pan, H. Hembrooke, F. Radlinski, G. Gay, Evaluating the accuracy of implicit feedback from clicks and query reformulations in web search. ACM Trans. Inf. Sci. (TOIS) **25**(2) (2007), Article 7
17. D. Kelly, J. Teevan, Implicit feedback for inferring user preference: A bibliography. ACM SIGIR Forum **37**(2), 18–28 (2003)
18. J. Kozielecki, *Psychological Decision Theory* (Kluwer, 1981)
19. D. Laming, *Sensory Analysis* (Academic, 1986)
20. Y. Liu, Y. Fu, M. Zhang, S. Ma, L. Ru, Automatic search engine performance evaluation with click-through data analysis, in *Proceedings of the International World Wide Web Conference (WWW)* (2007)
21. C.D. Manning, P. Raghavan, H. Schuetze, *Introduction to Information Retrieval* (Cambridge University Press, 2008)
22. F. Radlinski, T. Joachims, Query chains: Learning to rank from implicit feedback, in *Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining (KDD)* (2005)
23. F. Radlinski, M. Kurup, T. Joachims, How does clickthrough data reflect retrieval quality, in *Proceedings of the ACM Conference on Information and Knowledge Management (CIKM)* (2008), pp. 43–52
24. S. Rajaram, A. Garg, Z.S. Zhou, T.S. Huang, Classification approach towards ranking and sorting problems, in *Lecture Notes in Artificial Intelligence* (2003), pp. 301–312
25. J. Reid, A task-oriented non-interactive evaluation methodology for information retrieval systems. Inf. Retr. **2**, 115–129 (2000)
26. I. Soboroff, C. Nicholas, P. Cahan, Ranking retrieval systems without relevance judgments, in *Proceedings of the ACM Conference on Research and Development in Information Retrieval (SIGIR)* (2001), pp. 66–73
27. A. Spink, D. Wolfram, M. Bernard, J. Jansen, T. Saracevic, Searching to web: The public and their queries. J. Am. Soc. Inf. Sci. Technol. **52**(3), 226–234 (2001)
28. A. Turpin, F. Scholer, User performance versus precision measures for simple search tasks, in *Proceedings of the ACM Conference on Research and Development in Information Retrieval (SIGIR)* (2006), pp. 11–18
29. E.M. Voorhees, D.K. Harman (eds.), *TREC: Experiment and Evaluation in Information Retrieval* (MIT, 2005)
30. Y. Yue, T. Joachims, Interatively optimizing information systems as a dueling bandits problem, in *NIPS 2008 Workshop on Beyond Search: Compuitations Intelligence for the Web* (2008)
31. Y. Yue, T. Joachims, Interatively optimizing information retrieval systems as a dueling bandits problem, in *Proceedings of the International Conference on Machine Learning (ICML)* (2009)