Large-Scale Validation and Analysis of Interleaved Search Evaluation

OLIVIER CHAPELLE, Yahoo! Research
THORSTEN JOACHIMS, Cornell University
FILIP RADLINSKI, Microsoft
YISONG YUE, Carnegie Mellon University

Interleaving is an increasingly popular technique for evaluating information retrieval systems based on implicit user feedback. While a number of isolated studies have analyzed how this technique agrees with conventional offline evaluation approaches and other online techniques, a complete picture of its efficiency and effectiveness is still lacking. In this paper we extend and combine the body of empirical evidence regarding interleaving, and provide a comprehensive analysis of interleaving using data from two major commercial search engines and a retrieval system for scientific literature. In particular, we analyze the agreement of interleaving with manual relevance judgments and observational implicit feedback measures, estimate the statistical efficiency of interleaving, and explore the relative performance of different interleaving variants. We also show how to learn improved credit-assignment functions for clicks that further increase the sensitivity of interleaving.

Categories and Subject Descriptors: H.3.3 [**Information Systems**]: Information Storage and Retrieval—*Information Search and Retrieval*

General Terms: Measurement, Experimentation, Algorithms

Additional Key Words and Phrases: Interleaving, clicks, judgments, search engine, sensitivity, online evaluation

ACM Reference Format:
Chapelle, O., Joachims, T., Radlinski, F., and Yue, Y. 2012. Large-scale validation and analysis of interleaved search evaluation. ACM Trans. Inf. Syst. 30, 1, Article 6 (February 2012), 41 pages.
DOI = 10.1145/2094072.2094078 http://doi.acm.org/10.1145/2094072.2094078

1. INTRODUCTION

Proper evaluation of search quality is essential for developing effective information retrieval systems. While the conventional approach of using expert judgments has proven itself effective in many respects [Voorhees and Harman 2005], it has at least two limitations. First, expert judgments may not reflect the actual relevance and utility that users experience while using a retrieval system, especially since judges often cannot reliably estimate the users' intents (as analyzed, for example, in Chapelle and Zhang [2009, Section 6] and Agrawal et al. [2009, Section 5.3.1]). Second, its associated cost and turnaround times are substantial and often prohibitive. As such, more flexible and efficient evaluation methods are required—especially for applications with resource constraints including desktop search, personalized web search, intranet search, helpdesk support, and academic literature search.

This research was funded in part through NSF Award IIS-0905467 and through a gift from Yahoo!
Author's address: O. Chapelle; email: chap@yahoo-inc.com.

DOI 10.1145/2094072.2094078 http://doi.acm.org/10.1145/2094072.2094078

These limitations have motivated research on alternative approaches to retrieval evaluation, especially approaches based on observable user behavior such as clicks, query reformulations, and response times [Kelly and Teevan 2003]. This offers several potential advantages. Unlike expert judgments, usage data can be collected at essentially zero cost, is available in real time, and reflects the judgments of the users rather than 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 article, we analyze the *interleaving* approach [Joachims 2002, 2003; Radlinski et al. 2008] to solving this key problem. The basic idea behind all variants of the interleaving approach is to perform paired online comparisons of two rankings. This involves merging the two rankings into a single interleaved ranking, and then presenting the interleaved ranking to the user. The algorithm used to produce the interleaved ranking is designed to be "fair," so that users' clicks can be interpreted as unbiased judgments about the relative quality of the two rankings. In this way, interleaving interactively modifies, in a controlled experiment, the search results presented to the user so that the observed user behavior (i.e., clickthrough) is more interpretable. This avoids the problem of post-hoc interpretation of observational data common to most other approaches to interpreting implicit feedback [Fox et al. 2005; Kelly 2005; Agichtein et al. 2006; Dupret et al. 2007; Craswell et al. 2008].

We aim to provide a comprehensive body of evidence regarding its effectiveness, accuracy, and limitations. Specifically, this paper reviews, analyzes, and extends the *Balanced Interleaving* and *Team-Draft Interleaving* methods [Joachims 2002; Radlinski et al. 2008]. The analysis relies on results from a new large-scale field study on the Yahoo! search engine, tied into published and additional unpublished data from experiments on the Bing search engine [Radlinski and Craswell 2010], and the full-text search of the ArXiv.org repository of scientific articles [Radlinski et al. 2008; Yue et al. 2010]. After introducing the two interleaving methods and the systems used for evaluation in the next two sections, we validate and analyze interleaved evaluation by answering a series of specific questions. We ask whether interleaving agrees with the conventional evaluation approach based on relevance judgments collected from experts; whether it agrees with other online metrics; how the statistical sensitivity and reliability of these different alternatives compares; and how to select among different credit-assignment schemes for clicks. Finally, we address the limitations of interleaved evaluation as compared to other approaches in depth.

To provide a complete picture of interleaving as an evaluation technique, this paper combines new data and experiments with data and results from past collaborations with other authors. In particular, we would like to acknowledge the contributions of Madhu Kurup [Radlinski et al. 2008, 2010a], Nick Craswell [Radlinski and Craswell 2010], and Yue Gao and Ya Zhang [Yue et al. 2010].

2. RETRIEVAL EVALUATION AND RELATED WORK

Retrieval quality is most commonly evaluated using one of two approaches: either by using manual judgments of the relevance of documents to queries, or by using observations of how users behave when presented with search results. The former is usually called the Cranfield approach [Cleverdon et al. 1966]. It relies on relevance judgments provided by trained experts, and is commonly used when comparing ranked retrieval systems, such as part of the annual TREC conferences [Voorhees and Harman 2005]. Given a query, the expert judge must provide a label that specifies the relevance of each document on a graded or binary relevance scale. Given a ranking produced in response to a query, the judgments for the top-ranked documents can then be aggregated using

metrics such as Normalized Discounted Cumulative Gain (NDCG), Average Precision (AP) or Precision at K. Across many queries, the mean of these metrics measure the quality of the ranking function (for further details see Manning et al. [2008]).

Since obtaining relevance judgments is both time consuming and expensive [Allan et al. 2008; Carterette et al. 2008], substantial research has focused on how to reduce the amount of labeling effort necessary for reliable evaluation [Soboroff et al. 2001; Buckley and Voorhees 2004; Aslam et al. 2005; Carterette et al. 2006]. However, a benefit of the Cranfield approach is that it produces reusable training and test collections for future research. In addition to the expense of manual labeling, a second challenge of any judgment-based approach is that judges may be unable to infer the user's actual information need simply based on the query issued. This can lead to relevance judgments that inaccurately reflect user utility. For instance, it is known that different users often issue the same textual query while having different information needs or intents [Teevan et al. 2007]. This difficulty can be exacerbated when evaluating retrieval systems that require judges to have appropriate expert background knowledge, such as retrieval engines for specialized user groups (e.g., medical practitioners) or specialized document collections (e.g., digitial video collections). Third, even when expert judgments are available for computing standard metrics, some metrics have been shown to not necessarily correlate with user-centric performance measures [Turpin and Scholer 2006].

Rather than relying on expert relevance judgments, a popular contrasting approach is to evaluate retrieval performance based on implicit feedback directly from the users. Such methods can generally be grouped into two classes, namely Absolute Metrics and Pairwise Preferences. Most prior works fall into the former category, and assume that metrics computed from observable user behavior change monotonically with retrieval quality. In the simplest case, single actions such as clicking or reading time are used as substitutes for explicit relevance judgments [Lieberman 1995; Boyan et al. 1996; Joachims et al. 1997; Morita and Shinoda 1994; White et al. 2002; Liu et al. 2007]. Evaluations and categorizations of such approaches can be found in [Kelly 2005; White et al. 2005; Kelly and Teevan 2003]. More sophisticated approaches combine multiple observable actions to infer user satisfaction at the document, query, or session level [Claypool et al. 2001; Oard and Kim 2001; Fox et al. 2005; Carterette and Jones 2007; Huffman and Hochster 2007]. For instance, Fox et al. [2005] present an approach for predicting session-level user satisfaction from indicators such as time spent on result pages and how the session was terminated (e.g., by closing the browser window or by typing a new Internet address). A key issue here is dealing with presentation bias (e.g., the position of results in the ranking), which has motivated approaches to correct for presentation bias [Dupret et al. 2007; Becker et al. 2007; Craswell et al. 2008; Chapelle and Zhang 2009]. Related to this, Wang et al. [2009] showed that the frequency with which users skip search results can be used to measure ranking relevance. Kelly and Teevan [2003] give an overview of many additional absolute metrics.

Methods based on Pairwise Preferences provide an alternative to such Absolute Metrics. Rather than assuming that user behavior provides an absolute quality score, preference methods merely assume that the better of two (or more) options can be identified based on user behavior. One example is the heuristic that clicked results are preferred over results previously skipped in the ranking [Joachims 2002; Radlinski and Joachims 2006; Joachims et al. 2007] and more sophisticated variants thereof [Radlinski and Joachims 2005; Agichtein et al. 2006]. Interleaving as studied in this paper is a Pairwise Preference method as well. However, unlike most preference methods, its goal is to directly assess the relative quality of different rankings, rather than first eliciting the relevance of individual documents. Furthermore, it actively intervenes in the presentation of results to avoid biases.

ALGORITHM 1: Balanced Interleaving

Input: Rankings $A = (a_1, a_2, \dots)$ and $B = (b_1, b_2, \dots)$
$I \leftarrow (); k_a \leftarrow 1; k_b \leftarrow 1;$
$AFirst \leftarrow RandomBit()$ ..... *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

The first interleaving method, called Balanced Interleaving, was proposed in Joachims [2002, 2003]. However, the idea of mixing search results from multiple retrieval systems for evaluation was already present in Kantor [1988]. It was sketched as a method for having users provide explicit relevance judgments, but to our knowledge was never implemented or empirically tested. In addition to the Balanced Interleaving method, we will also provide a detailed evaluation of the Team-Draft Interleaving method proposed in Radlinski et al. [2008].

The first larger-scale user study investigating the accuracy of interleaving was performed by Ali and Chang [2006] on the Yahoo! Web Search engine. Aggregated over a distribution of queries, they found Balanced Interleaving to accurately reflect the relative quality of two retrieval functions as determined by set-level ratings of human experts. They also investigated the per-query correlation between the preference generated by interleaving and the expert-judged preference between two rankings. They found the correlation to be high for navigational queries and queries with one dominating user intent. For general queries, the correlation is only moderate, but it is unclear to what extent this is due to noise or biases of the interleaving method, or due to ambiguity of the query leading to low interjudge agreement.

He et al. [2009] conducted a simulation-based comparative evaluation of Balanced and Team-Draft interleaving. Using manual judgments of individual documents, they simulated user click behavior to determine which interleaving method provides a less noisy signal. They considered a broad range of query and user types, and found that Balanced Interleaving generally performs as well as or better than Team-Draft Interleaving. They also proposed a new scoring scheme for Balanced Interleaving, which performed well in their simulations. Conversely, Hofmann et al. [2011] found Team-Draft Interleaving to outperform Balanced Interleaving in simulations, and proposed an improved algorithm to further increase the sensitivity of interleaving.

3. INTERLEAVING ALGORITHMS

Interleaving experiments [Joachims 2002, 2003] formulate retrieval evaluation as a paired comparison test between two rankings. Paired comparison tests are one of the central experiment designs used in sensory analysis [Laming 1986]. When testing a perceptual quality of an item (e.g., taste, sound), it is widely recognized that absolute (Likert-scale) evaluations are difficult to perform. 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 asked to choose between two alternatives.

The key design issue for a paired comparison test between two retrieval functions is the method of presentation. As outlined in Joachims [2003], the design should (a) be blind to the user with respect to the underlying conditions, (b) be robust to biases in the user's decision process that do not relate to retrieval quality, (c) not substantially alter the search experience, and (d) 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 allow for meaningful clicks.

Interleaving methods address these problems by merging the two rankings A and B into a single interleaved ranking I, which is presented to the user. The retrieval system observes clicks on the documents in I and attributes them to A, B, or both, depending on the origin of the document. The goal is to make the interleaving process and click attribution as "fair" as possible with respect to biases in user behavior (e.g., position bias [Joachims et al. 2007]), so that clicks in the interleaved ranking I can be interpreted as unbiased feedback for a paired comparison between A and B. The precise definition of "fair" varies for different interleaving methods, but all have the goal of equalizing the influence of biases on clicks in I for A and B. This equalization of behavioral biases is conjectured to be more reliable than explicitly quantifying and correcting for bias after data collection. Furthermore, unlike absolute metrics (e.g., clickthrough rate, abandonment rate, see Section 6), interleaving methods do not assume that observable user behavior changes with retrieval quality on some absolute scale. Rather, they assume users can identify the preferred alternative in a direct comparison.

The following two sections present the methods of *Balanced Interleaving* and *Team-Draft Interleaving*, which differ in the way duplicate documents are treated. The presentation of the methods follows that in Radlinski et al. [2008, 2010a].

3.1. Balanced Interleaving Method

The first interleaving method, called *Balanced Interleaving*, was proposed in Joachims [2002, 2003]. The name reflects the intuition that the results of the two rankings A and B should be interleaved into a single ranking I in a balanced way. This particular method 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 both 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 [Joachims 2003]. The algorithm constructs this ranking by maintaining two pointers, namely k_a and k_b, and then interleaving greedily. The pointers 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.

Two examples of such combined rankings are presented in the column "Balanced" of Figure 1. 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 the user's clicks. For this, we assume that the user examines results from top to bottom (as supported by eye-tracking studies [Joachims et al. 2007]). We denote the number of links in I that the user considers as l. For analysis, we assume that l is known and fixed a priori. This means that the user has l choices to click on, and an almost equal number came from A and from B. As such, a randomly clicking user has an approximately 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, then we can infer a preference.

	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. 1. Examples illustrating how Balanced and Team-Draft Interleaving combine input rankings A and B over different randomizations. Superscript for the Team-Draft interleavings indicates team membership.

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 clicked documents in I. To estimate the number of choices l that the user considered, Joachims [2003] proposed to use the rank of the lowest click observed, namely $l \approx c_{max} = \max_i c_i$. 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. Intuitively, k is the minimum value such that $\{a_1, \ldots, a_k\}$ union $\{b_1, \ldots, b_k\}$ includes all documents in $(i_1, \ldots, i_l)$. 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 and declare a tie.

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 1. Here, $k = 2$, since $l = 3$ 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 of ranking A. Hence the user has expressed a preference for ranking B.

We use statistical hypothesis tests to decide whether users show a significant preference for one retrieval function on a distribution of queries. In the simplest case, one can use the binomial sign test to decide whether users prefer rankings from one retrieval function significantly more often than those from the other [Radlinski et al. 2008].

For convenience, we will use a single statistic to quantify the degree of preference and summarize the outcome of an interleaving experiment. For this purpose, we first define two quantities $wins(A)$ and $wins(B)$. $wins(A)$ is incremented by 1 for any query where A was preferred ($h_a > h_b$). When B was preferred ($h_a < h_b$), $wins(B)$ is incremented by 1. $ties(A, B)$ is incremented by 1 if the users click on at least one result, but $h_a = h_b$. Queries without clicks are ignored. We can now define the statistic

$$\Delta_{AB} = \frac{wins(A) + \frac{1}{2}ties(A, B)}{wins(A) + wins(B) + ties(A, B)} - 0.5 \tag{4}$$

to summarize an interleaving experiment. A positive value of Δ_{AB} indicates that $A \succ B$, a negative value indicates that $B \succ A$. For example if ranker A is preferred to ranker B for 40% of the queries, B is preferred to A for 30% of the queries, and the remaining 30% of queries are ties, this would correspond to $\Delta_{AB} = 5\%$. This counting of binary

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, $TeamA$, $TeamB$

wins and losses is the most basic scoring scheme, and we will explore more nuanced credit assignment methods in Sections 9 and 10.

One drawback of using Equation (1) is that it 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 either produce $I = (a, b, c, d)$ or $I = (b, a, c, d)$. 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. The following alternative interleaving approach does not suffer from this problem.

3.2. Team-Draft Interleaving Method

The *Team-Draft Interleaving* method, introduced in Radlinski et al. [2008], follows the analogy of selecting teams for a friendly team-sports match. When assigning teams, one common approach is to first select two team captains. These captains then take turns selecting players for their team. We can use an adapted version of this algorithm for creating interleaved rankings. Suppose each document represents a player, and rankings A and B are the preference orders of the two team captains. In each round, the 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 1 gives three illustrative examples (for example, 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 $I = (i_1, i_2, \dots)$ with $c_1, c_2, \dots$. 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,

$$h_a = |\{c_j : i_{c_j} \in TeamA\}| \tag{5}$$

$$h_b = |\{c_j : i_{c_j} \in TeamB\}|. \tag{6}$$

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 1, 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 produce equally many preferences for A and for B in expectation. This avoids the problem of Balanced Interleaving described at the end of Section 3.1.

Analogous to Balanced Interleaving, we define the quantities $wins(A)$, $wins(B)$, $ties(A, B)$, and Δ_{AB} as in Equation (4).

While Team-Draft Interleaving has desirable behavior with respect to a random (i.e. noninformative) user, one can also construct examples for Team-Draft Interleaving where the wrong preference is returned. Consider an ambiguous query, where 49% of the users have intent A, 49% have intent B, and 2% have intent C. Assume that all users with intent A are satisfied by document a, all users with intent B are satisfied by document b, and all users with intent C are satisfied by document c. Ranking $R_1 = (a, b, \ldots)$ will therefore satisfy 98% of all users with the top two results, while ranking $R_2 = (b, c, \ldots)$ will satisfy only 51%. However, assuming that users make only a single click on the very first relevant document they encounter, it is easy to verify that R_2 will win under Team-Draft Interleaving (as well as under Balanced Interleaving) in 51% of the comparisons. An alternative construction demonstrating a bias in Team Draft Interleaving is presented in Hofmann et al. [2011, Section 3.2].

Whether an interleaving method exists that does not exhibit such issues is an open question, and we will revisit this question with further discussion in Section 11. Furthermore, in Sections 9 and 10 we will discuss methods that address some of these issues by assigning credit for clicks in a more refined manner. In general, however, it is not clear whether such biases in the comparison of two individual rankings really distort the evaluation of two retrieval functions, since one is evaluating many ranking pairs over a distribution of queries. If erroneous preferences occur uniformly in both directions, they merely add noise but do not change which ranking function wins more often. We will therefore now investigate the accuracy of interleaving evaluation empirically over a wide range of settings.

4. EXPERIMENT DESIGN AND SEARCH ENGINES USED

To explore the accuracy of interleaving evaluation compared to manual judgments and to conventional implicit feedback metrics, we conducted experiments on three search engines: the ArXiv.org full-text search engine,[1] the Bing web search engine,[2] and the Yahoo! web search engine.[3] We now describe how these experiments were designed, what retrieval functions were compared, and what was measured in the experiments.

4.1. Experiment Types

Each experiment on each search engine was designed to address a specific question, and we ensure that every major conclusion is verified in at least two separate experiments and search engines. Experiments were selected to cover different search settings and user populations. Furthermore, we used sets of retrieval functions where the difference in retrieval quality ranges from large to very small. We distinguish two types of experiments, which we term *Noncomparative* and *Paired-Comparison*, respectively.

[1]http://search.arxiv.org.
[2]http://bing.com.
[3]http://www.yahoo.com.

In noncomparative experiments, each experiment condition corresponds to a single retrieval function that is shown to a random sample of users. We refer to such random samples as *buckets*. Non-comparative experiments are used to estimate absolute metrics such as clickthrough rate and abandonment rate for each retrieval function.

In paired-comparison experiments, the experimental condition corresponds to a pair of retrieval functions that is presented via one of the interleaving methods. Using a randomly sampled bucket of users, the respective preference score Δ_{AB} for the pair of retrieval functions as defined in Equation (4) can be estimated from the clicks.

We now describe the experiment setup for each search engine.

4.2. ArXiv.org Full-Text Search

ArXiv.org is a digital collection of more than 700,000 academic articles. It is used daily by many thousands of users, predominantly scientists from the fields of physics, mathematics, and computer science. Hundreds of visitors use its full-text search engine on any particular day. Search results are presented similar to a web-search engine, grouping 10 results per page. For each result, the system shows authors, title, year, a query-sensitive snippet, and the ArXiv identifier of the paper. A "click" is registered whenever a user follows a hyperlink associated with a result. These clicks lead to a metadata page from where the full article is available for download. Radlinski et al. [2008] instrumented this search engine for the experiments on this collection. The following provides an overview of the experiment design for ArXiv.org. More details can be found in Radlinski et al. [2008].

Two triplets of retrieval functions were designed so that their relative retrieval quality is known by construction. Starting with an initial (hand-tuned) ranking function called $\mathcal{E}_{\text{A}}$, several other ranking functions were derived by artificially degrading $\mathcal{E}_{\text{A}}$. In particular, $\mathcal{E}_{\text{A}}$ scores each document by computing the match between the query and several document fields like authors, title, abstract, full text, etc. A degraded retrieval function, $\mathcal{B}_{\text{A}}$, was derived by ignoring fields and aggregating most metadata into a single field. An even stronger degradation was achieved by the third retrieval function, $\mathcal{A}_{\text{A}}$, which randomly reorders the top 11 results of $\mathcal{B}_{\text{A}}$. It is reasonable to conclude that, by construction, $\mathcal{E}_{\text{A}} \succ \mathcal{B}_{\text{A}} \succ \mathcal{A}_{\text{A}}$, 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.

To create a second triplet of ranking functions that shows a more subtle difference in retrieval quality, performance was degraded in a different way. Starting again with the ranking function $\mathcal{E}_{\text{A}}$, $\mathcal{D}_{\text{A}}$ randomly selects two documents in the top 5 positions and swaps them with two random documents from ranks 7 through 11. This swapping pattern is then replicated on all later result pages. Increasing the degradation, $\mathcal{C}_{\text{A}}$ randomly selects four documents to swap. This provides a second triplet of ranking functions, where by construction $\mathcal{E}_{\text{A}} \succ \mathcal{D}_{\text{A}} \succ \mathcal{C}_{\text{A}}$.

Users were randomly but permanently assigned to one experimental condition based on an MD5-hash of their IP address. This MD5-hash is also the seed for any random computations during a search.

Data was collected in three phases, as summarized in Table I. Each phase consisted of six experimental conditions to which users were uniformly assigned. In the non-comparative experiments, some obvious crawlers were removed, since they would have substantially distorted metrics like abandonment rate (see Section 6). Also, we first computed any statistic in the non-comparative experiments for each user, and only then averaged over all users. This gives all users equal "vote" and improves robustness. In the comparative experiments, both Balanced and Team-Draft interleaving were evaluated. No data cleaning was done for the interleaving experiments, since these metrics are naturally robust to crawlers.

Table I.
Summary of experiments conducted and data collected. "Non-Comp" refers to non-comparative experiments, "Balanced" to Balanced Interleaving, and "Team-Draft" to Team-Draft Interleaving. The subscript in the function name refers to the search engine.

	Experimental Condition		Number of	Number of	
	Type	Function(s)	Searches	Days	First Day
ArXiv	Non-Comp	$\mathcal{E}_A$	3,754	30	Jan 27, 2008
	Non-Comp	$\mathcal{B}_A$	4,008	30	Jan 27, 2008
	Non-Comp	$\mathcal{A}_A$	3,798	30	Jan 27, 2008
	Non-Comp	$\mathcal{E}_A$	3,919	38	Dec 19, 2007
	Non-Comp	$\mathcal{D}_A$	3,908	38	Dec 19, 2007
	Non-Comp	$\mathcal{C}_A$	3,439	38	Dec 19, 2007
	Balanced	$\mathcal{E}_A \succ \mathcal{B}_A$	3,410	30	Jan 27, 2008
	Balanced	$\mathcal{E}_A \succ \mathcal{A}_A$	3,510	30	Jan 27, 2008
	Balanced	$\mathcal{B}_A \succ \mathcal{A}_A$	3,785	30	Jan 27, 2008
	Balanced	$\mathcal{E}_A \succ \mathcal{D}_A$	3,860	38	Dec 19, 2007
	Balanced	$\mathcal{E}_A \succ \mathcal{C}_A$	4,036	38	Dec 19, 2007
	Balanced	$\mathcal{D}_A \succ \mathcal{C}_A$	3,848	38	Dec 19, 2007
	Team-Draft	$\mathcal{E}_A \succ \mathcal{B}_A$	4,268	37	Mar 15, 2008
	Team-Draft	$\mathcal{E}_A \succ \mathcal{A}_A$	4,157	37	Mar 15, 2008
	Team-Draft	$\mathcal{B}_A \succ \mathcal{A}_A$	4,911	37	Mar 15, 2008
	Team-Draft	$\mathcal{E}_A \succ \mathcal{D}_A$	4,128	37	Mar 15, 2008
	Team-Draft	$\mathcal{E}_A \succ \mathcal{C}_A$	4,560	37	Mar 15, 2008
	Team-Draft	$\mathcal{D}_A \succ \mathcal{C}_A$	4,406	37	Mar 15, 2008
Bing	Team-Draft	$\mathcal{B}_B \succ \mathcal{A}_B$	220,000	4	July 21, 2009
	Team-Draft	$\mathcal{C}_B \succ \mathcal{A}_B$	190,000	4	Aug 4, 2009
	Team-Draft	$\mathcal{C}_B \succ \mathcal{B}_B$	220,000	4	Aug 11, 2009
	Team-Draft	$\mathcal{D}_B \succ \mathcal{C}_B$	220,000	4	July 7, 2009
	Team-Draft	$\mathcal{F}_B \succ \mathcal{E}_B$	220,000	4	Sept 1, 2009
Yahoo!	Non-Comp	$\mathcal{A}_Y$	73.9 M	33	Mar 17, 2010
	Non-Comp	$\mathcal{B}_Y$	10.4 M	33	Mar 17, 2010
	Non-Comp	$\mathcal{C}_Y$	41.8 M	33	Mar 17, 2010
	Non-Comp	$\mathcal{D}_Y$	72.4 M	33	Mar 17, 2010
	Balanced	$\mathcal{D}_Y \succ \mathcal{C}_Y$	13.9 M	42	May 12, 2010
	Balanced	$\mathcal{D}_Y \succ \mathcal{B}_Y$	1.5 M	5	Apr 14, 2010
	Balanced	$\mathcal{D}_Y \succ \mathcal{A}_Y$	677,000	2	Apr 7, 2010
	Balanced	$\mathcal{C}_Y \succ \mathcal{B}_Y$	1.5 M	5	Apr 14, 2010
	Balanced	$\mathcal{C}_Y \succ \mathcal{A}_Y$	680,000	2	Apr 7, 2010
	Balanced	$\mathcal{B}_Y \succ \mathcal{A}_Y$	1.6 M	5	Apr 9, 2010

4.3. Bing Web Search

Team-Draft Interleaving was implemented on the Bing web search engine, and it was fielded to a small fraction of US users for five pairs of previously developed Web search retrieval functions [Radlinski and Craswell 2010]. The pairs of retrieval functions can be split into two sets, based on the magnitude of the differences between the ranking functions. The first set uses three retrieval functions, named $\mathcal{A}_B$, $\mathcal{B}_B$, and $\mathcal{C}_B$, which represent major revisions of the web search ranker. Experiment $\mathcal{B}_B \succ \mathcal{A}_B$ compared rankers $\mathcal{A}_B$ and $\mathcal{B}_B$, with experiments $\mathcal{C}_B \succ \mathcal{B}_B$ and $\mathcal{C}_B \succ \mathcal{A}_B$ named analogously. The differences between these rankers involve changes of over half a percentage point, in absolute terms, of MAP and NDCG. Although such differences can be considered small, they are also of similar magnitude to those commonly reported in research publications.

The remaining two pairs of retrieval functions involve smaller modifications to the ranking system, we term these $\mathcal{D}_B \succ \mathcal{C}_B$ and $\mathcal{F}_B \succ \mathcal{E}_B$. The overall differences involve changes in retrieval performance of under 0.2 percentage points of MAP and NDCG. Such differences are typical for incremental changes made during algorithm development. This is one reason why very sensitive methods like interleaving are needed to

compare functions at reasonable costs during the development cycle. The pair $\mathcal{D}_B \succ \mathcal{C}_B$ involves a small change in search engine parameters, resulting in a small effect on many queries. The pair $\mathcal{F}_B \succ \mathcal{E}_B$ involves a change in the processing of some rare queries, resulting in a large effect on a small fraction of queries.

During the experiment, the rankings produced by Team-Draft Interleaving were shown to a small fraction of Bing users over multiple days until 220,000 user searches with non-adult queries with clicks had been observed.[4] The experiments were performed in succession over two months from July 2009, with each experiment run on the same days of the week (Tuesday through Friday) to avoid any weekday/weekend effects. A summary of the experiments is given in Table I.

In addition to the online experiments, manual relevance judgments were collected as well. Specifically, judgements for approximately 12,000 queries previously sampled from the search engine workload were gathered during the same time period when the interleaving experiments were performed. The relevance of the top results returned by each ranker in the interleaving experiments was assessed by trained judges on a five-point scale ranging from "bad" to "perfect." As MAP requires binary relevance judgments, binarized versions of the ratings were created by taking the top two levels as relevant, and bottom three as nonrelevant. These rating were then used to measure difference in MAP[5] and NDCG@5 for each pair of retrieval functions.

4.4. Yahoo! Web Search

Balanced Interleaving was implemented and fielded on the Yahoo! web search engine. A bucket of users from the US market was assigned to each experiment performed, consisting of a small percentage[6] of incoming traffic. We compared all pairings of four ranking functions. Function $\mathcal{A}_Y$ was the current production version at the time of the comparison. Functions $\mathcal{B}_Y$, $\mathcal{C}_Y$ and $\mathcal{D}_Y$ were candidate functions for the next release, trained on a larger training set and with more features than function $\mathcal{A}_Y$. Functions $\mathcal{C}_Y$ and $\mathcal{D}_Y$ are most similar: they were trained with the same algorithm, but different parameters. Function $\mathcal{B}_Y$ was trained using a different objective function.

Table I summarizes the online experiments that were conducted between March and May 2010 for all six pairs of retrieval functions. The Balanced Interleaving algorithm was used for all comparative experiments. Furthermore, all retrieval functions were also fielded in the noncomparative experiment setup to compute absolute metrics such as clickthrough and abandonment rate.

The four noncomparative experiments—one for each ranking function—were run simultaneously over more than a month. The interleaving experiments lasted for shorter periods of time, either two or five days. The only exception is the interleaving experiment comparing functions $\mathcal{C}_Y$ and $\mathcal{D}_Y$. These functions are extremely close in terms of relevance, and the original 5 day bucket was not long enough to reach a statistically significant conclusion. It was therefore reprogrammed to run for 42 days.

Similar to the experiments on Bing, manual relevance judgments were gathered for the Yahoo! experiments. 2,000 queries were sampled for manual judging, focussing on DCG@5 as the performance measure. All four retrieval functions have very similar DCG@5 scores, and the maximum relative difference[7] between them is 0.65%. Ranking the functions by DCG@5 suggests the ordering $\mathcal{D}_Y \succ \mathcal{C}_Y \succ \mathcal{B}_Y \succ \mathcal{A}_Y$. However, none of

[4]With the exception of the $\mathcal{C}_B \succ \mathcal{A}_B$ pair, for which only 190,000 user searches were collected.

[5]Note that instead of measuring MAP down to a deep rank (such as 1,000 in TREC), we limit ourselves to only the top ten documents due to the large number of documents that would otherwise have had to be assessed. Essentially, we assume that anything not in the top 10 is unranked.

[6]This percentage varied between experiments.

[7]Computed with respect to the inferior ranking function, that is, $\delta = \frac{a-b}{min(a,b)}$.

these differences is statistically significant at the 95% level according to the bootstrap method described in the following section. Much like the functions tested in Bing, this closeness in retrieval quality is typical when trying to improve an already highly optimized production function, where no single change provides a substantial gain.

4.5. Statistical Methodology

Wherever possible we use bootstrap estimators [Efron and Tibshirani 1993] to evaluate the statistical significance of our findings. Unlike methods based on parametric statistical models, bootstrap methods require fewer assumptions and provide a unified approach to the various statistical inference problems we encountered.

In particular, we use the bootstrap percentile method [Shao and Tu 1995] to compute confidence intervals for point estimates. Given n independently and identically distributed observations $X_1, \ldots, X_n \sim P(X)$, let $\hat{\theta}$ be an estimator (e.g., the sample mean) of some unknown quantity θ (e.g., the population mean). Two statistics $\hat{\theta}_l$ and $\hat{\theta}_u$ of $X_1, \ldots, X_n$ are a $(1-\alpha)$-confidence interval, if $P(\theta \in [\hat{\theta}_l, \hat{\theta}_u]) \geq 1-\alpha$. If not noted otherwise, we use two-sided confidence intervals with equal confidence coefficient $\frac{\alpha}{2}$ in each tail. Our default value for α is 0.05, which results in 95% confidence intervals.

The boostrap percentile method computes the confidence bounds $\hat{\theta}_l$ and $\hat{\theta}_u$ by approximating the population distribution $P(X)$ with the empirical distribution of the observed sample. It then uses the lower and upper $\alpha/2$-percentiles of the empirical distribution as the values of $\hat{\theta}_l$ and $\hat{\theta}_u$. This is a well-justified approximation for large n that is know to provide accurate confidence sets [Shao and Tu 1995].

To compute the percentiles of the empirical distribution, we use the Monte Carlo method. In particular, we draw k bootstrap-samples (i.e., sampling with replacement) of size n from the observed sample and compute $\hat{\theta}$ for each sample. The lower confidence bound θ_l is the $\lfloor \frac{\alpha k}{2} \rfloor$-th lowest value of $\hat{\theta}$, and the upper confidence bound is the $\lceil \frac{\alpha k}{2} \rceil$-th highest value of $\hat{\theta}$. If not noted otherwise, we use $k = 10{,}000$ since larger values of k did not provide substantial improvements in accuracy.

In some of the following experiments we must also measure the reliability of an estimator for sample sizes n' smaller than n. For example, we might want to know whether Team-Draft Interleaving would have already provided confident results with only half the data, that is, $n' = n/2$. It is easy to compute confidence intervals using the bootstrap percentile method simply by using $n' = n/2$ during the Monte Carlo simulation.

Furthermore, we can use the same bootstrap approximation and the Monte Carlo method to directly estimate the consistency of a decison D after n' observations. For example, D may be binary decision indicating whether Team-Draft Interleaving prefers retrieval function A or B after n' observations. Note that D is a random variable, since it depends on a sample. To estimate $P(D|n')$, one can use the fraction of the k Monte-Carlo samples on which D takes any particular value. For clarity, this method is summarized in Algorithm 3.

5. DOES INTERLEAVING AGREE WITH EXPERT ASSESSMENTS?

The first question we address in detail is whether the ranking identified as better by interleaving is also judged as better by human experts. In the case of ArXiv.org, the expert judgment results from the designed degradation of the retrieval functions. On Yahoo! and Bing, we compare the outcome of interleaving evaluations to relevance differences as measured using MAP, DCG and NDCG [Manning et al. 2008]. Note that one should not treat human judgments as the ground truth. Rather, agreement between expert judgments and interleaving builds confidence that both methods accurately reflect user satisfaction with search results (see Section 11 for a detailed discussion).

ALGORITHM 3: Bootstrap Monte Carlo Method for Consistency Estimation.

Input: Data X, decision rule D, bootstrap sample size n'.
$k = 10{,}000$ ..... *Number of repetitions*
$f[] = 0$ ..... *Counter for outcomes*
for $i = 1$ to k **do**
  $X^* =$ sample with replacement of size n' from X.
  $d =$ value of D on sample X^*.
  $f[d] = f[d] + 1$
end for
Return: $\hat{P}(D = d|n') = f[d]/k$

Table II.
Interleaving results for $\mathcal{E}_A \succ \mathcal{B}_A \succ \mathcal{A}_A$ and $\mathcal{E}_A \succ \mathcal{D}_A \succ \mathcal{C}_A$ ranking function triplets on ArXiv.org. The numbers show Δ_{AB} (in percentage) per Equation (4). Bold numbers are statistically significantly greater than zero with 95% confidence.

	$\mathcal{E}_A \succ \mathcal{B}_A$	$\mathcal{B}_A \succ \mathcal{A}_A$	$\mathcal{E}_A \succ \mathcal{A}_A$	$\mathcal{E}_A \succ \mathcal{D}_A$	$\mathcal{D}_A \succ \mathcal{C}_A$	$\mathcal{E}_A \succ \mathcal{C}_A$
Balanced	**4.32**	**2.54**	**5.38**	**1.74**	**3.06**	**3.83**
Team-Draft	**5.23**	**3.52**	**12.92**	2.05	1.96	**4.92**

5.1. Agreement in the Direction of Preference

For ArXiv.org, as described in Section 4.2, five ranking functions were created by intentionally degrading a well performing ranking function. We now evaluate whether the direction of the interleaving preference agrees with this degradation by design. While independent expert judgments were not performed, two of the authors of this paper inspected a sample of the results and agreed that retrieval quality was degraded as expected. These results were previously published in Radlinski et al. [2008].

Table II shows that interleaving agrees with the design on all 6 pairwise comparisons. For both interleaving methods, the sign of Δ_{AB} (as defined in Equation (4)) is always positive, reflecting the expected ordering in both $\mathcal{E}_A \succ \mathcal{B}_A \succ \mathcal{A}_A$ and $\mathcal{E}_A \succ \mathcal{D}_A \succ \mathcal{C}_A$. Computing confidence intervals for Δ_{AB} as described in Section 4.5, the bold entries in Table II indicate where the expected difference in wins/losses is significantly different from zero. While the remaining two pairs fail the 95% confidence level, they are significant at the 90% level.

For Bing, we compare the preferences from interleaving with conventional expert judgments. For each interleaved pair of ranking functions, Figure 2 shows absolute difference in NDCG@5 and MAP versus the value of Δ_{AB} using Team-Draft Interleaving. These results were previously published in [Radlinski and Craswell 2010]. The error bars indicate 95% confidence intervals for expert judgments and interleaving computed from $k = 1{,}000$ bootstrap samples. Whenever there is a significant difference in either NDCG or MAP (i.e., the vertical error bars do not intersect with zero), interleaving agrees with the direction of preference. Furthermore, interleaving is significant for all pairs. Note that the apparent disagreement on $\mathcal{D}_B \succ \mathcal{C}_B$ simply indicates that according to expert judgments ranker $\mathcal{D}_B$ is slightly better but not statistically significantly, while interleaving finds ranker $\mathcal{C}_B$ significantly preferred. In other words, interleaving finds a significant preference while expert judgments do not.

The experiments on Yahoo! are analogous to those for Bing, but use Balanced Interleaving instead of Team-Draft Interleaving. For each interleaved pair of Yahoo ranking functions, Figure 3 shows the relative[8] DCG@5 difference between the ranking functions interleaved versus Δ_{AB} using Balanced interleaving. All points lie in the positive quadrant. This means that the direction of the preference agrees with expert

[8]All relative differences δ are computed with respect to the inferior ranking function, that is, $\delta = \frac{a-b}{min(a,b)}$.

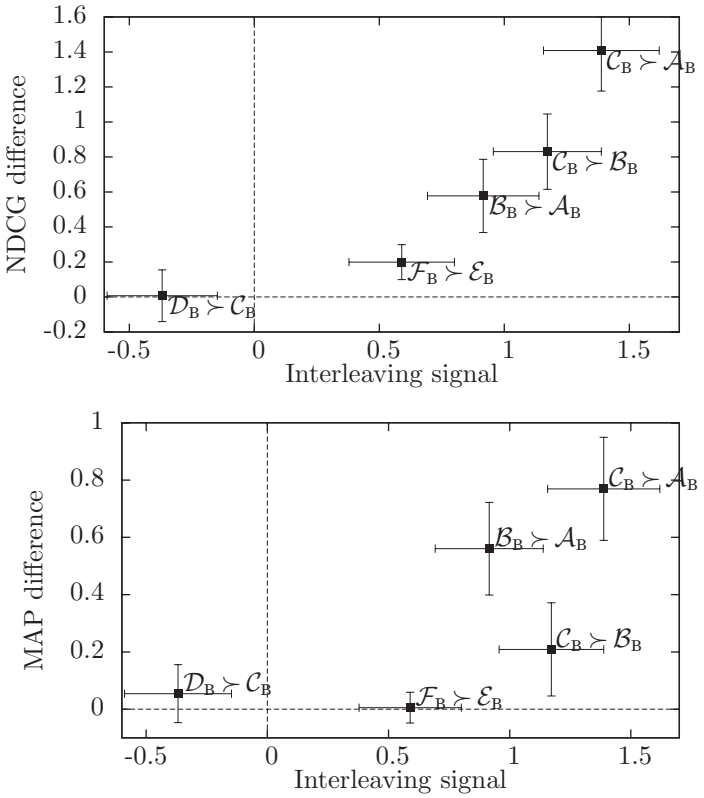

Fig. 2. Agreement between expert judgments and interleaving on the Bing search engine. The interleaving signal is the Δ_{AB} quantity defined in Equation (4). The bars around each point indicate confidence intervals computed by bootstrap.

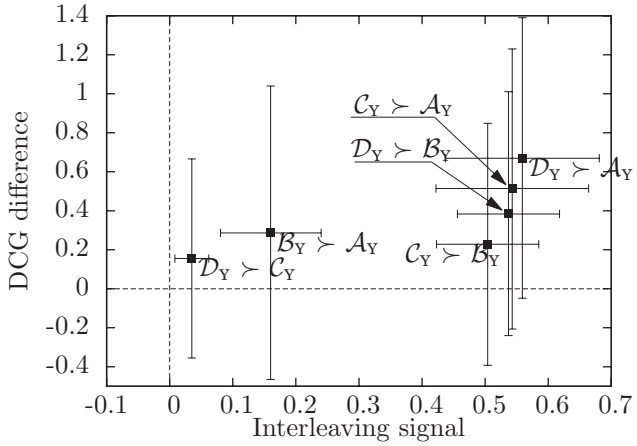

Fig. 3. Agreement between expert judgments and interleaving on Yahoo! search, similarly to Figure 2.

judgments in all cases. However, while the interleaving signal is significant for all pairs, the confidence intervals for DCG@5 difference are large and not significant (because only 2,000 judged queries are available).

Table III.
Correlation between expert judgments and interleaving on the Bing and Yahoo search engine, computed across all pairs interleaved on each system. Bootstrapped correlation involved resampling from the judged queries and the interleaved impressions, and then measuring correlation.

Setting	Interleaving	Metric	Correlation	Bootstrapped correlation
Bing	Team-Draft	NDCG@5	0.882	0.864 ± 0.048
	Team-Draft	MAP	0.689	0.669 ± 0.104
Yahoo!	Balanced	DCG@5	0.696	0.417 ± 0.419

Table IV.
Summarizing the interleaving signals Δ (in percentage) for all triplets $A \succ B \succ C$ using the full unsampled data. In all cases, they satisfy strong stochastic transitivity, $\Delta_{AC} > \max\{\Delta_{AB}, \Delta_{BC}\}$.

Ranking function			Interleaving signal		
A	B	C	Δ_{AC}	Δ_{AB}	Δ_{BC}
$\mathcal{E}_{\text{A}}$	$\mathcal{B}_{\text{A}}$	$\mathcal{A}_{\text{A}}$	5.38	4.32	2.54
$\mathcal{E}_{\text{A}}$	$\mathcal{D}_{\text{A}}$	$\mathcal{C}_{\text{A}}$	3.83	1.74	3.06
$\mathcal{C}_{\text{B}}$	$\mathcal{B}_{\text{B}}$	$\mathcal{A}_{\text{B}}$	1.38	1.17	0.92
$\mathcal{C}_{\text{Y}}$	$\mathcal{B}_{\text{Y}}$	$\mathcal{A}_{\text{Y}}$	0.54	0.50	0.16
$\mathcal{D}_{\text{Y}}$	$\mathcal{B}_{\text{Y}}$	$\mathcal{A}_{\text{Y}}$	0.56	0.54	0.16
$\mathcal{D}_{\text{Y}}$	$\mathcal{C}_{\text{Y}}$	$\mathcal{A}_{\text{Y}}$	0.56	0.03	0.54
$\mathcal{D}_{\text{Y}}$	$\mathcal{C}_{\text{Y}}$	$\mathcal{B}_{\text{Y}}$	0.54	0.03	0.50

5.2. Correlation of the Magnitude of Difference

The previous section showed that the direction of preference predicted by interleaving agreed with the direction of preference derived from expert relevance judgments whenever the latter was significant. In this section we go a step further by analyzing the agreement of the magnitudes of these preferences signals. More precisely, how do the differences in editorial metric correlate with the interleaving signal?

Figures 2 and 3 show that, for the Bing experiments, NDCG@5 is highly correlated with interleaving, while MAP is somewhat less correlated. For the Yahoo! experiments, DCG@5 also appears to be well correlated with interleaving, although the relatively small number of queries for editorial judgments (2,000) induces large error bars on the DCG differences.

The correlations corresponding to these plots are shown in Table III. The bootstrap correlation is computed by taking bootstrap samples of queries from both the expert-judgment corpus as well as the interleaving buckets. For each iteration of the Monte Carlo algorithm, the correlation is computed. The mean and standard deviation of these correlations is then reported in the last column of Table III.

A similar study was conducted on a smaller scale by Ali and Chang [2006]. In that study, a correlation between interleaving and side-by-side relevance assessments was computed across 89 queries. The authors reported a correlation of 0.40, which is substantially smaller than the correlations shown in Table III. This can be explained by the various differences in experimental design—most notably due to the fact that the correlation in Ali and Chang [2006] was computed at the query level. Query-level measurements tend to be substantially more noisy and less reliable than the aggregate measures considered in this article.

5.3. Internal Consistency of Interleaving Preferences

Using the results presented in this section, we can also analyze the internal consistency of interleaving preferences. In particular, Table IV shows that a magnitude-preserving

property called strong stochastic transitivity [Kozielecki 1981] is satisfied for all interleaving preferences evaluated thus far in this section. This property implies that for any triplet $A \succ B \succ C$, we have

$$\Delta_{AC} \geq \max\{\Delta_{AB}, \Delta_{BC}\}. \tag{7}$$

Properties such as strong stochastic transitivity provide internal consistency, or structure, that allows for efficient selection of the best amongst a set of functions (e.g., as formulated in the Dueling Bandits Problem [Yue et al. 2009; Yue and Joachims 2011]).

5.4. Summary

We found that both interleaving algorithms reliably agree with judgments collected from experts whenever both outcomes are statistically significant. Furthermore, the magnitude of Δ_{AB} shows substantial correlation with the magnitude of NDCG difference, and somewhat less with DCG and MAP difference. Finally, all the values of Δ_{AB} exhibit strong stochastic transitivity.

6. DO ABSOLUTE METRICS AND INTERLEAVED EVALUATION AGREE?

Noncomparative experiments are the conventional approach for estimating the quality of a ranking function based on implicit user feedback (see Section 2 for a discussion of relevant related work). Each ranking function is fielded on a different bucket of users, and absolute metrics are estimated for each bucket. The estimated values are then used to order the ranking functions, assuming that the metrics change monotonically with retrieval quality. We now explore several absolute metrics that quantify the clicking and session behavior of users, and verify whether they do indeed order ranking functions by their retrieval quality. Furthermore, we investigate the agreement of absolute metrics with the preferences elicited via interleaving, which we have already established to agree with expert judgments.

6.1. Absolute Metrics

We measured the following ten absolute metrics, including the metrics used in Radlinski et al. [2008]. They reflect the key observable 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 average number of queries issued by a user during a session.
Clicks per Query	The average number of results that are clicked for each query.
Clicks@1	The fraction of queries for which there was a click on the top-ranked document.
pSkip†	The average of one minus the number of clicked documents divided by the rank of the lowest click (i.e. the fraction of documents viewed but not clicked) [Wang et al. 2009].
Max Reciprocal Rank†	The average value of $1/r$, where r is the rank of the highest ranked result clicked on.
Mean Reciprocal Rank†	The average value of the mean of $1/r_i$ over the ranks r_i of all clicks for each query.
Time to First Click†	The median time from query being issued until first click on any result.
Time to Last Click†	The median time from query being issued until last click on any result.

Table V. Hypothesized Change of the Absolute Metrics with Decreasing Retrieval Quality

Metric	Hypothesis	Rationale
Abandonment rate	↗	(more bad result sets)
Reformulation rate	↗	(more need to reformulate)
Queries per session	↗	(more need to reformulate)
Clicks per query	↘	(fewer relevant results)
Clicks@1	↘	(top result is worse)
pSkip	↗	(more skipped results)
Max reciprocal rank	↘	(top results are worse)
Mean reciprocal rank	↘	(more need for many clicks)
Time to first click	↗	(good results are lower)
Time to last click	↘	(fewer relevant results)

Table VI.
Results of the absolute metrics comparison on the ArXiv data. The numbers are the relative difference (in percent) of the absolute metric between the two functions to be compared. The signs are set such that a positive number means that the comparison outcome is in agreement with the hypothesized change in Table V.

	$\mathcal{E}_A \succ \mathcal{B}_A$	$\mathcal{B}_A \succ \mathcal{A}_A$	$\mathcal{E}_A \succ \mathcal{A}_A$	$\mathcal{E}_A \succ \mathcal{D}_A$	$\mathcal{D}_A \succ \mathcal{C}_A$	$\mathcal{E}_A \succ \mathcal{C}_A$
Abandonment Rate	6.22	0.14	6.35	−3.56	2.61	−0.86
Reformulation Rate	3.86	1.30	5.12	0.82	−0.84	−0.01
Queries per Session	1.90	1.86	3.73	−0.70	−3.90	−4.62
Clicks per Query	28.26	4.16	33.59	−5.17	3.49	−1.86
Clicks@1	36.01	−2.93	32.02	−11.77	35.65	19.68
pSkip	2.28	3.00	5.22	−2.38	12.51	10.43
Max Reciprocal Rank	6.48	0.48	6.99	−3.80	14.69	10.33
Mean Reciprocal Rank	3.67	0.51	4.20	−4.86	18.45	12.69
Time to First Click	−3.33	6.25	3.12	−0.00	12.50	12.50
Time to Last Click	4.59	−5.22	−0.87	24.04	−18.11	1.57
Balanced Interleaving Δ_{AB}	4.32	2.54	5.38	1.74	3.06	3.83
Team-Draft Interleaving Δ_{AB}	5.23	3.52	12.92	2.05	1.96	4.92

A session is defined 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 computing the metrics marked with †, we exclude queries with no clicks to avoid conflating this metric with abandonment rate. For each metric, we hypothesize in Table V how we expect the metric to change as retrieval quality decreases. Even if the hypothesized directions of change are incorrect, we at least expect these metrics to change in a consistent direction as retrieval quality decreases.[9]

6.2. Results

For ArXiv.org, Table VI lists the relative differences of the absolute metrics for all pairwise comparisons in $\mathcal{E}_A \succ \mathcal{B}_A \succ \mathcal{A}_A$ and in $\mathcal{E}_A \succ \mathcal{D}_A \succ \mathcal{C}_A$. The absolute values of the respective metrics can be found in Radlinski et al. [2008], with the exception of Clicks@1 and pSkip. A positive number in Table VI indicates that the change in the absolute metric is consistent with the hypothesis of Table V. For example, the value of 6.22 in the top left corner of Table VI means that the abandonment rate of $\mathcal{E}_A$ is lower than that of $\mathcal{B}_A$ by 6.22%. Note again that the relative difference is always computed with respect to the inferior ranking function.

We observe 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 VII. It shows that, for example, the abandonment rate agrees with the hypothesis for four pairs of ranking

[9]Although these metrics can be very noisy at the query-level, we should expect them to change consistently with (average) retrieval quality when aggregrated over a large sample of queries.

Table VII.
Comparing the number of correct ("✓") and false ("↯") preferences implied by an absolute metric or interleaving, aggregated over the "$\mathcal{E}_A \succ \mathcal{B}_A \succ \mathcal{A}_A$" and the "$\mathcal{E}_A \succ \mathcal{D}_A \succ \mathcal{C}_A$" comparisons on ArXiv. 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		significant	
	✓	↯	✓	↯
Abandonment Rate	4	2	2	0
Reformulation Rate	4	2	0	0
Queries per Session	3	3	0	0
Clicks per Query	4	2	2	0
Clicks@1	4	2	4	0
pSkip	5	1	2	0
Max Reciprocal Rank	5	1	3	0
Mean Reciprocal Rank	5	1	2	0
Time to First Click	4	1	0	0
Time to Last Click	3	3	1	0
Balanced Interleaving	6	0	6	0
Team-Draft Interleaving	6	0	4	0

functions ($\mathcal{E}_A \succ \mathcal{B}_A$, $\mathcal{B}_A \succ \mathcal{A}_A$, $\mathcal{E}_A \succ \mathcal{A}_A$ and $\mathcal{D}_A \succ \mathcal{C}_A$). 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 VII are estimates of a population mean/median. The column "significant" of Table VII shows for how many pairs $A \succ B$ the difference in the hypothesized direction of change (✓) or its opposite (↯) is significant based on the 95% bootstrap confidence intervals. We do not see a significant difference for more than four out of the six pairs $A \succ B$ for any of the absolute metrics. With the exception of Max Reciprocal Rank and Clicks@1, even the "large difference" pairs $\mathcal{E}_A \succ \mathcal{A}_A$ and $\mathcal{E}_A \succ \mathcal{C}_A$ are not consistently significant for any of the metrics. This suggests that, at best, substantially more data is needed to use these absolute metrics reliably, making them unsuitable for low-volume search applications like desktop search, digital library search, and intranet search.

We performed a similar analysis on Yahoo!, but using orders of magnitude more data. Recall from Table I that each of the Yahoo! functions was evaluated using tens of millions of queries. We drop the session-based metrics (reformulation rate, and queries per session) since they did not appear to be predictive of ranking quality in previous experiments.

The results are listed in Table VIII. Only one of the absolute metrics, Clicks@1, always agrees with the ranking of the retrieval functions by DCG@5. Furthermore, if we consider correlation with DCG@5 as a measure of quality of these metrics, the highest correlation achieved by any absolute metric is 0.43 (again, by Clicks@1). Table IX summarizes the performance of these absolute metrics. Most of the differences are statistically significant at the 95% confidence level, but some still lack significance even after millions of queries.

The number of clicks per query does not seem to follow any meaningful pattern, which was already noted in Chapelle et al. [2009]. This can be explained as follows. On the one hand, one would expect fewer clicks per query if the results are less relevant. This is probably true for informational queries. On the other hand, for navigational queries, when the navigational result is not at the top of the ranking, the user may

Table VIII.
Results of absolute metric comparison on Yahoo! dataset. The first row is the DCG5 averaged over a set of 2,000 queries. The next 8 rows are the relative difference (%) between each metric for each pair of ranking functions. The last row is the deviation from 50% for interleaving. For all metrics, a positive number indicates that the second function is to preferred to the first one. The last column is the correlation coefficient between a given metric and the DCG, computed across the 6 pairs of functions.

	$\mathcal{B}_Y \succ \mathcal{A}_Y$	$\mathcal{C}_Y \succ \mathcal{A}_Y$	$\mathcal{D}_Y \succ \mathcal{A}_Y$	$\mathcal{C}_Y \succ \mathcal{B}_Y$	$\mathcal{D}_Y \succ \mathcal{B}_Y$	$\mathcal{D}_Y \succ \mathcal{C}_Y$	Corr.
DCG5	0.27	0.50	0.65	0.23	0.38	0.16	1
Abandonment Rate	−0.214	0.084	0.228	0.298	0.441	0.144	0.16
Clicks per Query	−0.365	−0.033	−0.097	0.334	0.269	−0.064	−0.11
Clicks@1	0.049	0.245	0.844	0.196	0.795	0.597	0.43
pSkip	−0.039	0.140	0.465	0.180	0.504	0.326	0.38
Max Reciprocal Rank	−0.069	0.155	0.527	0.224	0.596	0.371	0.35
Mean Reciprocal Rank	−0.014	0.178	0.599	0.192	0.613	0.420	0.40
Time to First Click	0.646	0.245	0.381	−0.403	−0.266	0.137	0.28
Time to Last Click	−0.725	−0.282	−0.233	0.446	0.496	0.050	−0.23
Balanced Interleaving Δ_{AB}	0.160	0.543	0.559	0.504	0.537	0.035	**0.69**

Table IX.
Similarly to Table VII, number of correct and false preferences on the Yahoo! data.

	weak		significant	
	✓	↯	✓	↯
Abandonment Rate	5	1	5	1
Clicks per Query	2	4	2	3
Clicks@1	6	0	5	0
pSkip	5	1	5	0
Max Reciprocal Rank	5	1	5	0
Mean Reciprocal Rank	5	1	5	0
Time to First Click	4	2	3	1
Time to Last Click	3	3	2	3
Balanced Interleaving	6	0	6	0

click on several results before finding the navigational result. Thus, this metric could both increase or decrease as the ranking function deteriorates. A similar argument could be made for time to last click: a time increase can either be due to more relevant results (user engaged) or worse results (user struggles to find what he needs).

Interleaving correctly predicts the preference for each function pair, and the interleaving signal Δ_{AB} shows a 0.69 correlation with the difference in DCG, as already noted in Table III.

6.3. Summary

For all absolute metrics, most differences are not statistically significant on the ArXiv dataset, indicating a substantially lower sensitivity than interleaving. While the absolute metrics become mostly significant on the Yahoo! data, even the best absolute metrics (i.e., Clicks@1, pSkip, Max Reciprocal Rank, and Mean Reciprocal Rank) are substantially less correlated with DCG@5 than interleaving.

7. HOW MUCH CLICK DATA IS NEEDED TO OBTAIN A STATISTICALLY RELIABLE PREFERENCE?

The results in the previous section have already indicated that different metrics differ in sensitivity and reliability. The more sensitive an implicit feedback signal, the less data is needed and the faster one can act upon the evaluation results. Before comparing the sensitivity of interleaving to that of human judgments in Section 8, we first investigate the sensitivity of interleaving in comparison to the absolute metrics.

For this purpose, we used Algorithm 3 from Section 4.5 to estimate how confidently a given metric predicts a preference between two retrieval functions given n' queries. Here, n' is typically much smaller than the size of the full dataset X. In particular, to estimate the sensitivity of interleaving for ranking functions $\mathcal{A}$ and $\mathcal{B}$, we repeatedly draw bootstrap samples X^*_{AB} of n' queries from the overall dataset X_{AB}, compute the preference on each X^*_{AB} according to interleaving, and count the fraction of times the preference goes each direction. The respective fraction is an estimate of $\Pr(\mathcal{A} \succ_m \mathcal{B}|n')$, the probability that metric m will prefer $\mathcal{A}$ over $\mathcal{B}$ after n' queries. For the absolute metrics, we use an analogous resampling procedure that draws two bootstrap samples X^*_A and X^*_B, each of size n', from the corresponding noncomparative experiment.

7.1. Results

The results for the six pairs of ArXiv.org retrieval functions are plotted in Figure 4. The x-axis indicates the number of queries n, while the y-axis shows the probability that a metric disagrees with the true preference. The metrics for which the probability is increasing as a function of the data size are the ones which do not agree with the true preference. Note that at the extreme right of these plots (i.e., resampling the data at the same size as the original size), a value under 0.05 corresponds to a statistically significant agreement (with 95% confidence) in Tables II and VII.

In most cases, the interleaving methods reach a high level of confidence much faster than conventional methods based on absolute metrics. This difference is most notable for the "large-difference" pairs $\mathcal{E}_{\text{A}} \succ \mathcal{A}_{\text{A}}$ and $\mathcal{E}_{\text{A}} \succ \mathcal{C}_{\text{A}}$, where less than a third of the data would have sufficed to state a confident preference. The two interleaving methods show similar sensitivity and no method emerges as a clear winner.

Figure 5 shows analogous plots for the Yahoo! data, but using a log-scale on the x-axis due to the large amount of data. In addition, since different buckets vary greatly in the number of queries (see Table I), the sizes of the resampled data were absolute and not a fraction of the original data as in Figure 4. The plots show the probability that each absolute metric or interleaving disagrees with the DCG@5 difference resulting from expert judgments. As expected (and consistent with the positive numbers in the last row of Table VIII), this probability drops to zero for interleaving on all pairs of retrieval functions. Furthermore, interleaving typically produces a statistically reliable preference after much less data than any absolute metric.[10]

We now quantify how much data interleaving saves in comparison to the absolute metrics. We focus on comparing interleaving with Clicks@1, since Clicks@1 is the only absolute metric that predicts the correct preference for all buckets. Furthermore, Clicks@1 has the highest correlation with DCG@5 among the absolute metrics (see Table VIII). Table X describes our results. There is only one pair for which interleaving was less sensitive than Clicks@1. In all others, interleaving was much more sensitive, significantly reducing the number of queries needed by over an order of magnitude.

From an application perspective, the quantity of interest is the evaluation duration required to produce reliable results. Note that this is not necessarily equivalent to the number of queries as considered above, since a temporally consecutive sequence of queries may not be an independent and identically distributed sample. We therefore verify the above results using a variant of Algorithm 3 that uses a different resampling protocol. In particular, we use the sequence-based resampling protocol described in Algorithm 4 on ranking functions $\mathcal{B}_{\text{Y}}$ and $\mathcal{D}_{\text{Y}}$. We chose these two functions due to their relatively large difference in relevance and ample amount of logged queries, both in the interleaving and noncomparative buckets. This estimator no longer draws bootstrap

[10]Note that, for a given x value on these plots, an absolute comparison uses twice as much data as an interleaved comparison, since two non-comparative buckets need to be run for absolute metrics.

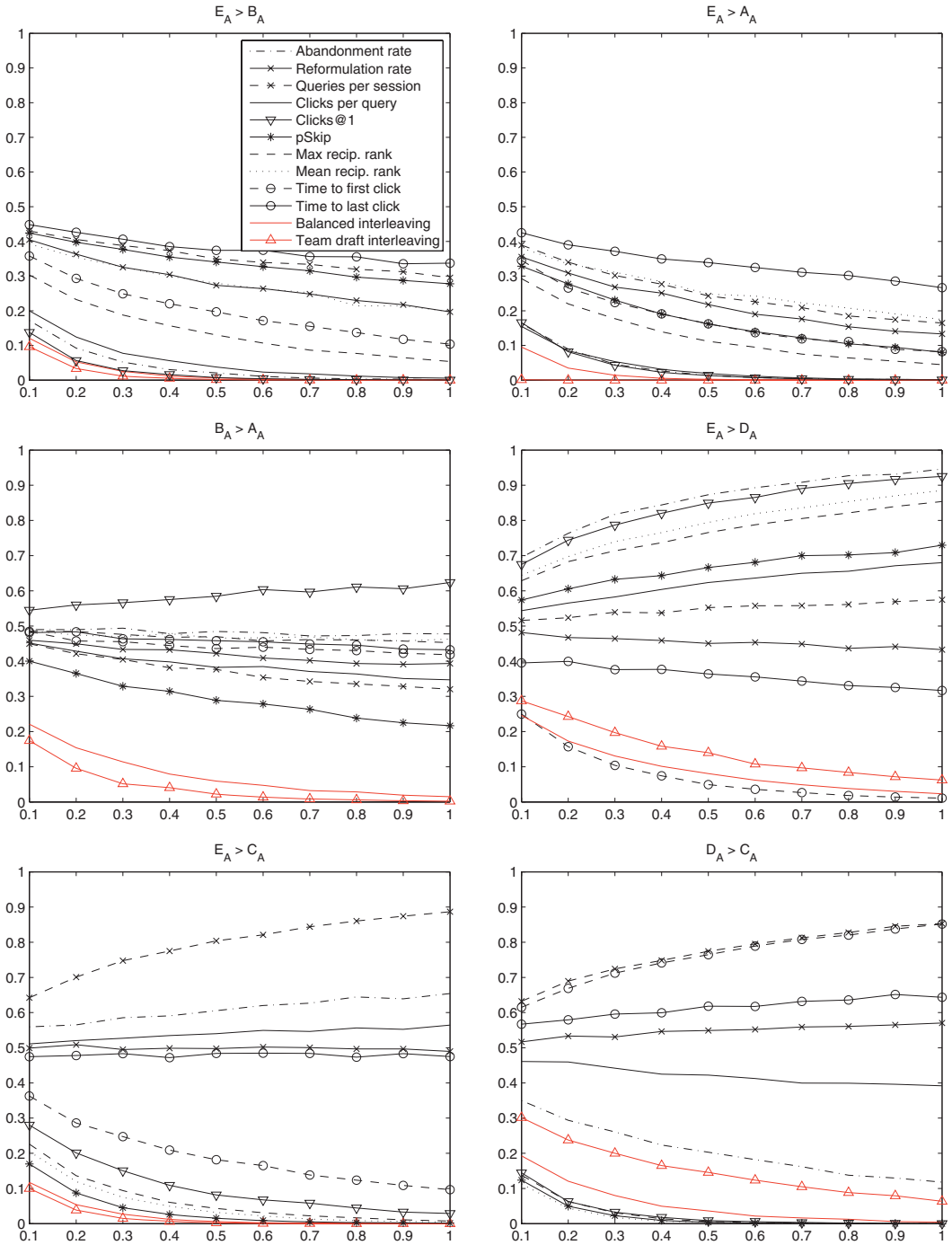

Fig. 4. Probability (as approximated via Algorithm 3) that evaluation based on implicit feedback disagrees with the true preference, plotted individually for the six preference pairs of $\mathcal{E}_A \succ \mathcal{B}_A \succ \mathcal{A}_A$ and $\mathcal{E}_A \succ \mathcal{D}_A \succ \mathcal{C}_A$. The x-axis is the amount of data relative to the size of the original data.

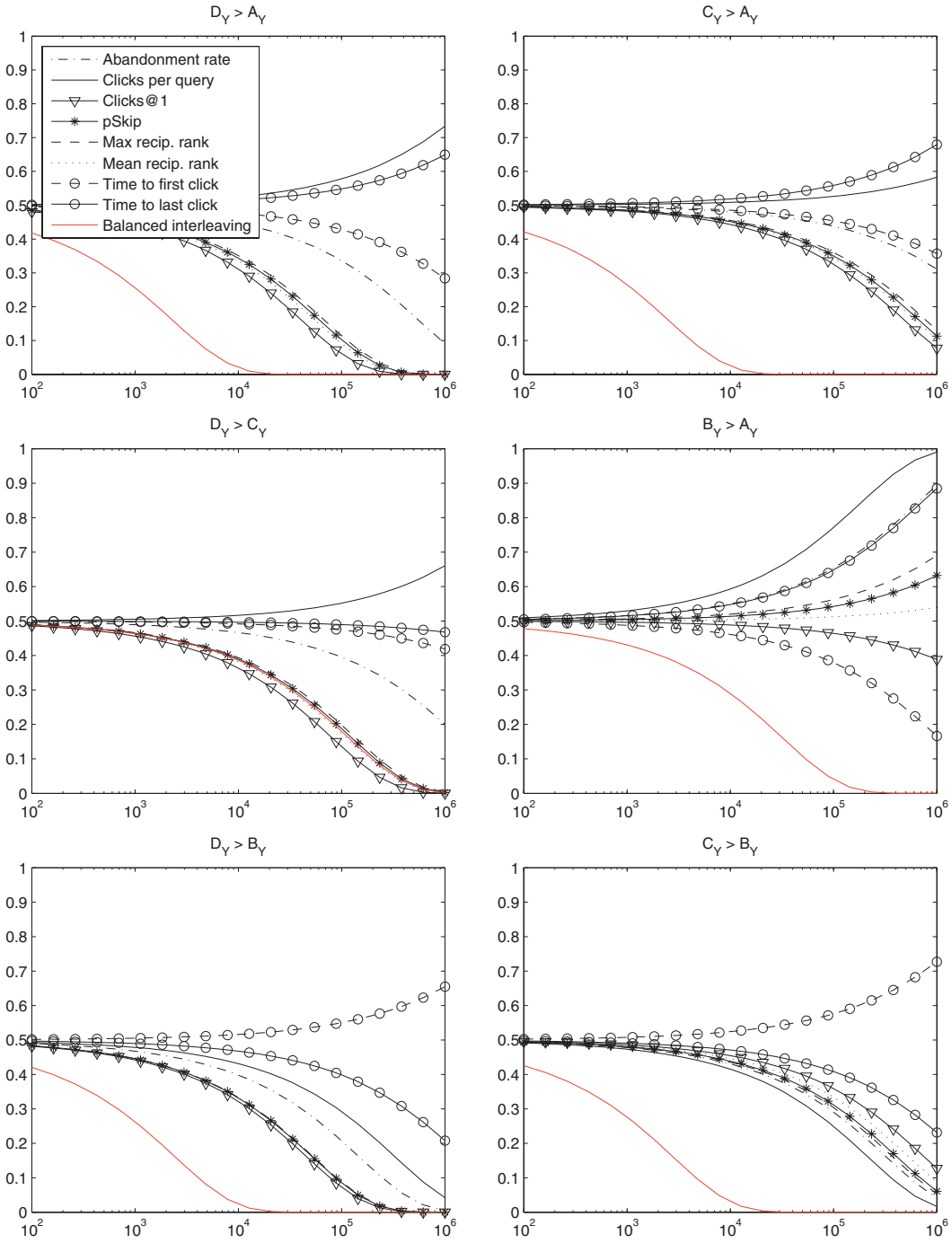

Fig. 5. Analogous to Figure 4, but on the Yahoo! search data. The x-axis is the number of user queries used.

ALGORITHM 4: Bootstrap Monte Carlo Method Using Resampling of Sequences.

Require: Data X, time span τ
1: $f = 0; g = 0$
2: **for** $i = 1 \ldots 100$ **do**
3: Subsample uniformly non-comparative buckets $\mathcal{B}_Y$ and $\mathcal{D}_Y$ such that they have the same query rate as the interleaved bucket $\mathcal{D}_Y \succ \mathcal{B}_Y$.
4: **for** $t = 1 \ldots (T_{max} - \tau)$ **do**
5: Get all data from the 3 buckets between time t and $t + \tau$.
6: **if** $\mathcal{B}_Y$ wins according to interleaving **then** $f = f + 1$
7: **if** $\mathcal{B}_Y$ wins according to Clicks@1 in noncomparative buckets **then** $g = g + 1$
8: **end for**
9: **end for**
10: $P(\mathcal{B}_Y \succ_{inter} \mathcal{D}_Y | \tau) = f/(100(T_{max} - \tau))$.
11: $P(\mathcal{B}_Y \succ_{C@1} \mathcal{D}_Y | \tau) = g/(100(T_{max} - \tau))$.

Table X.
Results comparing ratio of data required by Clicks@1 versus balanced interleaving for the Yahoo! dataset. We first computed $\Pr(\mathcal{A} \succ_{C@1} \mathcal{B}|n')$ for Clicks@1 by resampling n' queries from both buckets, where n' is the number of queries in the smaller of the two buckets. We then calculated the number of queries n'' for interleaving to reach the same preference probability. Finally, we report the ratio n'/n'' of the two data set sizes. Values larger than 1 indicate that interleaving requires less data relative to Clicks@1.

	$\mathcal{B}_Y \succ \mathcal{A}_Y$	$\mathcal{C}_Y \succ \mathcal{A}_Y$	$\mathcal{D}_Y \succ \mathcal{A}_Y$	$\mathcal{C}_Y \succ \mathcal{B}_Y$	$\mathcal{D}_Y \succ \mathcal{B}_Y$	$\mathcal{D}_Y \succ \mathcal{C}_Y$
Data Ratio n'/n''	379.1	193.1	17.8	270.7	18.9	0.69

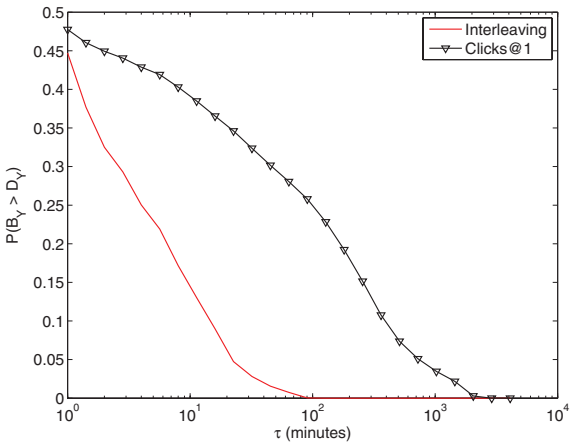

Fig. 6. Probability that Clicks@1 and Balanced Interleaving identify an inconsistent preference between $\mathcal{B}_Y$ and $\mathcal{D}_Y$ after a given duration of data collection. The methodology is summarized in algorithm 4.

samples with replacement, but samples entire sequences of queries as they were issued in the respective Yahoo! buckets. In particular, it first subsamples the noncomparative buckets to yield the same quantity of data as the interleaving bucket. The algorithm then slides a window of length τ over the time series and computes the fraction of times the preference prediction is consistent with the full dataset. The results are shown in Figure 6. The interleaving bucket requires around one hour to reach an inconsistency under 0.01, whereas the non-comparative buckets require slightly more than one day. The ratio in efficiency is about 22, which is in line with the 18.9 reported in the Table VI.

More generally, we observe no substantial change in conclusions drawn from this time based analysis and from the standard bootstrap (i.e., Figure 5).

7.2. Summary

We find that the interleaving methodology is very sensitive and can reveal small differences in retrieval quality with relatively small query samples. In comparison, absolute metrics typically need orders of magnitude more data to confidently detect any difference (in either the correct or the incorrect direction). We conjecture that the advantages of interleaving over absolute measurements are due to the following two differences in experiment design. First, interleaving appears to have increased sensitivity because it is a paired test, paired on both queries and users. Second, by directly eliciting a preference between two alternatives, interleaving appears to more directly and reliably measure differences in relevance.

8. WHAT IS THE VALUE OF A CLICK RELATIVE TO A JUDGED QUERY?

For conventional evaluation using manually judged queries, information retrieval researchers have a good understanding of the number of manual judgments required for reliable evaluation [Voorhees and Buckley 2002; Sanderson and Zobel 2005]. Analogous to this question, we now assess how many clicks are required for interleaving to produce results comparable in statistical reliability to those from manual judgments. In particular, we investigate how many clicks are needed to replace one manually judged query. The Bing results in this section extend the analysis from Radlinski and Craswell [2010].

8.1. Results

To estimate the relative value of a click in interleaving compared to a manual relevance judgment, we use the following procedure. Following Algorithm 3, we start with the set of about 12,000 judged queries on the Bing search engine. From this set, we subsample n' queries (with replacement) and measure NDCG@5 for each ranker on this sample, repeating the sampling $k = 1{,}000$ times for each n'. We then count the fraction of bootstrap samples where each ranker scored higher, ignoring cases when the scores were identical. The result is an estimate of the probability p that NDCG@5 prefers one ranking function over the other after n' queries. Using the resampling procedure that produced Figures 4 and 5, we now find the number of impressions n'' necessary to obtain the same preference probability p for interleaving. Figure 7 plots n' vs. n''.

More specifically, the top plot of Figure 7 considers NDCG@5 as the conventional evaluation metric, and the bottom plot considers MAP. Each curve corresponds to a pair of ranking functions, and each point shows the number of judged queries n' vs. the number of interleaved queries n'' to achieve a particular level of consistency p. The pairs of rankers for which the direction of change of NDCG or MAP is not statistically significant with 95% confidence using 10,000 queries are omitted.

The relationship between judged queries and interleaving impressions is approximately linear in all cases, with some variations depending on the pair of rankers being compared. On average, one judged query is approximately equivalent to ten clicked queries using interleaving. Further breaking the analysis down from queries to documents, manually judging one query corresponds to judging five to ten documents for NDCG@5 (depending on whether the input rankers agree on the top results). As such, one judged (query, document) pair appears to provide as much evaluation sensitivity as one or two queries with clicks. As can be seen in the lower plot of Figure 7, when evaluating with MAP even more queries need to be judged by expert judges relative to the number of clicked queries necessary.

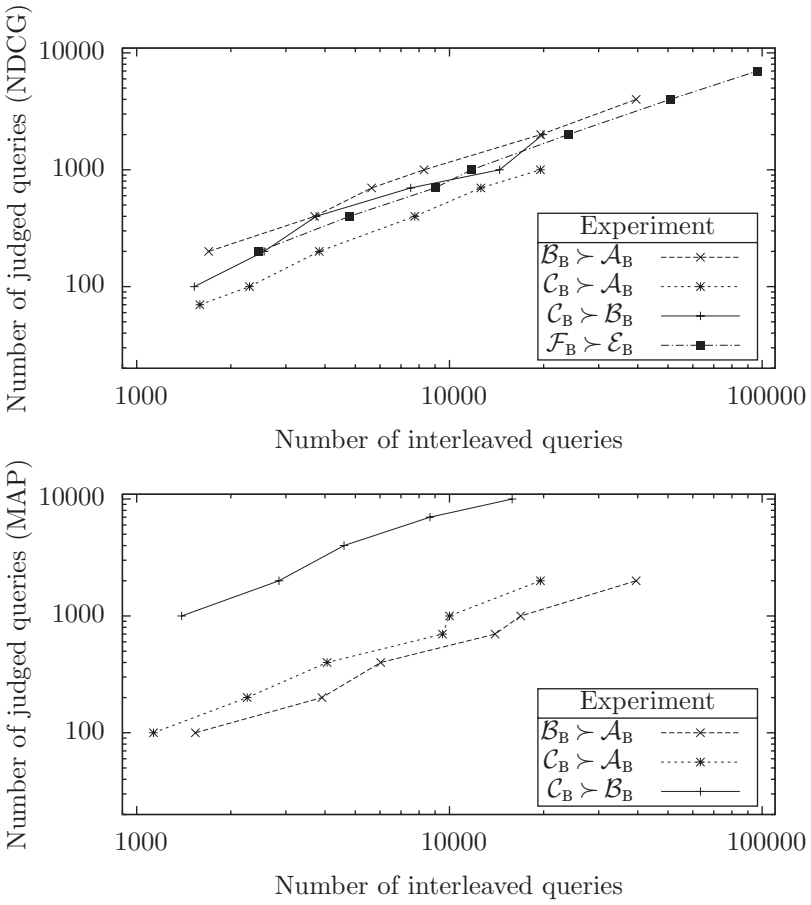

Fig. 7. Number of judged queries versus number queries with clicks necessary with interleaving to obtain the consistency p on Bing, using NDCG@5 (top plot) and MAP (bottom plot).

The same evaluation performed on Yahoo! shows a correspondence between judged and interleaved queries that is consistent with the Bing results. However, the number of queries judged on the Yahoo corpus is substantially smaller, hence the results are very noisy. Therefore, we do not include the corresponding plot.

8.2. Summary

We find that approximately ten interleaved queries with clicks provide equivalent evaluation power as compared to one manually judged query. Assuming that each manually judged query requires at least five document judgments, then feedback from two interleaved queries corresponds to at least one judged document. It is unclear, however, how consistent these relationships are across different document collections and user populations.

9. HOW SENSITIVE IS INTERLEAVING TO DIFFERENT CLICK AGGREGATION SCHEMES?

Thus far we have used the simple scheme in Equation (4) to convert clicks into a preference between two interleaved rankings. In general, the ranking that gets more clicks wins the comparison, and the ranking function with more wins over a sample of queries is preferred overall. This basic scoring scheme ignores multiple aspects,

for example, that some clicks may be more informative than others and that some wins may be more definitive than others. In this section we explore a number of more sophisticated click credit assignment and aggregation strategies. Our goal is to evaluate the sensitivity of these methods for correctly identifying the superior ranking function.

9.1. Click Aggregation Strategies

Generalizing the notation from Section 3, let $\Delta(Q)$ denote any aggregate statistic of a credit assignment strategy computed over an empirical sample of query logs Q. Each logged query $(q, C_a, C_b) \in Q$ contains the clicks, C_a and C_b, credited to the two retrieval functions. We only consider queries which have at least one logged click, that is, $\forall (q, C_a, C_b) \in Q, |C_a \cup C_b| > 0$. In each of the methods described below, the sign of $\Delta(Q)$ indicates the direction of preference. We evaluate the following aggregation strategies.

Binary (Original). For each query, we call the binary scoring rule the one which assigns full credit to the ranking that receives more clicks,

$$\Delta_{bin}(Q) = \sum_{(q,C_a,C_b)\in Q} sign(|C_a| - |C_b|), \tag{8}$$

which is equivalent to Equation (4), since the two always have the same sign.

Click. For each query, the click scoring rule computes the difference in click counts for the two rankings,

$$\Delta_{click}(Q) = \sum_{(q,C_a,C_b)\in Q} |C_a| - |C_b|. \tag{9}$$

This assumes that each click contributes equal credit, rather than each query.

Normalized Click. For each query, the normalized click scoring rule computes the difference in click counts for the two rankings normalized by the total number of clicks for that query,

$$\Delta_{norm}(Q) = \sum_{(q,C_a,C_b)\in Q} \frac{|C_a| - |C_b|}{|C_a \cup C_b|}. \tag{10}$$

This assumes that each query contributes an equal amount of total credit (regardless of the number of clicks for the query). However, unlike the binary scoring rule, the credit of any query is split proportional to the number of clicks that each ranking received.

Balanced Interleaving-specific Scoring Strategies. In the case of Balanced Interleaving, we consider three additional scoring rules that result from using an alternative method to crediting clicks to each retrieval function. Recall from Section 3.1 that, for any query, the clicks credited to each retrieval function, C_a and C_b, are those that are above some rank threshold k (see Equation (1)) in original rankings A and B. One can alternatively credit each click to the original ranking, either A or B, that ranked the clicked document higher (or to both if there is a tie). This leads to the following "click-direct" credit assignment methods: for each logged query $(q, C_a, C_b) \in Q$, each click is assigned to C_a^d or C_b^d depending on which original ranking ranked the clicked result higher (the d superscript is used to denote the click-direct sets of clicks).

Binary-Direct. This is exactly the binary scoring rule defined above except using the click-direct entries, (q, C_a^d, C_d^d).

Click-Direct. This is exactly the click scoring rule defined above except using the click-direct entries, (q, C_a^d, C_d^d).

Normalized Click-Direct. This is exactly the normalized click scoring rule defined above except using the click-direct entries, (q, C_a^d, C_d^d).

Team-Draft Interleaving-specific Scoring Strategies. In the case of Team-Draft Interleaving, recall that the scoring algorithm always creates a preference for one of the rankers when there is one click: this click will be on a result from Team A or Team B. This scoring scheme leads to unnecessary variance when the rankings A and B are very similar. We therefore also define a slightly modified scoring proposed by Radlinski and Craswell [2010], which we call "deduped" scoring. If A and B share identical top k results, a click on any of these results is ignored and not counted towards h_a and h_b in Equations (5) and (6). In other words, suppose that both input rankings place documents d_1 through d_k in the top k positions. Any clicks on these results would previously create a preference based solely on the random team preference bit, and cannot include any information. Thus, for each logged query $(q, C_a, C_b) \in Q$, each click is assigned to C'_a or C'_b only if it is not in the shared top k results. Not counting the clicks in the top k will not bias the outcome of Team-Draft interleaving, but will decrease the variance of the experiment.

Deduped Binary. This is exactly the binary scoring rule defined above except using the deduped entries, (q, C'_a, C'_b).

Deduped Click. This is exactly the click scoring rule defined above except using the deduped entries, (q, C'_a, C'_b).

Deduped Normalized Click. This is exactly the normalized click scoring rule defined above except using the deduped entries, (q, C'_a, C'_b), and normalizing also using the duplicate clicks C'_t:

$$\Delta'_{norm}(Q) = \sum_{(q,C_a,C_b)\in Q} \frac{|C'_a| - |C'_b|}{|C'_a \cup C'_b \cup C'_t|}. \tag{11}$$

Time-based Query Aggregation. We also evaluate multi-query aggregation strategies. Intuitively, two queries issued in quick succession (e.g., with less than 1 minute delay time) may belong to the same information seeking session. Thus, it may be beneficial to compute the above measures over aggregated sessions as opposed to individual queries. For our experiments, we defined sessions using a timeout threshold (i.e., two consecutive queries with dwell time less than the timeout threshold are considered to be from the same session).

9.2. Evaluation Methodology

We evaluate over a set of scored queries $S(Q) = \{s_1, \ldots, s_n\}$, where s_i corresponds to the score or credit assignment of the i-th query.[11] It is easy to see that the sample mean $\mu_{S(Q)} = \frac{1}{n}\sum s_i$ directly corresponds the scoring rules described above.

Assuming that $\mu_{S(Q)}$ is normally distributed (which is approximately true for large n), then we can use the *z-score* to characterize the confidence of each method [Mood et al. 1974]. We define the z-score of $S \equiv S(Q)$ to be

$$z_S = \frac{\mu_S}{\sigma_S}\sqrt{n}, \tag{12}$$

where n is the number of queries used, μ_S is the sample mean, and σ_S is the standard deviation. Note that n can vary depending on how queries are aggregated into sessions (i.e., using larger timeouts will result in larger sessions, and thus smaller n).

[11]For example, if using Equation (9), then s_i corresponds to the signed difference in clicks for the i-th query.

Table XI.
Showing z-score scaled to binary z-score for different scoring strategies on ArXiv.org (top) and Yahoo (bottom) for Balanced interleaving.

	$\mathcal{E}_A \succ \mathcal{B}_A$	$\mathcal{B}_A \succ \mathcal{A}_A$	$\mathcal{E}_A \succ \mathcal{A}_A$	$\mathcal{E}_A \succ \mathcal{D}_A$	$\mathcal{D}_A \succ \mathcal{C}_A$	$\mathcal{E}_A \succ \mathcal{C}_A$	Median
Binary	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Click	1.14	0.99	1.25	1.11	1.00	1.04	1.08
Norm. Click	1.06	0.89	1.18	1.23	1.14	0.83	1.10
Binary-Direct	0.64	0.87	1.10	0.95	1.19	1.09	1.02
Click-Direct	0.41	0.85	0.69	1.01	1.40	1.14	0.93
Norm. Click-Direct	0.83	0.86	1.18	1.20	1.23	0.96	1.07

	$\mathcal{B}_Y \succ \mathcal{A}_Y$	$\mathcal{C}_Y \succ \mathcal{A}_Y$	$\mathcal{D}_Y \succ \mathcal{A}_Y$	$\mathcal{C}_Y \succ \mathcal{B}_Y$	$\mathcal{D}_Y \succ \mathcal{B}_Y$	$\mathcal{D}_Y \succ \mathcal{C}_Y$	Median
Binary	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Click	0.89	1.00	0.98	1.01	1.00	1.15	1.00
Norm. Click	1.14	1.05	1.05	0.96	0.99	0.97	1.02
Binary-Direct	0.76	1.03	0.99	1.10	1.06	0.85	1.01
Click-Direct	0.38	0.98	0.95	1.18	1.09	1.13	1.035
Norm. Click-Direct	0.95	1.05	1.06	1.05	1.05	0.98	1.05

Since μ_S is normally distributed by assumption, the z-score of μ_S corresponds exactly to its deviation from zero in a normal distribution with zero mean and unit variance. This offers an intuitive way to interpret the z-score. For example, a z-score of $z_S = 2$ corresponds to a confidence of approximately 95%.

9.3. Results

We evaluated using both the Balanced Interleaving Datasets (ArXiv.org and Yahoo!) and the Team-Draft Interleaving datasets (ArXiv.org and Bing). For each interleaving experiment, we compared the ratio of z-scores of the various click aggregation strategies to the binary click strategy. This allows us to measure the relative differences in sensitivity for each interleaving experiment. For example, a z-score ratio of 1.05 indicates that the proposed click aggregation strategy is 5% more confident than the binary strategy. Since confidence increases at a rate of $\sqrt{n}$, this means the binary strategy requires $1.05^2 \approx 1.10$ times more data to achieve the same confidence level.

Table XI shows the results for Balanced Interleaving on ArXiv.org and Yahoo!. The last column of each table summarizes each row by its median. We observe that none of the first three methods dominate the others, although the Normalized Click method seems to be best overall. On the Yahoo! experiments, we find that Click-Direct versions are typically slightly better than their counterparts.

Table XII shows the results for Team-Draft Interleaving on ArXiv.org and Bing. While we again observe that none of the first three methods dominate the others, the deduped methods are typically much more sensitive. This effect is more pronounced on the Bing interleaving experiments, which is likely due to (1) the higher percentage of clicks at the top rank positions for Web search as compared to scholarly paper search, and (2) the ranking functions used on the ArXiv.org experiments often produce results that differ at all positions, while the various Bing ranking functions often return the same top results for most queries. For the Bing experiments, note that the dramatic difference between the non-deduped and deduped metrics on $\mathcal{F}_B \succ \mathcal{E}_B$ is due to the experiment affecting an especially small fraction of queries. Deduping removes clicks from queries where two identical rankings were interleaved, after which the signal from interleaving is much stronger on the remaining queries.

Table XIII shows the results the Normalized Click-Direct for Balanced Interleaving on Yahoo! when aggregating multiple query sessions into a single session. This is done using a timeout threshold: all queries issued within the timeout threshold of the

Table XII.
Showing z-score scaled to binary z-score for different scoring strategies on ArXiv.org (top) and Bing (bottom) for Team-Draft interleaving.

	$\mathcal{E}_A \succ \mathcal{B}_A$	$\mathcal{B}_A \succ \mathcal{A}_A$	$\mathcal{E}_A \succ \mathcal{A}_A$	$\mathcal{E}_A \succ \mathcal{D}_A$	$\mathcal{D}_A \succ \mathcal{C}_A$	$\mathcal{E}_A \succ \mathcal{C}_A$	Median
Binary	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Click	1.25	0.78	0.84	0.71	0.36	0.91	0.81
Norm. Click	0.98	0.90	1.04	1.23	0.75	0.95	0.97
Deduped Binary	1.24	1.04	0.97	1.40	1.42	1.18	1.21
Deduped Click	1.38	0.81	0.83	1.00	0.58	1.02	0.91
Norm. Deduped Click	1.34	0.99	1.03	1.44	1.22	1.19	1.21

	$\mathcal{B}_B \succ \mathcal{A}_B$	$\mathcal{C}_B \succ \mathcal{B}_B$	$\mathcal{C}_B \succ \mathcal{A}_B$	$\mathcal{D}_B \succ \mathcal{C}_B$	$\mathcal{F}_B \succ \mathcal{E}_B$	Median
Binary	1.00	1.00	1.00	1.00	1.00	1.00
Click	1.04	1.05	1.07	1.05	1.00	1.05
Norm. Click	1.00	0.97	0.98	0.97	1.01	0.98
Deduped Binary	1.35	1.60	1.55	1.52	2.86	1.55
Deduped Click	1.18	1.55	1.54	1.33	3.12	1.54
Norm. Deduped Click	1.48	1.58	1.57	1.50	2.87	1.57

Table XIII.
Same as bottom of Table XI, with Normalized Click-Direct, but where the counting and normalization are done at the session level. Different rows correspond to different timeout for the definition of a session.

Timeout (s)	$\mathcal{B}_Y \succ \mathcal{A}_Y$	$\mathcal{C}_Y \succ \mathcal{A}_Y$	$\mathcal{D}_Y \succ \mathcal{A}_Y$	$\mathcal{C}_Y \succ \mathcal{B}_Y$	$\mathcal{D}_Y \succ \mathcal{B}_Y$	$\mathcal{D}_Y \succ \mathcal{C}_Y$	Median
0	0.95	1.05	1.06	1.05	1.05	0.98	1.05
10	0.94	1.05	1.06	1.05	1.06	0.99	1.05
20	0.95	1.06	1.08	1.06	1.06	1.02	1.06
50	1.02	1.08	1.08	1.05	1.05	1.06	1.055
100	1.10	1.09	1.09	1.00	1.02	1.03	1.06
200	1.17	1.09	1.10	0.96	0.99	1.00	1.045
500	1.32	1.07	1.06	0.88	0.90	0.81	0.98

previous query are considered to be in the same session. Using timeouts of 20 and 50 seconds both consistently, but only slightly, improve the z-score.

9.4. Summary

Click, Normalized Click, Binary-Direct, Click-Direct and Normalized Click-Direct scoring have a small effect on the outcome of interleaving evaluations compared to Binary scoring. However, Normalized Click-Direct tends to improve the sensitivity of Balanced Interleaving. Deduped Click scoring has a large effect on the sensitivity of Team-Draft Interleaving, improving sensitivity by over 50% on the Bing dataset. We also observe small benefits from appropriately grouping queries into sessions.

10. HOW CAN ONE LEARN A MORE SENSITIVE CLICK SCORING STRATEGY?

One shortcoming of the scoring approaches considered in Section 9 is that they give equal weight to all clicks on each query, which is likely to be suboptimal in practice. For example, observing a user returning to the search results page immediately after clicking on a result a is probably an indication that the landing page of a was not relevant. Discounting such clicks should lead to a noise reduction in the interleaving signal. We now examine whether one can learn a more refined click scoring function that distinguishes between different types of clicks.

In this section, we evaluate the effectiveness of inverting the z-test first proposed in Yue et al. [2010]. In Section 10.4 we reanalyze learning results from Yue et al. [2010] for Team-Draft Interleaving on an extended ArXiv.org dataset, and in Section 10.5 we present new learning results for Balanced Interleaving on a previously unanalyzed extended Yahoo! dataset.

10.1. Problem Formulation

Following Yue et al. [2010], we will use a linear model $score(q, c) = w^\top \varphi(q, c)$ to score clicks, where w is a vector of parameters to be learned and $\varphi(q, c)$ returns a feature vector describing each click c in the context of the entire query session q. Focusing on the normalized click scoring rule, we can now rewrite $\Delta(Q)$ as

$$\Delta_w(Q) = \sum_{(q,C_a,C_b)\in Q} w^\top \Phi(q, C_a, C_b), \tag{13}$$

where

$$\Phi(q, C_a, C_b) = \frac{1}{|C_a \cup C_b|}\left(\sum_{c\in C_a} \varphi(q, c) - \sum_{c\in C_b} \varphi(q, c)\right). \tag{14}$$

Feature vectors $\varphi(q, c)$ contain features that describe the click in relation to position in the interleaved ranking, order, and presentation.

The idea behind learning is to find a scoring function that results in the most sensitive hypothesis test. To illustrate this goal, consider the following hypothetical scenario where the scoring function $score(q, c) = w^\top \varphi(q, c)$ differentiates the last click of a query session from other clicks within the same session. The corresponding feature vector $\varphi(q, c)$ would then have two binary features

$$\varphi(q, c) = \begin{pmatrix} 1, \text{ if c is last click; 0 otherwise} \\ 1, \text{ if c is not last click; 0 otherwise} \end{pmatrix}. \tag{15}$$

Assume for simplicity that every query session has 3 clicks, with "not last clicks" being completely random while "last clicks" favor the better retrieval function with 60% probability. Using the weight vector $w^\top = (1, 1)$ (i.e., the conventional scoring function), one will eventually identify the better retrieval function (typically after ≈280 queries using a t-test with $p = 0.05$). However, the optimal weight vector $w^\top = (1, 0)$ will identify the better retrieval function much faster (typically after ≈150 queries), since it eliminates noise from the non-informative clicks.

The learning problem can be thought of as an "inverse" hypothesis test: given data for pairs (h, h') of retrieval functions where we know $h \succ h'$, find the weights w that maximizes the power of the test statistic on new pairs. More concretely, we assume that we are given a set of ranking function pairings $\{(h_1, h'_1), \ldots, (h_k, h'_k)\}$ for which we know without loss of generality that h_i is better than h'_i, that is, $h_i \succ h'_i$. This preference may be known by construction (e.g., h'_i is a degraded version of h_i), by running interleaving until the conventional test statistic that scores each click uniformly becomes significant, or through some expensive annotation process (e.g., user interviews, manual assessments).

For each pair (h_i, h'_i), we assume that n_i queries have been interleaved (ignoring queries with no clicks). For each query q_j, the clicks C^j_a and C^j_b for each ranking are recorded in a triple (q_j, C^j_a, C^j_b). All triples are combined into one training sample $Q = ((q_1, C^1_a, C^1_b), \ldots, (q_n, C^n_a, C^n_b))$.[12] After training, the learned w and the resulting scoring function Δ_w will be applied to new pairs of retrieval functions (h_{test}, h'_{test}) of yet unknown relative retrieval quality.

[12]We are essentially treating all interleaving pairs as a single combined example. A more principled approach may be to explicitly treat each interleaving pair as a separate example.

10.2. Inverse z-Test

The z-test is the significance test that directly relates to the z-score (12) described in Section 9.2 (the relationship will be made clear in the following). For any dataset, the z-test assumes that the sample mean (e.g., the average normalized difference of clicks) is normally distributed, which is approximately satisfied for large sample sizes. As discussed in Section 9.2, the z-score corresponds exactly to the standard deviation of a normal distribution with zero mean and unit variance. The z-test is then the interpretation of the z-score into a p-value. For example, a z-test on a sample S with z-score $z_S = 2$ will yield a p-value of approximately 0.05 (i.e., 95% confidence).

We can write the z-score (12) of a set of logged queries Q as

$$\frac{\mu_w(Q)}{\sigma_w(Q)}\sqrt{n}, \tag{16}$$

where

$$\mu_w(Q) = \frac{1}{n}\sum_j w^\top \Phi\big(q_j, C_a^j, C_b^j\big),$$

and

$$\begin{aligned}\sigma_w(Q) &= \sqrt{\frac{1}{n}\sum_j \big(\mu_w(Q) - w^\top \Phi\big(q_j, C_a^j, C_b^j\big)\big)^2} \\ &= \sqrt{\frac{1}{n}\sum_j \big[w^\top \Phi\big(q_j, C_a^j, C_b^j\big)\big]^2 - \left[\frac{1}{n}\sum_j w^\top \Phi\big(q, j, C_a^j, C_b^j\big)\right]^2}.\end{aligned}$$

The *inverse z-test* is then the learning method that optimizes the statistical power of a z-test, which amounts to finding the w that optimizes the z-score (16) on the training set. This corresponds to solving the following optimization problem.

Optimization Problem 1 (Inverse z-Test).

$$\begin{aligned} w^* &= \operatorname*{argmax}_w \frac{\frac{1}{n}\sum_j w^\top \Phi\big(q_j, C_a^j, C_b^j\big)}{\sqrt{\frac{1}{n}\sum_j \big[w^\top \Phi\big(q_j, C_a^j, C_b^j\big)\big]^2 - \big[\frac{1}{n}\sum_j w^\top \Phi\big(q_j, C_a^j, C_b^j\big)\big]^2}}\sqrt{n} \\ &= \operatorname*{argmin}_w \frac{\frac{1}{n}\sum_j \big[w^\top \Phi\big(q_j, C_a^j, C_b^j\big)\big]^2 - \big[\frac{1}{n}\sum_j w^\top \Phi\big(q_j, C_a^j, C_b^j\big)\big]^2}{\frac{1}{n}\big[\sum_j w^\top \Phi\big(q_j, C_a^j, C_b^j\big)\big]^2} \\ &= \operatorname*{argmin}_w \frac{\sum_j \big[w^\top \Phi\big(q_j, C_a^j, C_b^j\big)\big]^2}{\big[\sum_j w^\top \Phi\big(q_j, C_a^j, C_b^j\big)\big]^2} \end{aligned} \tag{17}$$

While (17) has two symmetric solutions, we are interested only in the one where $\sum_j w^{*\top}\Phi(q_j, C_a^j, C_b^j) > 0$. Using the abbreviated notation $\Psi_j = \Phi(q_j, C_a^j, C_b^j)$, this optimization problem can be rewritten as

$$w^* = \operatorname*{argmin}_w \frac{w^\top\big[\sum_j \Psi_j \Psi_j^\top\big]w}{\big(w^\top \sum_j \Psi_j\big)^2} = \operatorname*{argmax}_w \frac{\big(w^\top \sum_j \Psi_j\big)^2}{w^\top\big[\sum_j \Psi_j \Psi_j^\top\big]w}. \tag{18}$$

For any w solving (18), cw with $c > 0$ is also a solution. We can thus rewrite (18) as

$$w^* = \operatorname*{argmax}_w \left[w^\top \sum_j \Psi_j \right] \text{ s.t. } w^\top \left[\sum_j \Psi_j \Psi_j^\top \right] w = 1. \tag{19}$$

Using the Lagrangian

$$L(w, \alpha) = w^\top \sum_j \Psi_j - \alpha \left(w^\top \left[\sum_j \Psi_j \Psi_j^\top \right] w - 1 \right), \tag{20}$$

and solving for zero derivative with respect to w and α, one arrives at a closed form solution. Denoting $\Psi = \sum_j \Psi_j$ and $\Sigma = \sum_j \Psi_j \Psi_j^\top$ the solution can be written as

$$w^* = \frac{\Sigma^{-1}\Psi}{\sqrt{\Psi^\top \Sigma^{-1} \Psi}}.$$

While not used in our experiments, a regularized version of the covariance matrix Σ_{reg} can be used to prevent overfitting. One straightforward approach is to add a ridge $\Sigma_{reg} = \Sigma + \gamma I$, where I is the identity matrix[13] and γ is the regularization parameter.

10.3. Evaluation Methodology

One difficulty in evaluating across different interleaving pairs is the fact that different buckets contain different amounts of data. For instance, a pair of retrieval functions that are inherently difficult to distinguish might have logged more data than an easier to distinguish pair, so the z-score measure from (12) may not accurately reflect the per-query gains in performance. For any dataset of scored queries $S = \{s_1, \ldots, s_n\}$, we will evaluate here using the *query-normalized z-score*,

$$\bar{z}_S = \frac{\mu_S}{\sigma_S}, \tag{21}$$

for μ_S and σ_S as defined in Section 9.2. Intuitively, the query-normalized z-score measures the expected per-query contribution to the confidence of the aggregate statistic μ_S. The z-score and query-normalized z-score are equivalent when comparing on a single interleaving experiment, since the number of queries n is constant.

10.4. Experiments on ArXiv.org (Team-Draft Interleaving)

In this section, we re-analyze a subset of the learning results from [Yue et al. 2010] for Team-Draft Interleaving. We trained a model using the inverse z-test on the six Team-Draft Interleaving pairs from ArXiv.org (see Section 4.2). Afterwards, the learned model was used to score queries on an extended ArXiv.org dataset described below.

The extended dataset from Yue et al. [2010] was generated by interleaving pairs of retrieval functions without necessarily having knowledge of which retrieval function is superior within each pair. For example, one retrieval function that we considered modifies the original retrieval function by giving additional weight to query/title similarity. It is a priori unclear whether this would result in improved retrieval quality. We report the effectiveness of the inverse z-test on twelve interleaving experiments which interleave all pairs of six distinct retrieval functions. Each interleaving experiment was performed observing between 400 and 650 queries.

[13]One can alternatively use a partial identity matrix (with zeros in some of the diagonal entries). Then the learning objective will only regularize the weights of features corresponding to nonzero diagonal entries.

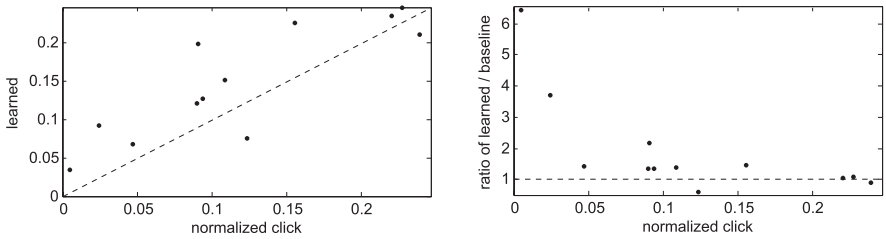

Fig. 8. Query-normalized z-score values of learned model versus baseline on extended ArXiv.org dataset.

The features used describe a diverse set of properties related to clicking behavior, including the rank and order of clicks, and whether search result clicks led to a PDF download on ArXiv.org.[14] A total of 14 features were used. See Section 5.2 in Yue et al. [2010] for a more detailed description.

The results are shown in Figure 8. The left graph plots the query-normalized z-scores of the learned model versus the baseline (which scores all clicks equally) using the Normalized Click scoring rule. We see that the learned model has improved the expected per-query confidence on 10 of the 12 interleaving pairs. We also note that the direction of the tests all agree (both the learned model and the baseline agree on which retrieval function is preferred).

The right graph in Figure 8 plots the ratio of z-scores (i.e. the relative z-scores) versus the query-normalized z-score of the baseline. This captures the relative gain in sensitivity with respect to the inherent difficulty of the interleaving pair. The relative gains are more substantial for interleaving pairs *comparing more similar ranking functions*.[15] The median relative z-score value is 1.37.[16]

10.5. Experiments on Yahoo! (Balanced Interleaving)

We also evaluated the inverse z-test for Balanced Interleaving using a new, previously unanalyzed, extended Yahoo! dataset. This extended dataset contains 71 interleaving experiments and is spread across 16 markets (e.g., Brazil, U.S., India), with 4 to 6 interleaving experiments conducted in each market. The six interleaving experiments described in Section 4.4 represent the U.S. market in this dataset. The number of query sessions logged varied between 139 thousand and 30 million, with a mean of approximately 1.5 million and median of around 930 thousand.

Similar to Section 10.4, we used features that describe a diverse set of properties related to clicking behavior. Given the size of the Yahoo! dataset, we used a more fine-grained feature space with 51 features in total, which are summarized in the following.

(A) Functions of the click rank, such as $\log(rank)$ and $\sqrt{rank}$. These features are then normalized either by the number of clicks or total score of that feature for the query. (9 features)
(B) Indicator features of the click rank normalized by the # of clicks. We also use versions that are only active for single-click queries and multi-click queries. (20 features)
(C) Indicator features of whether the click rank was below the top 4, with the same normalization and variations as (B) above. (8 features)

[14]All search results correspond to research papers that are available for download.
[15]This is unsurprising since relative gains tend to be larger when the denominator values are smaller.
[16]This means the baseline typically requires $1.37^2 \approx 1.88$ times more data to achieve the same confidence.

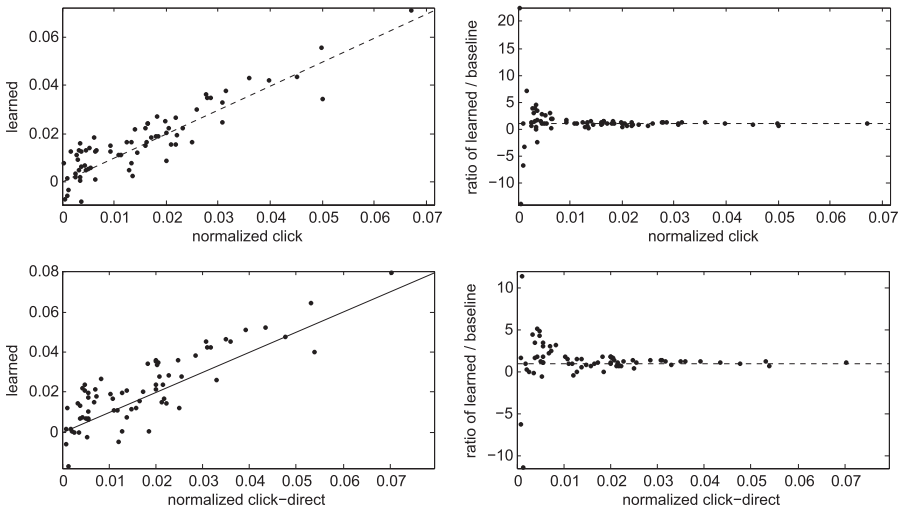

Fig. 9. Query-normalized z-score values of learned model versus baseline on extended Yahoo! dataset.

(D) Indicator features of whether the click was the first, last or a regression click, with the same normalization as (B) above. We also use a version that is active only for multi-click queries where the first click has $rank > 1$. (6 features)
(E) Indicator features of whether the dwell time of the click was at least 30, 60, 90 or 120 seconds. We use the same normalization as (B) above, and the same variations as in (D) above. (8 features)

We performed leave-one-market-out testing, where we trained our model on 15 markets and compared with the baseline on interleaving experiments from the remaining market. This is a realistic scenario for services such as commercial search engines.

The results are shown in Figure 9. The left graphs plot the query-normalized z-scores of the learned model versus the baseline (which scores all clicks equally) using the Normalized Click and Normalized Click-Direct scoring rules. In both cases, we see that the learned model has improved the expected per-query confidence on most of the interleaving experiments (48/71 for Normalized Click and 47/71 for Normalized Click-Direct). When comparing the two learned models, we find the learned Normalized Click-Direct model to have superior z-score in 50/71 of the interleaving pairs.

The right graphs in Figure 9 plot the ratio of z-scores versus the query-normalized z-score of the baseline. This captures the relative gain in performance with respect to the inherent difficulty of the interleaving pair. The median relative z-score values for Normalized Click and Normalized Click-Direct are 1.09 and 1.25, respectively.[17] For a few of the difficult interleaving pairs, the learned model interprets user preferences to be opposite of the baseline (i.e., the z-scores are negative; this happens in 4/71 cases for Normalized Click and 7/71 cases for Normalized Click-Direct).

10.6. Summary

We find that the inverse z-test consistently improves the confidence of the resulting test statistic for both the extended ArXiv.org and Yahoo! datasets, with 37% and 25% median gains in relative confidence, respectively. For the Team-Draft Interleaving experiments

[17]This means the baseline typically require $1.09^2 \approx 1.19$ and $1.25^2 \approx 1.56$ times more data to achieve the same confidence, respectively.

on ArXiv.org, the learned scoring function is always in agreement with the baseline. For the Balanced Interleaving experiments on Yahoo!, the learned scoring function disagrees with the baseline for some interleaving pairs. Note that these disagreeing interleaving pairs are always those where using the baseline scoring function shows only a slight preference for one retrieval function in the interleaving pair.

11. LIMITATIONS, DISCUSSION, AND FUTURE WORK

Our empirical results suggest that Balanced and Team-Draft Interleaving are attractive online evaluation methods due to their reliability, efficiency, and widespread applicability. However, as already indicated throughout the paper, one should keep in mind the inherent limitations of the experiments and of the methods themselves. We now discuss these limitations in detail and suggest interesting directions for future work.

As a general comment, we note that our study is a field study, and not a controlled lab study with qualitative feedback. This inherently limits the types of evaluations we can conduct. Furthermore, our experiments were conducted using small to moderate numbers of retrieval functions, and only within a limited number of search domains. As such, many properties of user behavior (many of which to be discussed in the following) could not be reliably measured to high precision or generality in our study.

The rest of this section is organized as follows. In Sections 11.1 and 11.2, we provide detailed discussions of two key limitations that arise immediately from our study on applying interleaving methods to web and scholarly paper search. In Section 11.3, we discuss other interesting future directions that are motivated by considering interleaving methods in broader contexts.

11.1. Click versus Relevance

Interleaving evaluation relies on clicks as a signal of relevance. But this assumption is not always true, at least not for conventional ways of defining relevance in Information Retrieval [Cleverdon et al. 1966]. In this section, we review some findings of an earlier analysis on the discrepancies between relevance inferred from clicks and relevance as assessed by human experts [Chapelle and Zhang 2009] as well as providing further observations. As we will see in the following, it is important to keep in mind that the expert judgments should not necessarily be considered as ground truth for relevance.

A first cause of inconsistency is that clicks mostly measure the user's expectation of relevance, whereas editors judge the relevance of the destination page. The user's expectation is based on the summary shown, which usually consists of a title, short text description, and destination web page address, along with visual cues such as bolding of query words. Click-relevance disagreement can be divided into at least two subcategories: cases where the relevance of the search result snippets is very different from that of the landing page; and cases where users click based on the trustworthiness of the page rather than the relevance of the page. The first sub-category is most often related to the presentation of the title and summary of the url (cf. Yue et al. [2010]). For instance, the summary might look attractive, with the result page in fact being nonrelevant. The opposite effect is also possible, where the user may not click on a relevant result if the user finds the information he was looking for directly in the snippet. An example query for the second subcategory is "travel insurance". While many small insurance companies focus on selling travel insurance (more relevance in terms of relevance judgment), users often click on sites of well-known insurance companies for whom travel insurance is only a small fraction of their business.

This leads into a second potential cause of inconsistency, which arises due to the difficulty in defining *relevance*. For instance, consider the query "adobe." The company home page `www.adobe.com` is usually considered the most relevant. However, few users

who issue this query have a goal of just reaching the company home page. Most Web search users click on the link to download Adobe's most popular software, Acrobat Reader. Another example query is "bank of america." Most users prefer to click on the online banking page `www.bankofamerica.com/onlinebanking`, while editors tend to consider the company home page `www.bankofamerica.com` more relevant for this query. Further cases illustrating the difficulty in defining relevance include: Acronyms such as *SIGIR* and ambiguous queries such as *jaguar* (e.g., see Clarke et al. [2008]) where users have a single intent in mind while judges must estimate the relative weight to give different possible meanings; queries where the relevance of results changes over time, such as *Iraq war* [Kulkarni et al. 2011], often involve trading off depth of information versus recency in addition to the challenge due to ambiguity in this particular example; cases where the amount of information must be traded off against the authoritativeness of results (e.g., a seemingly authoritative Wikipedia page versus a nonauthoritative yet more detailed personal blog); queries where the background knowledge of the user is relevant (e.g., *information retrieval*); settings where the search results are personalized for individual users or groups of users; potentially misspelled queries that may be meaningful (e.g., *aim* is often mistyped *ai* [Radlinski et al. 2010b]).

Finally, standard summary metrics such as NDCG and MAP do not discount the utility of documents with redundant information, whereas rankings that include a diversity of views on a topic may be preferred by real users [Radlinski et al. 2009].

Given that editorial judgments and implicit relevance estimates from clicks do not capture the same notion of relevance and can be seen as complementary, an even more robust method of evaluation could be to combine both types of information into a single metric. The best way to combine them is a question for further research.

11.2. Biases in Interleaving

As discussed in Section 3, one can adversarially construct cases where both Balanced Interleaving and Team-Draft Interleaving produce biased preferences even for perfectly rational and informative clicks. The problem lies in their definition of "fairness," and how it relates to multiple user intents and similarities between rankings. Given further assumptions about these relationships (e.g. only non-ambiguous queries), it may be possible to make theoretical statements about the accuracy of the interleaving methods studied in this article. Whether a different interleaving method exists that is provably unbiased without any assumptions is an open question. An interesting connection can be drawn with social choice theory due to the resemblance between interleaving and voting. Empirically, however, we find that both Balanced Interleaving and Team-Draft Interleaving are highly accurate in determining the preference even between retrieval functions of small quality difference.

One additional assumption of both interleaving methods is that user utility decomposes into clicks on individual documents in a ranking. Due to this focus on individual documents, it is unclear how to evaluate more global qualities such as the diversity of a set of documents. Similarly, it is unclear how to handle results not presented as a linear ranking. An example is domain collapsing, where indented results are inserted into a Web search ranking.

Finally, interleaving needs an observable action in order to draw any inference. As such, current interleaving methods ignore the utility of results that yield no clicks (e.g., returning the current time when searching for "current time"). However, with high-resolution video cameras becoming popular, interleaving techniques may be able to use eye-tracking as an additional source of feedback beyond clicks [Salojärvi et al. 2005; Moe et al. 2007; Buscher et al. 2008].

11.3. Other Limitations and Future Research

Interleaving itself is an intervention that changes the search experience. Although interleaving is minimally disruptive to the presentation format, the quality of the search results will vary depending on the ranking functions used. It remains to characterize or quantify the utility of the rankings produced by interleaving (as opposed to the input rankings that are compared using interleaving). This is further complicated when comparing ranking functions that optimize for global objectives such as diversity. For example, two ranking functions that generate well diversified rankings might combine to yield poorly diversified interleaved rankings.

While this article has focused on comparing small sets of retrieval functions, in practice one often needs to find the best retrieval function among a large set of n retrieval functions. Assuming stochastic preferences are transitive, then there exist algorithms for identifying the best ranking whose complexity scales as $O(n)$ [Feige et al. 1994]. Furthermore, one might also wish to account for the utility lost to the user (i.e. running an interleaving experiment instead of using the best retrieval function that is known only in hindsight). This issue of exploration versus exploitation is captured within a formal model named the Dueling Bandits Problem [Yue and Joachims 2009; Yue et al. 2009; Yue and Joachims 2011], which in this context effectively assumes that (A) the quality of any interleaved ranking is bounded between the utilities of the original two rankings, and (B) user preferences obey strong stochastic transitivity (see Equation (7)). Under these assumptions it can be shown that the expected regret scales as $O(n)$, which is information-theoretically optimal [Yue et al. 2009]. However, while the assumption of stochastic transitivity did indeed hold in all our experiments (see Section 5.3), due to the relatively small number of retrieval functions tested, it is difficult to determine if these properties apply generally for the interleaving mechanism.

The inverse z-Test method presented in Section 10 learns a click scoring function w that maximizes z-score, or confidence, of the resulting test statistic Δ_w. Another interesting direction for future work is learning a (click) scoring function to optimize a wider range of criteria that might better reflect practical concerns, such as minimizing the p-value given a fixed budget of queries. For example, given a budget of ten thousand queries, increasing the z-score of a very confident interleaving pair by 10% might prove less beneficial than increasing the z-score of a less confident interleaving pair by 10%, despite the absolute gains for the first interleaving pair being greater. In such a setting, the inverse z-Test would favor learning a scoring function that maximizes the confidence of the first interleaving pair, which can be suboptimal in practice.

12. CONCLUSIONS

In this article, we explored two interleaving methods for unobtrusively eliciting preference feedback from observable user behavior. Using data from large-scale field studies on three different search engines, including scholarly search and commercial web search, we provide the first comprehensive evaluation of interleaving compared to other implicit feedback methods and conventional evaluation based on manual relevance judgments. Our conclusions can be summarized as follows:

Does interleaving agree with expert assessments? Yes, preferences from interleaving generally agreed with evaluation methods based on relevance judgments, and with relevance differences known by construction.

Do absolute metrics and interleaving agree? Many absolute metrics derived from implicit feedback did not agree with interleaving and with manual relevance judgments. Clicks@1 is the absolute metric that showed the highest agreement.

How much click data is needed to obtain a statistically reliable preference? Interleaving requires approximately 1–2 orders of magnitude less data than absolute metrics,

with data requirements on the order of thousands to tens of thousands of queries for detecting even small differences in retrieval quality.

What is the value of a click relative to a judged query? We found that clicks from ten interleaved queries provide roughly the same statistical evaluation power as one manually judged query. This makes interleaving substantially more economical and timely than manual judgments.

How sensitive is interleaving to different click aggregation schemes? Varying the click aggregation strategy generally yields small differences, with the deduped strategies in Team-Draft Interleaving yielding the largest improvement. Aggregating queries into longer sessions can also slightly improve the quality of the resulting test statistic.

How can one learn a more sensitive click scoring strategy? By applying the inverse z-Test using a linear combination of click features, one can learn a more sensitive click scoring strategy that appropriately weights the importance of different types of clicks.

Overall, our evidence suggests interleaving as an attractive methodology to complement or replace existing approaches based on manual judgments or absolute metrics, especially in search applications where manual judgments are not economically feasible. However, as discussed above in Section 11, further research is needed to fully understand the range of search domains interleaving can effectively be applied to, as well as its strengths and weaknesses across all domains.

ACKNOWLEDGMENTS

We thank Nick Craswell, Yue Gao, Madhu Kurup and Ya Zhang for their collaboration in the research that led to this article. We also thank Paul Ginsparg and Simeon Warner for interesting discussions and for enabling the experiments on ArXiv.org. Olivier Chapelle would like to thank all the people in Yahoo! Search who made the the Yahoo! experiments possible by designing and implementing the interleaving module, in particular Olivier Galizzi, Laurent Goujon and Ya Zhang. Filip Radlinski would like to thank the developers of the Bing search engine, and in particular Rishi Agarwal, Nikunj Bhagat, Eric Hecht, Manish Malik and Sambavi Muthukrishnan for making the research using Bing possible.

REFERENCES

Agichtein, E., Brill, E., Dumais, S., and Ragno, R. 2006. Learning user interaction models for prediction web search results preferences. In *Proceedings of the ACM Conference on Research and Development in Information Retrieval (SIGIR)*. 3–10.

Agrawal, R., Halverson, A., Kenthapadi, K., Mishra, N., and Tsaparas, P. 2009. Generating labels from clicks. In *Proceedings of the 2nd ACM International Conference on Web Search and Data Mining*. 172–181.

Ali, K. and Chang, C. 2006. On the relationship between click-rate and relevance for search engines. In *Proceedings of the Conference on Data-Mining and Information Engineering*. 213–222.

Allan, J., Aslam, J. A., Carterette, B., Pavlu, V., and Kanoulas, E. 2008. Million query track 2008 overview. In *Proceedings of TREC*.

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

Becker, H., Meek, C., and Chickering, D. M. 2007. Modeling contextual factors of click rates. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*. AAAI Press, 1310–1315.

Boyan, J., Freitag, D., and Joachims, T. 1996. A machine learning architecture for optimizing web search engines. In *Proceedings of the AAAI Workshop on Internet Based Information Systems*. 1–8.

Buckley, C. and Voorhees, E. M. 2004. Retrieval evaluation with incomplete information. In *Proceedings of the ACM Conference on Research and Development in Information Retrieval (SIGIR)*. 25–32.

Buscher, G., Dengel, A., and van Elst, L. 2008. Eye movements as implicit relevance feedback. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, M. Czerwinski, A. M. Lund, and D. S. Tan, Eds. ACM, 2991–2996.

Carterette, B., Allan, J., and Sitaraman, R. 2006. Minimal test collections for retrieval evaluation. In *Proceedings of the ACM Conference on Research and Development in Information Retrieval (SIGIR)*. 268–275.

CARTERETTE, B., BENNETT, P. N., CHICKERING, D. M., AND DUMAIS, S. T. 2008. Here or there: Preference judgements for relevance. In *Proceedings of the European Conference on Information Retrieval (ECIR)*. 16–27.

CARTERETTE, B. AND JONES, R. 2007. 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)*. 217–224.

CHAPELLE, O., METLZER, D., ZHANG, Y., AND GRINSPAN, P. 2009. Expected reciprocal rank for graded relevance. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management (CIKM'09)*. ACM, New York, NY, 621–630.

CHAPELLE, O. AND ZHANG, Y. 2009. A dynamic bayesian network click model for web search ranking. In *Proceedings of the 18th International Conference on World Wide Web*. ACM, New York, NY, 1–10.

CLARKE, C. L. A., KOLLA, M., CORMACK, G. V., VECHTOMOVA, O., ASHKAN, A., BÜTTCHER, S., AND MACKINNON, I. 2008. Novelty and diversity in information retrieval evaluation. In *Proceedings of the ACM Conference on Research and Development in Information Retrieval (SIGIR)*. 659–666.

CLAYPOOL, M., LE, P., WASEDA, M., AND BROWN, D. 2001. Implicit interest indicators. In *Proceedings of the International Conference on Intelligent User Interfaces (IUI)*. 33–40.

CLEVERDON, C. W., MILLS, J., AND KEEN, E. M. 1966. *Factors Determining the Performance of Indexing Systems*. Cranfield: College of Aeronautics.

CRASWELL, N., ZOETER, O., TAYLOR, M. J., AND RAMSEY, B. 2008. An experimental comparison of click position-bias models. In *Proceedings of the ACM International Conference on Web Search and Data Mining (WSDM)*. M. Najork, A. Z. Broder, and S. Chakrabarti, Eds., ACM, 87–94.

DUPRET, G., MURDOCK, V., AND PIWOWARSKI, B. 2007. Web search engine evaluation using clickthrough data and a user model. In *Proceedings of the WWW Workshop on Query Log Analysis*.

EFRON, B. AND TIBSHIRANI, R. 1993. *An Introduction to the Bootstrap*. Chapman & Hall/CRC.

FEIGE, U., RAGHAVAN, P., PELEG, D., AND UPFAL, E. 1994. Computing with noisy information. *SIAM J. Comput. 23,* 5, 1001–1018.

FOX, S., KARNAWAT, K., MYDLAND, M., DUMAIS, S., AND WHITE, T. 2005. Evaluating implicit measures to improve web search. *ACM Trans. Inf. Syst. 23,* 2, 147–168.

HE, J., ZHAI, C., AND LI, X. 2009. Evaluation of methods for relative comparison of retrieval systems based on clickthroughs. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management (CIKM'09)*. ACM, New York, NY, 2029–2032.

HOFMANN, K., WHITESON, S., AND DE RIJKE, M. 2011. A probabilistic method for inferring preferences from clicks. In *Proceedings of the ACM Conference on Information and Knowledge Management (CIKM)*.

HUFFMAN, S. B. AND HOCHSTER, M. 2007. How well does result relevance predict session satisfaction? In *Proceedings of the ACM Conference on Research and Development in Information Retrieval (SIGIR)*. 567–574.

JOACHIMS, T. 2002. Optimizing Search Engines Using Clickthrough Data. In *Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining (KDD)*. ACM, New York, NY, 132–142.

JOACHIMS, T. 2003. Evaluating Retrieval Performance using Clickthrough Data. In *Text Mining*, J. Franke, G. Nakhaeizadeh, and I. Renz, Eds., Physica/Springer Verlag, 79–96.

JOACHIMS, T., FREITAG, D., AND MITCHELL, T. 1997. WebWatcher: a tour guide for the World Wide Web. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*. Vol. 1. Morgan Kaufmann, 770–777.

JOACHIMS, T., GRANKA, L., PAN, B., HEMBROOKE, H., RADLINSKI, F., AND GAY, G. 2007. Evaluating the accuracy of implicit feedback from clicks and query reformulations in web search. *ACM Trans. Inf. Syst. 25*, 2, Article 7.

KANTOR, P. 1988. National, language-specific evaluation sites for retrieval systems and interfaces. In *Proceedings of the International Conference on Computer-Assisted Information Retrieval (RIAO)*. 139–147.

KELLY, D. 2005. Implicit feedback: Using behavior to infer relevance. In *New Directions in Cognitive Information Retrieval,* 169–186.

KELLY, D. AND TEEVAN, J. 2003. Implicit feedback for inferring user preference: A bibliography. *Proceedings of the ACM SIGIR Forum 37,* 2, 18–28.

KOZIELECKI, J. 1981. *Psychological Decision Theory*. Kluwer.

KULKARNI, A., TEEVAN, J., SVORE, K., AND DUMAIS, S. 2011. Understanding temporal query dynamics. In *Proceedings of the ACM International Conference on Web Search and Data Mining (WSDM)*. 167–176.

LAMING, D. 1986. *Sensory Analysis*. Academic Press, London.

LIEBERMAN, H. 1995. Letizia: An agent that assists Web browsing. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*. Morgan Kaufmann, 924–929.

LIU, Y., FU, Y., ZHANG, M., MA, S., AND RU, L. 2007. Automatic search engine performance evaluation with click-through data analysis. In *Proceedings of the International World Wide Web Conference (WWW)*. 1133–1134.

MANNING, C. D., RAGHAVAN, P., AND SCHUETZE, H. 2008. *Introduction to Information Retrieval*. Cambridge University Press.

MOE, K. K., JENSEN, J. M., AND LARSEN, B. 2007. A qualitative look at eye-tracking for implicit relevance feedback. In *Proceedings of the Workshop on Context-Based Information Retrieval*, B.-L. Doan, J. M. Jose, and M. Melucci, Eds., CEUR Workshop Proceedings, vol. 326, CEUR-WS.org.

MOOD, A., GRAYBILL, F., AND BOES, D. 1974. *Introduction to the Theory of Statistics* 3rd Ed. McGraw-Hill, New York, NY.

MORITA, M. AND SHINODA, Y. 1994. Information filtering based on user behavior analysis and best match text retrieval. In *Proceedings of the ACM Conference on Research and Development in Information Retrieval (SIGIR)*. 272–281.

OARD, D. W. AND KIM, J. 2001. Modeling information content using observable behavior. In *Proceedings of the Annual Meeting of the American Society for Information Science and Technology*. 28–45.

RADLINSKI, F., BENNETT, P. N., CARTERETTE, B., AND JOACHIMS, T. 2009. Sigir workshop report: Redundancy, diversity and interdependent document relevance. *SIGIR Forum 43,* 2, 46–52.

RADLINSKI, F. AND CRASWELL, N. 2010. Comparing the sensitivity of information retrieval metrics. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, New York, NY, 667–674.

RADLINSKI, F. AND JOACHIMS, T. 2005. Query chains: Learning to rank from implicit feedback. In *Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining (KDD)*. 239–248.

RADLINSKI, F. AND JOACHIMS, T. 2006. Minimally invasive randomization for collecting unbiased preferences from clickthrough logs. In *Proceedings of the Conference of the Association for the Advancement of Artificial Intelligence (AAAI)*. 1406–1412.

RADLINSKI, F., KURUP, M., AND JOACHIMS, T. 2008. How does clickthrough data reflect retrieval quality. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management (CIKM'08)*. ACM, New York, NY, 43–52.

RADLINSKI, F., KURUP, M., AND JOACHIMS, T. 2010a. Evaluating search engine relevance with click-based metrics. In *Preference Learning*, J. Fuernkranz and E. Huellermeier, Eds., Springer, 337–362.

RADLINSKI, F., SZUMMER, M., AND CRASWELL, N. 2010b. Inferring query intent from reformulations and clicks. In *Proceedings of the International World Wide Web Conference (WWW)*. 1171–1172.

SALOJÄRVI, J., PUOLAMÄKI, K., AND KASKI, S. 2005. Implicit relevance feedback from eye movements. In *Proceedings of the International Conference on Artificial Neural Networks (ICANN)*. W. Duch, J. Kacprzyk, E. Oja, and S. Zadrozny, Eds., Springer, 513–518.

SANDERSON, M. AND ZOBEL, J. 2005. Information retrieval system evaluation: Effort, sensitivity and reliability. In *Proceedings of the ACM Conference on Research and Development in Information Retrieval (SIGIR)*. 162–169.

SHAO, J. AND TU, D. 1995. *The Jackknife and Bootstrap*. Springer.

SOBOROFF, I., NICHOLAS, C., AND CAHAN, P. 2001. Ranking retrieval systems without relevance judgments. In *Proceedings of the ACM Conference on Research and Development in Information Retrieval (SIGIR)*. 66–73.

TEEVAN, J., DUMAIS, S., AND HORVITZ, E. 2007. The potential value of personalizing search. In *Proceedings of SIGIR*. 756–757.

TURPIN, A. AND SCHOLER, F. 2006. User performance versus precision measures for simple search tasks. In *Proceedings of the ACM Conference on Research and Development in Information Retrieval (SIGIR)*. 11–18.

VOORHEES, E. AND HARMAN, D. 2005. *TREC: Experiment and Evaluation in Information Retrieval*. MIT press.

VOORHEES, E. M. AND BUCKLEY, C. 2002. The effect of topic set size on retrieval experiment error. In *Proceedings of the ACM Conference on Research and Development in Information Retrieval (SIGIR)*. 316–323.

WANG, K., WALKER, T., AND ZHENG, Z. 2009. PSkip: estimating relevance ranking quality from web search clickthrough data. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'09)*. ACM, New York, NY, 1355–1364.

WHITE, R., RUTHVEN, I., JOSE, J., AND VAN RIJSBERGEN, C. 2005. Evaluating implicit feedback models using searcher simulations. *ACM Trans. Inf. Syst. 23,* 3, 325–361.

WHITE, R., RUTHVEN, I., AND JOSE, J. M. 2002. The use of implicit evidence for relevance feedback in web retrieval. In *Proceedings of the European Conference on Information Retrieval (ECIR)*. F. Crestani, M. Girolami, and C. J. van Rijsbergen, Eds., Lecture Notes in Computer Science, vol. 2291, Springer, 93–109.

YUE, Y., BRODER, J., KLEINBERG, R., AND JOACHIMS, T. 2009. The *K*-armed Dueling Bandits Problem. In *Proceedings of the Annual Conference on Learning Theory (COLT)*.

YUE, Y., GAO, Y., CHAPELLE, O., ZHANG, Y., AND JOACHIMS, T. 2010. Learning more powerful test statistics for click-based retrieval evaluation. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, New York, NY, 507–514.

YUE, Y. AND JOACHIMS, T. 2009. Interactively optimizing information retrieval systems as a dueling bandits problem. In *Proceedings of the International Conference on Machine Learning (ICML)*. 1201–1208.

YUE, Y. AND JOACHIMS, T. 2011. Beat the mean bandit. In *Proceedings of the International Conference on Machine Learning (ICML)*. 241–248.

YUE, Y., PATEL, R., AND ROEHRIG, H. 2010. Beyond position bias: Examining result attractiveness as a source of presentation bias in clickthrough data. In *Proceedings of the International World Wide Web Conference (WWW)*. 1011–1018.

Received February 2011; revised October 2011; accepted December 2011