Evaluating Aggregated Search Using Interleaving

Aleksandr Chuklin*
Yandex & ISLA, University of Amsterdam
Moscow, Russia
a.chuklin@uva.nl

Anne Schuth
ISLA, University of Amsterdam
Amsterdam, The Netherlands
anne.schuth@uva.nl

Katja Hofmann†
ISLA, University of Amsterdam
Amsterdam, The Netherlands
k.hofmann@uva.nl

Pavel Serdyukov
Yandex
Moscow, Russia
pavser@yandex-team.ru

Maarten de Rijke
ISLA, University of Amsterdam
Amsterdam, The Netherlands
derijke@uva.nl

ABSTRACT

A result page of a modern web search engine is often much more complicated than a simple list of "ten blue links." In particular, a search engine may combine results from different sources (e.g., *Web*, *News*, and *Images*), and display these as grouped results to provide a better user experience. Such a system is called an *aggregated* or *federated* search system.

Because search engines evolve over time, their results need to be constantly evaluated. However, one of the most efficient and widely used evaluation methods, *interleaving*, cannot be directly applied to aggregated search systems, as it ignores the need to group results originating from the same source (*vertical* results).

We propose an interleaving algorithm that allows comparisons of search engine result pages containing grouped vertical documents. We compare our algorithm to existing interleaving algorithms and other evaluation methods (such as A/B-testing), both on real-life click log data and in simulation experiments. We find that our algorithm allows us to perform unbiased and accurate interleaved comparisons that are comparable to conventional evaluation techniques. We also show that our interleaving algorithm produces a ranking that does not substantially alter the user experience, while being sensitive to changes in both the vertical result block and the non-vertical document rankings. All this makes our proposed interleaving algorithm an essential tool for comparing IR systems with complex aggregated pages.

Categories and Subject Descriptors

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

*Now at Google Switzerland.

†Now at Microsoft Research Cambridge.

CIKM'13, Oct. 27–Nov. 1, 2013, San Francisco, CA, USA.
ACM 978-1-4503-2263-8/13/10.
http:///dx.doi.org/10.1145/2505515.2505698.

Keywords

Evaluation; vertical search; implicit feedback; A/B-testing

1. INTRODUCTION

In a result page returned by a modern search system some results look different and may be more visually attractive than others. Moreover, results from different subcollections (e.g., News, Images, Finance, Mobile) are usually grouped (i.e., presented adjacent in the ranking) to improve the search result browsing experience. These results are often called *vertical* documents. If the vertical results are grouped, we call such group a *vertical block*. In Figure 1 we provide a schematic picture of two document lists containing vertical blocks; the vertical block occupies positions 3 to 5 in ranking A and positions 4 to 5 in ranking B.

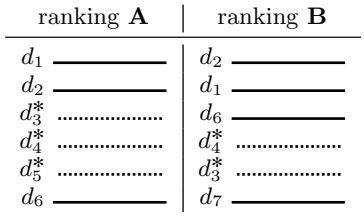

Figure 1: Two rankings with a vertical block present. Vertical documents are shown as dotted lines and also marked with *.

There is an efficient way of comparing two rankings called *interleaving* [4]: it produces an interleaved ranked list out of rankings A and B, shows it to the user and then infers user preferences from their clicks. However, if we want to interleave ranked lists from Figure 1 using *balanced*, *team-draft* or *probabilistic* interleaving (see [15], [19] and [11], respectively), we may end up in a situation where the resulting interleaved ranking L has vertical documents mixed with regular documents. That is, those interleaving methods do not respect the grouping of vertical results. As was found by Dumais et al. [9], this can significantly alter the user experience, which violates one of the core principles of user-based evaluations formulated by Joachims [15].

The main research questions we address in this paper are:

1. Can we perform interleaving measurements in a way that respects grouping of the vertical results and does

not change the user experience? How does the quality of the interleaved list compare to the quality of the aggregated result pages being interleaved?
2. How does an interleaving algorithm that respects grouping compare to state-of-the-art interleaving algorithms and A/B-testing? To which extent do they agree?
3. Does a comparison based on a vertical-aware interleaving method represent a fair and unbiased comparison, or does it erroneously infer any preference for a randomly clicking user? Can it capture quality differences between rankings as well as conventional interleaving methods, while preserving vertical blocks?

The main contributions of this paper are answers to these questions, which includes a proposed vertical-aware interleaving method and the corresponding experiments.

We outline related work in Section 2. Section 3 is dedicated to the design of an interleaving method that respects grouping of vertical results. Sections 4 and 5 describe our experiments to compare the proposed and non vertical-aware interleaving algorithms. In Section 6 we sketch and discuss additional possibilities for designing vertical-aware interleaving algorithms, after which we conclude in Section 7.

2. RELATED WORK

2.1 Interleaving

While the traditional Cranfield approach [8] to ranking evaluation is still widely used, there is a growing trend to use implicit feedback to evaluate and learn ranking models. Starting from the work of Joachims [14, 15] the idea of interleaving methods has become increasingly popular.

Two widely used interleaving algorithms are *team-draft interleaving* (TDI) [19] and *balanced interleaving* (BI) [15]. TDI can be described as follows. For each user query we build an interleaved list L whose documents are contributed by rankings A and B (rankings that we want to compare). This combined list is then shown to the users and the users' clicks are recorded. The system that contributes most of the documents clicked by the user is inferred to be the winner for the particular query-user pair; the system that wins for most such pairs is then considered to be the better system. Balanced interleaving uses a different algorithm to build an interleaved list L and a more sophisticated procedure for determining the winner. These algorithms, as well as their modifications, were extensively studied in [4]. Our vertical-aware interleaving method is based on TDI.

In addition, there are recent methods that explore new evaluation approaches. Hofmann et al. [11] propose a method called *probabilisitic interleaving* (PI) that allows more interleaved lists to be shown to the user. This method is reported to be unbiased and sensitive to even small ranking changes. Radlinski and Craswell [18] propose to choose possible interleaved lists and their probabilities as an optimization problem (*optimized interleaving* (OI)). We sketch a solution for making OI vertical aware in our discussion in Section 6.

2.2 Aggregated Search

Dumais et al. [9] find that grouping results of the same type improves the user experience. Following these findings, many commercial search engines provide this functionality in their search result page interface. The search engine that introduced this *aggregated* or faceted search early on was Naver; there, result pages allow many more than 10 results per query, and results are always grouped [21]. Yahoo! and Bing historically inserted blocks of vertical documents on fixed positions (slots) in addition to the organic web search documents [2, 17]. Yandex allows vertical blocks to appear in any position [6], and the same appears to hold for Google.

Evaluation of aggregated search was often viewed as an offline task for human editors. Arguello et al. [2] propose to use pairwise preference evaluation to rank the vertical blocks for a given query. Ponnuswami et al. [17] describe the process of manual assessments of both vertical and organic documents. However, they neither discuss the combined effect of these two ranking aspects, nor suggest a way to compare vertical documents inside one block to each other. They also propose a click-based assessment of the vertical block placement similar to Joachims et al. [16].

Most of the evaluation methods (apart from [22]) assume that the vertical block is atomic, i.e., it is considered to have no internal structure. However, this is not always the case. For example, a news block usually contains a list of headlines. Each of them can be judged separately and compared to organic results. One possible approach for evaluating results with vertical blocks could be to use intent-aware metrics such as nDCG-IA, ERR-IA [1] or α-nDCG [7]. If we have relevance judgements for different user needs (*intents*) for both vertical and non-vertical documents, we can then compute one value representing the quality of the whole result page.

Our work differs in important ways from the work just discussed. In contrast to previous work on interleaving, we propose an interleaving method that preserves the user experience on complex aggregated search engine result pages. In contrast to previous work on evaluating aggregated search, we base our algorithm on an efficient interleaving algorithm, which was proven to accurately infer user preference from implicit feedback.

3. VERTICAL-AWARE INTERLEAVING

We describe an algorithm that may be viewed as a generalization of the TDI method by Radlinski et al. [19]. The intuition is to start with the TDI procedure and then alter it to meet the following requirements:

1. Both of the systems being interleaved should contribute to the vertical block placement size and position in the interleaved list;
2. Both systems should contribute vertical and organic web documents;
3. Team assignment should be "fair"; and
4. The resulting interleaved list should not degrade the user experience.

While formalizing these criteria is an interesting question by itself, we leave it for future work.

We propose a method called *vertical-aware team-draft interleaving* or VA-TDI for short (Algorithm 1). The main idea is to enforce the grouping of vertical documents. Therefore, our algorithm proceeds like TDI until it hits the first vertical document. After that it interleaves only vertical documents (line 23) until the block has been formed (line 13), i.e., there are no vertical documents left or the desired block

Algorithm 1 Vertical-Aware Team-Draft Interleaving (VA-TDI). "First Vertical Document Starts the Block".

```
1: function VaTdi1(ranking A, ranking B)
2:     L ← []; Team_A ← ∅; Team_B ← ∅
3:     A_v ← {d ∈ A | d is a vertical doc}
4:     B_v ← {d ∈ B | d is a vertical doc}
5:     Size_A ← |A_v|; Size_B ← |B_v|
6:     Size_L ← SampleSmoothly(Size_A, Size_B)
7:     InsideBlock ← False; AfterBlock ← False
8:     while |L| < N do
9:         if |Team_A| < |Team_B| + RandBit() then
10:            AddNextDocFrom(A)
11:        else
12:            AddNextDocFrom(B)
13:        if |{d ∈ L | d is a vertical doc}| = Size_L or
       "unable to add document at line 25" then
14:            InsideBlock ← False; AfterBlock ← True
15:    return L

16: function SampleSmoothly(integer a, integer b)
17:    if a > b then
18:        Swap(a, b)
19:    Sample x randomly from [a − 1, b + 1] where all in-
       tegers from [a, b] have equal probability p; (a − 1) and
       (b + 1), if in allowed range, each has probability p/2
20:    return x

21: procedure AddNextDocFrom(ranking X)
                                        ▷ X is either A or B
22:    if InsideBlock then
23:        X_left ← {i | X[i] ∈ X_v\L}
24:        if X_left = ∅ then
25:            return                  ▷ unable to add document
26:    else if AfterBlock then
27:        X_left ← {i | X[i] ∈ X\(X_v ∪ L)}
28:    else                                          ▷ before block
29:        X_left ← {i | X[i] ∈ X\L}
30:    k ← min X_left
31:    Team_X ← Team_X ⋃{X[k]}     ▷ add the document
       to the team
32:    if X[k] is a vertical doc then
33:        InsideBlock ← True
34:    L ← L + X[k]                ▷ append the document to L
```

size is reached.[1] After that, the algorithm continues interleaving non-vertical documents (line 27).[2]

If we look back to our original goals, we see that Algorithm 1 explicitly chooses a block size between those of A and B (with some smoothing in order to do explaration), while the position of the block is contributed implicitly (although both systems can influence it). Requirements 2 and 3 are met automatically due to the TDI procedure that we re-use (after one system wins the coin flip and contributes the document, another system has to contribute the next document), though we also verify them along with requirement 4 in the next section.

To summarize, we have introduced an interleaving algorithm that respects vertical blocks. In the next two sections we evaluate this algorithm experimentally, first on log data in the next section and then using simulations in Section 5.

4. EXPERIMENTS WITH LOG DATA

When answering the research questions outlined in the introduction, we would like to experiment with real user clicks. For this purpose we adopt the setup proposed by Radlinski and Craswell [18] that makes use of historic user clicks in order to evaluate new interleaving methods (4.1). The main idea is to look at queries with sufficient variation in the ranked lists and then pretend that these different rankings are the interleaved lists of rankings produced by some rankers A and B. One of the limitations of this approach is that we cannot experiment with completely new document orders that are disallowed by the current production ranking algorithms. In particular, we cannot reproduce the outcomes of an interleaving algorithm that does not respect vertical blocks. Another problem is that the data we get using this method is skewed towards a relatively small number of highly frequent queries. Ideally, we would like to verify our findings using another set of experiments, which does not have these limitations. For this purpose we also use a click log simulation that allows for experiments with a larger amount of data and a broader variety of interleaving algorithms; see Section 5. One should not completely rely on the click simulation, however, because simulated user clicks may not reflect certain aspects of real user behavior.

In this section we address research questions RQ1 and RQ2 outlined in the introduction. In particular, we compare our interleaving algorithm to the A/B-testing method. Even though A/B-testing by itself has proved to be inefficient in terms of the amount of data it needs to make a decision [4], comparing different evaluation metrics to absolute click measures is still quite common [3]. We also look at how the outcomes of the proposed interleaving algorithm are different for the vertical and organic results and compare the latter with the TDI experiment when no vertical block is present. As a sanity check we also show that our way of interleaving two lists containing vertical blocks does not lead to a significant degradation in quality.

4.1 Experimental Setup

For our experiments we used a 2-months click log of the Yandex search engine. We only kept queries for which we have a vertical of mobile apps present in a result page. This vertical is triggered for queries with an information need related to smartphone or tablet[3] applications. While the algorithm used for information need detection is beyond the scope of the current paper, it is useful to name a few examples of queries leading to this vertical. These include queries containing a mobile application name (e.g., "cut the rope") or explicit markers of mobile platforms (e.g., "opera for iphone"). The result items in the vertical block are presented using a standard text snippet, enhanced with an application icon, price and rating. An example of such a snippet is shown in Figure 2.

[1]We pick the desired block size beforehand (line 6) to ensure requirement 1 is met.

[2]We also implemented a more complicated variant of the algorithm to handle multiple verticals, but we do not deal with this option in the current paper.

[3]That is, iOS or Android devices.

CityGuideDeals – **San Francisco**… — free

☆☆☆☆☆ no votes in the App Store

FREE **San Francisco** coupon discount app from BayCityGuide.com! **San Francisco**'s largest and most successful visitor **guide** has partnered with its advertisers to bring you these special...

itunes.apple.com QR code

Figure 2: Result item of the mobile application for the query "cityguide San Francisco iPhone."

Table 1: Filtering parameters setup

	N	K	M
Radlinski and Craswell [18]	4	10	4
current work	10	4	2

Let us call a query together with a result list a *configuration*. Each configuration that has been shown at least once to a user counts as an *impression*. Each impression has zero or more clicks.

In our evaluation setup we use an approach similar to the one taken by Radlinski and Craswell [18]. We sample queries that have sufficient variation in result lists and then assume that some of these rankings were interleaved lists of hypothetical A and B rankings. Specifically, we proceed in the following steps:

1. Keep only impressions that have at least one click on their top-N documents.
2. Keep only configurations that have at least K such impressions.
3. Keep only queries that have at least M such configurations.

After that, we name two configurations to be rankings A and B for each query. Following Radlinski and Craswell [18], we call the most frequent configuration *ranking* A and the one that differs at the highest rank *ranking* B. In case we have several candidates for B we choose the most frequent one. Once we have our rankings A and B, we compute all possible interleaved lists that can be produced by Algorithm 1 and proceed with the filtering:

4. Keep only queries for which we have all interleaved lists that can be produced by VA-TDI in the log.

In order to fully define the experimental setup we have to define the parameters K, M and N. We summarize the parameters we use in Table 1. Unlike [18] we cannot use only the top-4 documents as this is highly likely to be in-between the vertical block. This is why we are forced to decrease K and M in order to have a sufficient amount of log data. With these parameters we have 814 unique queries which we use in all the experiments in this section. If we relax the last filtering step and only require at least one interleaved list to be present in our query log, we obtain $5,755$ queries to experiment with (we consider the missing interleaved lists as ties). The reason to consider this relaxed setup is the following: if we require all possible interleaved lists to be present in the click log, we risk ending up in a situation where only queries having 2–4 interleaved lists are left (i.e., queries with very similar rankings A and B).

Given the click log data we compute the following values for each query:

- The average difference of the absolute click metrics for rankings A and B.
- The interleaving score[4] for each impression of each ranking allowed by the interleaving algorithm. As some configurations might have more impressions than the interleaving algorithm suggests, we normalize the scores to the correct probabilities as implied by the interleaving algorithm. Specifically, we compute the average score for each configuration, multiply it by the probability of such a configuration and then average across all found configurations similar to Radlinski and Craswell [18].
- The interleaving scores for the vertical documents and non-vertical documents separately.

As absolute metrics we use the metrics that are often used in A/B-testing experiments. We decided to use metrics that only require clicks and no additional information (like relevance judgements, user information, timestamps or session information): $MaxRR$, $MinRR$, $MeanRR$, PLC, $Clicks$@1. These metrics were also the best at identifying system superiority in the experiments performed by Chapelle et al. [4], who degraded system quality and looked at how often different metrics point in the right direction.

- $MaxRR$, $MinRR$, $MeanRR$ — maximum, minimum and mean reciprocal rank ($1/r$) of clicks.
- PLC (Precision at Lowest Click) — the number of clicks divided by the rank of the lowest click. This is the opposite of $pSkip$ used in [4].
- $Clicks$@1 — equals 1 if there was a click on top-1 document, 0 otherwise.

We also add one vertical-specific metric *VertClick*, which equals 1 if there was a click on a vertical document and 0 otherwise (15% of the documents in our dataset were vertical documents). This metric tends to score low for two reasons: (1) the vertical we study is often placed at the bottom of the result list; (2) the highly frequent queries we study can usually be answered by the top (often non-vertical) documents.

4.2 Results of Re-using Click Log Data

First, we analyze the impact of VA-TDI on the user experience (RQ1). Even though we explicitly group vertical documents, we might still get a result list of lower quality and we have to make sure that this is not the case. Second, we report on how the outcomes of VA-TDI agree (RQ2) with two commonly used evaluation methods: A/B-testing and non vertical-aware interleaving (here: TDI).

4.2.1 Impact on User Behavior

He et al. [10] stated that one of the main aspects when evaluating interleaved comparison methods should be the interleaved result lists' utility to users. Ideally, we want an interleaving method to produce ranked lists that are not worse than those of A and B (RQ1). Unlike He et al. [10] we are not using editorial judgements. The main reason is that

[4]We assume that the score is 1 if ranking A wins a particular impression, -1 if B wins and 0 if they tie.

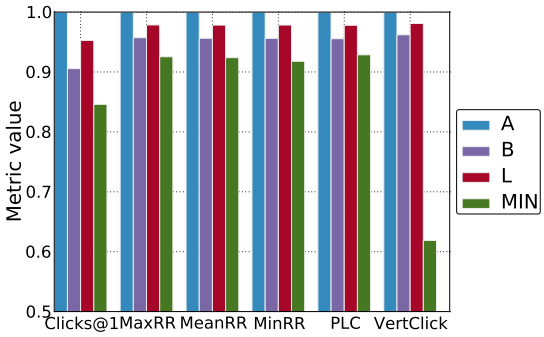

Figure 3: Absolute click metrics for the rankings A and B, for the interleaved list L and the *MIN* system (worst of A and B for each query).

the way we collect the data already binds us to queries that are very likely not to be judged by human editors since the overlap between the queries that a search engine has in order to train and evaluate its ranking algorithms and a random sample of queries from the query log is usually very small. At the same time, judging a large set of query-document pairs is too costly to do every time we need to justify changes in our online evaluation algorithm. Moreover, there are no standard guidelines on how to assess vertical documents and no universally accepted offline measures for the result pages containing vertical documents. On the other hand, we have already collected a click log in the previous step and can now use it to compare the interleaved list L to the original rankings A and B using absolute click metrics.

For each query we compute the average value of the click metrics for the ranked lists A and B as well as for all interleaved lists L. The results are summarized in Figure 3. As we are not interested in the absolute values of the metrics (and are not able to disclose them due to the proprietary nature of such information), we normalize all the metrics to the corresponding values of the system A. Most of the changes are not statistically significant when using a 99% Mann-Whitney U test, except for *VertClick* whose value for L is statistically significantly higher than that of B and lower than that of A. We also determined the worst system (A or B) for each individual query and showed the average metric values as *MIN*. The absolute click metrics for *MIN* are all statistically significantly lower than those of L.

We can conclude that the click metrics for the interleaved list L are always between those of A and B. This means that we do not have any degradation in user experience compared to the worst of two systems (B in this case). We should mention that we do have a degradation (not significantly) compared to the best system (A), but we cannot avoid that since we do not know which system is superior beforehand. It would be interesting to analyze various interleaving algorithms from the point of view of *balancing exploration and exploitation* (see, e.g., [12]). This, however, is beyond the scope of the current work.

4.2.2 Agreement with Other Evaluation Methods

In order to compare the direction of preference indicated by the absolute click metrics and the interleaving measures (RQ2) we split the data into six buckets corresponding to six equal time periods of ten days (t_1, t_2, t_3, t_4, t_5, t_6) and compute the weighted average of the absolute and interleaving metrics. The outcome for each impression (positive if A wins, negative if B wins, zero if it is a tie) is multiplied by the total frequency of the query[5] and summed up over all the queries. We report the winning system according to each measure in Table 2. Note that we consider three ways of interpreting clicks in the interleaving method: *total* — all clicks are counted, *organic only* — only the clicks on non-vertical documents are taken into account, and *vertical only* — only the clicks on vertical documents are counted. For example, if we want to evaluate only changes in the organic ranking we may want to look at *organic only* results. On the other hand, we risk having an unbalanced team assignment if we just skip the vertical block.

Table 2: Agreement between A/B-testing measures and VA-TDI. All changes are statistically significant except for ones marked by $\diamond$.

Measure	t_1	t_2	t_3	t_4	t_5	t_6
Absolute Metrics						
Clicks@1	B$\diamond$	B	B	A	A	B
MaxRR	A	B	B	A	A	B
MeanRR	A$\diamond$	B	B$\diamond$	A	A	B
MinRR	A$\diamond$	B	A$\diamond$	A	A	B
PLC	A	B	B$\diamond$	A	A	B
VertClick	B	B	B	B	B	B
Interleaving Algorithms						
total	B	B	B	A	A	B
organic only	B	B	B	A	A	B
vert only	A	A$\diamond$	A$\diamond$	B	B	B

Table 2 shows that in most cases VA-TDI (*total* and *organic only*) agrees with the majority of the absolute metrics. Similar to what was reported in [19], the cases of disagreement between absolute click metrics and interleaving outcomes are always accompanied by the lack of statistical significance. We mark such cases where the winning system is not statistically significantly better with the $\diamond$ sign.[6] We can also note that the agreement between the *vertical-only* interleaving measure and the *VertClick* absolute metric is low. Again, two of the three cases with disagreement do not detect a statistically significant preference. That means that either *VertClick* is too simple to correctly capture the ranking changes or *vertical-only* is not the right way to interpret interleaving outcomes (or both).

We also look at per-query correlations between interleaving measures and absolute click metrics. For the i-th query, let x_i be the value of some absolute metric, y_i the value of an interleaving measure and n_i the total query frequency. Following Chapelle et al. [3], where similar sets of absolute click metrics were used, we use a weighted correlation:[7]

$$Corr(x, y, n) = \frac{\sum_i n_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i n_i (x_i - \bar{x})^2} \sqrt{\sum_i n_i (y_i - \bar{y})^2}},$$

where

$$\bar{x} = \frac{\sum_i n_i x_i}{\sum_i n_i}, \; \bar{y} = \frac{\sum_i n_i y_i}{\sum_i n_i}.$$

[5]Remember that we previously normalized configurations to the correct probabilities.

[6]Here and below we use a bootstrap test with 1000 bootstrap samples and a significance level of 99%.

[7]Because we may miss some configuration when simulating interleaving we weight queries, not configurations.

Table 3: Correlation of VA-TDI and absolute click metrics. The upper left index here means that the values are computed using the *combined* dataset featuring both organic web (W) and vertical (V) documents.

	${}^{(W+V)}Clicks@1$	${}^{(W+V)}MaxRR$	${}^{(W+V)}MeanRR$	${}^{(W+V)}MinRR$	${}^{(W+V)}PLC$	${}^{(W+V)}VertClick$
${}^{(W+V)}total$	0.746 ± 0.006	0.684 ± 0.006	0.709 ± 0.007	0.662 ± 0.008	0.714 ± 0.007	0.048 ± 0.008
${}^{(W+V)}organic\ only$	0.728 ± 0.007	0.672 ± 0.008	0.704 ± 0.008	0.665 ± 0.009	0.696 ± 0.007	-0.060 ± 0.011
${}^{(W+V)}vertical\ only$	-0.005 ± 0.012	-0.026 ± 0.015	-0.058 ± 0.009	-0.087 ± 0.008	-0.001 ± 0.012	0.472 ± 0.026

Table 4: Weighted correlation of VA-TDI with click metrics and TDI for the *combined* (W+V) and *vertical-free* (W) datasets. The upper left index indicates which dataset is used to compute the corresponding metric.

	${}^{(W)}Clicks@1$	${}^{(W)}MaxRR$	${}^{(W)}MeanRR$	${}^{(W)}MinRR$	${}^{(W)}PLC$	${}^{(W)}TDI$
${}^{(W)}TDI$	0.606 ± 0.015	0.650 ± 0.013	0.894 ± 0.006	0.927 ± 0.005	0.787 ± 0.012	1
${}^{(W+V)}total$	0.501 ± 0.042	0.576 ± 0.030	0.570 ± 0.019	0.536 ± 0.017	0.608 ± 0.034	0.649 ± 0.015
${}^{(W+V)}organic\ only$	0.515 ± 0.044	0.590 ± 0.029	0.589 ± 0.019	0.555 ± 0.016	0.617 ± 0.033	0.665 ± 0.017

The results are summarized in Table 3. We can see that VA-TDI (*total* and *organic only*) is correlated with all conventional absolute metrics (except for *VertClick*), i.e., it measures the right thing. And the fact that an interleaving method, as a rule, needs much less data than A/B-testing [19] makes it a useful tool for comparing ranking algorithms. We also confirm that the *vertical only* method correlates only with the vertical metric *VertClick*. More interestingly, only *vertical only* is correlated with the user preferences within the vertical block: the *total* interleaving method shows a small positive correlation with the *VertClick* metric, while *organic only* is even negatively correlated with this metric. We can conclude that our interleaving algorithm (*total*) is indeed sensitive to changes in the vertical ranking, but the main signal originates from the organic ranking.

We also analyze the relaxed setup discussed at the end of Section 4.1, where we keep all queries for which at least one possible interleaving is observed in the log. The results are very similar to the ones shown above, so we do not include them in the paper. However, the fact that they are similar supports the validity of our experimental setup, and confirms that we do not introduce any additional bias by requiring all the possible lists to be in a click log.

For each query we also look for configurations that miss the vertical block, but share the same top-4 non-vertical documents with rankings A or B.[8] These situations are mainly due to different experiments running in parallel (with some of them experimenting with excluding vertical documents from search results) and system instability. We treat these configurations without vertical blocks as interleaved lists of the organic documents of the systems A and B defined above. By doing so we can see how the presence of the vertical block influences user behavior on the non-vertical documents. After all the filtering steps we have 820 unique queries featuring both *combined* and *vertical-free* impressions.[9] This additional, *vertical-free* data set for the same queries allows us to answer the following question:

- Do different online experiments agree when comparing general web results with and without a vertical block?

We answer this question by comparing the outcomes of VA-TDI to those obtained in the *vertical-free* setup.[10] We compute the weighted correlation of the interleaving outcomes on the *combined* dataset to the TDI outcomes and absolute click metrics on the *vertical-free* dataset.

The results are presented in Table 4. We can see that the *organic only* results are slightly better correlated with the *vertical-free* dataset metrics than *total*; however, none of these differences are statistically significant. This means that when we want to evaluate changes in the organic web ranking only, we can either skip clicks on the vertical documents or treat them as usual. The first approach (*organic only*) is closer to what we want to measure, but might potentially suffer from uneven team assignments (cf. Algorithm 1). For comparison we also included the correlation of the TDI algorithm to the absolute click metrics computed on the same *vertical-free* dataset (we still use weights from the *combined* dataset). We see that the correlation between VA-TDI and TDI in a *vertical-free* setup is not perfect. However, given that the filtering procedure requires only the top-4 documents to match between both interleaved rankings, we would not expect a perfect correlation.

To summarize, we have shown that:

- VA-TDI does not degrade the user experience compared to either ranking A or B (Figure 3).
- VA-TDI is reasonably correlated with TDI and conventional absolute click metrics (A/B-testing metrics) (Tables 2, 3, 4).
- The outcomes of VA-TDI are influenced by changes in the vertical block, but interleaving of *vertical-only* results best correlates with *vertical-only* click metrics (Table 3).

We conclude that VA-TDI can be applied to comparing two ranking systems in situations where only the organic web ranking has changed, only the vertical ranking or placement has changed, or they have both been altered.

5. SIMULATION EXPERIMENTS

Our experiments using real-life interaction data provide insights into the performance of VA-TDI in one specific setting. To assess VA-TDI under a wider range of conditions, we also address research questions 1, 2 and 3 (formulated in the introduction) using a simulation setup. In contrast

[8]Here we mimic the setup of Radlinski and Craswell [18] to obtain a reasonable amount of data (see Table 1). If we required a perfect match we would obtain less than 100 unique queries.

[9]These queries correspond to $1,293$ configurations allowed by our interleaving method. Of these 820 queries, 72 have all possible interleaving lists present in the filtered log.

[10]Not to be confused with the *organic only* computation scheme.

to our experiments on real log data (presented in Section 4) our simulation experiments allow us to generate a wide range of result lists, without the risk of hurting the user experience in a production system. We test several vertical block sizes, several block placements and different levels of relevance within the block. We compare VA-TDI to TDI [19].[11]

5.1 Experimental Setup

Our simulation approach is based on the experimental setup proposed in [11, 13]. Briefly, we first generate two ranked lists with blocks of vertical documents, then apply an interleaving method, and subsequently offer the interleaved list to a simulated user that produces clicks. Then it is up to the interleaving method to select a winning ranker. We measure how often the interleaving algorithm correctly identifies the correct winner and in how far an interleaving algorithm degrades the user experience.

5.1.1 Generating Synthetic Rankings

More precisely, we start with a synthetic list of organic documents, and randomly designate 1–3 of the organic documents as relevant to the simulated users' query. Then, two permutations of these documents are selected at random. We insert blocks of vertical documents in both rankings under several conditions. These blocks vary in size from 0 to 8 documents and in the position of the block. This position is based on the distribution reported by Chen et al. [5],[12] and is either (1) the same for both rankings (condition *dependent*); or (2) selected independently (condition *independent*). The ranking of the vertical documents within the block is always fixed, i.e., the same for both rankings. Vertical documents are either (1) non-relevant (condition *non relevant*); or (2) a number of them, proportional to the relevant non vertical documents, is relevant (condition *relevant*).

We repeat this process of generating two rankings and inserting vertical blocks until one ranking Pareto-dominates[13] the other in terms of how it ranks relevant documents. Because the rankings are constructed in such a way that one dominates the other, we know which ranking should be preferred by an interleaving algorithm.

5.1.2 Simulating Clicks

In order to see whether an interleaving algorithm indeed prefers the ranking that is known to be better, we do the following. Given the two generated rankings, we apply interleaving to generate an interleaved result list that is presented to the user. We simulate users' click behavior on the interleaved list using click models. Simulated users are always presented with the top 10 documents from the interleaved list.

We have two types of user simulations. First, the *random click model* (RCM) assumes that users click on each document in the presented ranking with probability 0.5, such that—in expectation—half the documents are clicked. Relevance or presentation of documents is not taken into account. This model is used to answer RQ3. Second, the *federated click model* (FCM), used to answer RQ2, implements the attention model in [5]. This click model is designed to capture user behavior when result pages contain vertical documents. It assumes that blocks of these vertical documents attract users' attention. We instantiate FCM in the following way, where $\mathcal{A}$ denotes the attention bias, and $\mathcal{E}$ denotes the examination probability:

$$P(\mathcal{A}=1) = P(\mathcal{A}=1|pos_v) = hpos_v \quad (1)$$
$$P(\mathcal{E}_i=1|\mathcal{A}=0) = \phi_i \quad (2)$$
$$P(\mathcal{E}_i=1|\mathcal{A}=1) = \phi_i + (1-\phi_i)\beta_{dist} \quad (3)$$
$$P(\mathcal{C}_i=1|\mathcal{E}_i=0) = 0 \quad (4)$$
$$P(\mathcal{C}_i=1|\mathcal{E}_i=1) = r_i, \quad (5)$$

The attention bias $\mathcal{A}$ in (1) depends only on pos_v, the position of the highest vertical document in the ranking. Also, $hpos = [.95, .9, .85, .8, .75, .7, .3, .25, .2, .15]$; we assume the fold to be after document 6 (i.e., the user can usually see 6 documents without scrolling the result page). In the absence of the attention bias, in (2), the probability $\mathcal{E}_i$ of examining document i depends only on its position. We use $\phi = [.68, .61, .48, .34, .28, .2, .11, .1, .08, .06]$, based on the fixations reported in [16]. If there is attention bias ($\mathcal{A}=1$), in (3), the probability of examining document i, $\mathcal{E}_i$, is calculated using both ϕ_i and β_{dist}, where *dist* is the distance to the nearest vertical document. We take β_{dist} such that resulting clicks resemble those reported in [5] and such that the click model in expectation slightly prefers rankings with vertical documents present over rankings without.

$$\beta_{dist} = \begin{cases} 1 & \text{if } dist = 0 \\ \frac{1}{|dist|+.1} & \text{otherwise.} \end{cases}$$

Then, a document i that is examined and relevant ($r_i = 1$) is always clicked ($\mathcal{C}_i = 1$), see (5).

5.1.3 Measurements

We generate 500 pairs of rankings, with one ranking dominating the other, as described above. These pairs are each interleaved 500 times by both VA-TDI and TDI. We repeat this process for several combinations of the conditions described in Section 5.1.1. We observe the portion of correctly identified ranking preferences (i.e., the accuracy) by each interleaving method. We calculate the mean and 95% binomial confidence bounds.

We measure the impact on the user experience by measuring the number of vertical blocks in the interleaved list. Consecutive vertical documents count as one block. Finally, we assess bias in terms of the number of incorrectly detected statistically significant preferences under random clicks.

5.2 Results of Simulation Experiments

We describe three sets of experiments. We validate that VA-TDI preserves the quality of the interleaved list and compare this to non vertical-aware algorithms (5.2.1). We assess the outcomes inferred by VA-TDI (5.2.2). We examine whether bias is introduced by vertical-awareness (5.2.3).

[11] The code of our simulation experiments, including a reference implementation of VA-TDI, is made publicly available at `https://bitbucket.org/ilps/lerot` [20].

[12] We take the distribution from Figure 2 in [5] and scale it back to the [1,10] interval.

[13] As in [13], we re-rank documents by examination probability $P(\mathcal{E}_i = 1)$. In [13], $P(\mathcal{E}_i = 1)$ is implicitly defined by the *cascade click model*. In our case, it is dictated by the *federated click model* [5]; we marginalize over $\mathcal{A}$, using (1), (2) and (3) below. We say that ranking A *dominates* B if and only if—re-ranked with $P(\mathcal{E}_i = 1)$—it ranks all relevant documents at least as high as B and at least one relevant document higher than B.

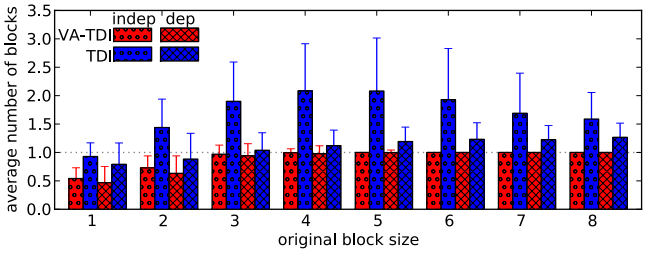

Figure 4: The number of blocks of vertical documents for block sizes 1–8 for VA-TDI (red) and TDI (blue, higher) under the dependent (×) and independent (○) block placement conditions. Error bars correspond to one standard deviation.

5.2.1 Impact on the User Experience

We first turn our attention to the impact of interleaving on the user experience (RQ1). Here, we assume that the major factor impacting the user experience is the number and size of vertical blocks. While vertical-aware interleaving is designed to generate interleaved result lists with only up to one vertical block, other methods may break up the block into individual results (which can be considered smaller blocks). Thus, we measure the effect of interleaving methods on the user experience in terms of the number of blocks that vertical results are broken up into during interleaving.

Figure 4 shows our results. Two methods are compared under the *dependent* and *independent* conditions (see Section 5.1.1). For the two vertical-aware runs, independent and dependent VA-TDI, we see that the number of generated vertical blocks is typically close to 1, as designed. When vertical blocks in the lists we interleave are small, the block may not be included in the interleaved list, resulting in an average number of blocks smaller than 1 (cf. Algorithm 1, line 16). This can also occur when the vertical blocks are placed in the lower halves of the original lists.

For TDI, we observe different behavior under dependent and independent block placement. When the compared lists place vertical blocks independently, TDI tends to generate several smaller blocks. The largest number of blocks is generated for original block sizes 4 and 5. These are split into 2 blocks on average. Variance is high; the minimum number of blocks shown is 1 and the maximum is 5 blocks. For all block sizes greater than 1, TDI with independent block placement produces a substantial number of impressions for which the vertical block is split up (52–67%). Under dependent block placement, the number of blocks generated by TDI is much lower. For large blocks of size 4 to 8 the vertical block is split up in 14–27% of the cases, respectively. The remaining impressions produce only one vertical block.

We conclude that VA-TDI keeps vertical documents together, as designed. It produces up to one vertical block per result list, thus bounding the impact of interleaving on the user experience. When vertical blocks are placed independently, the impact of TDI without vertical awareness is high. However, when blocks are placed at the same positions, the impact on the user experience is much lower (but still substantially higher than for VA-TDI).

5.2.2 Agreement with Other Evaluation Methods

Our second experiment measures to what degree VA-TDI can detect differences in the quality of result lists. This is, of course, the key test for our new interleaving method.

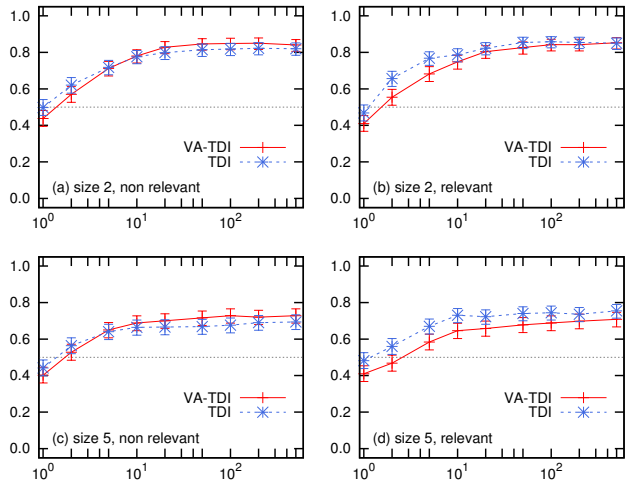

Figure 5: Portion of correctly identified ranker preferences (vertical axis) by VA-TDI (red, solid) versus TDI (blue, dashed) after 1–500 user impression (horizontal axis, log-scale). The dashed horizontal line at 0.5 denotes random preference. All figures have independent block placement. Error bars correspond to binomial confidence intervals.

We compare our vertical-aware interleaving method VA-TDI to the non-vertical-aware baseline TDI in terms of the accuracy of the identified ranker preferences (RQ2). Figure 5 shows the portion of correctly identified ranker preferences by VA-TDI and TDI after 1 to 500 user impressions modeled by the federated click model FCM (see Section 5.1.2). Figure 5(a) shows results averaged over all possible positions of a block of size 2 (independent block placement) under the assumption that none of the vertical documents are relevant. We see that TDI and VA-TDI converge to correctly identify 82% and 84% of the true preferences correctly, respectively. There is no significant difference in the number of impressions the two methods need to make these comparisons.

Figure 5(b) shows results in the setting with relevant vertical documents. Again, both TDI and VA-TDI converge to the same level of accuracy after observing 10–20 impressions. VA-TDI initially requires more sample data on average: TDI is significantly more accurate when we have a really small number of impressions (less than 10). We believe that the reason for this is the noise added by the drop-out of the relevant vertical documents when doing exploration on line 16 of Algorithm 1. Since this is noise—and not bias towards either ranking—this levels out as the number of observed impressions increases. However, this need for more samples is a small loss in efficiency for a method that preserves the original user experience as much as possible.

Figure 5(c) and (d) show results for the same conditions as (a) and (b), respectively, but with a block size of 5 instead of 2. The number of correctly identified preferences drops significantly to around 70%, which is a trend we also observe for other block sizes. This drop is due to the fact that the within block ordering is the same for both ranking A and B, making them less different and thus less distinguishable when the block size goes up. Note that this is an artifact stemming from the way we construct these rankings. Also with these larger block sizes, VA-TDI needs more impressions than TDI to reach the same level of accuracy when we have relevant vertical documents (Figure 5(d)).

Table 5: Percentage of significant differences between rankers detected under the random click model RCM for VA-TDI and TDI for $p < 0.05$ on 500 ranker pairs after 100–500 user impression (left column) under all combinations of conditions for block size 2. With $p < 0.05$, an interleaving method is expected to detect around 5% significant differences. Only for TDI for the relevant dependent condition, this is significantly higher (*) after 100 impressions.

	non relevant				relevant			
	dependent		indep.		dependent		indep.	
	VA	TDI	VA	TDI	VA	TDI	VA	TDI
100	4.4%	5.8%	5.2%	3.6%	4.2%	7.6%*	4.0%	5.0%
200	3.2%	4.4%	5.0%	5.4%	5.4%	7.2%	5.0%	4.4%
300	3.8%	4.0%	4.4%	3.4%	4.8%	6.4%	5.0%	5.8%
400	2.6%	6.8%	5.4%	4.6%	4.2%	5.2%	5.6%	4.2%
500	4.2%	5.8%	5.8%	5.2%	5.8%	6.6%	4.6%	4.0%

Our results on simulated result lists and interaction data confirm our results obtained on log data. We have shown that VA-TDI can accurately compare result lists while preserving vertical blocks. Accuracy is as high as under TDI, with only small losses in efficiency for small sample sizes.

5.2.3 Lack of Bias

Our final simulated experiment assesses the unbiasedness of VA-TDI under random clicks (RQ3). It is important that accounting for vertical documents does not introduce bias, as it may otherwise lead to wrong interpretations of interleaving results. VA-TDI was designed to be unbiased under many forms of noise; here we validate that our implementation does indeed fulfill this requirement.

Under the random click model RCM (Section 5.1.2), an unbiased interleaved comparison method should not systematically prefer either ranker, i.e., the rankers should tie in expectation. We measure this following the methodology proposed in [13], by counting the number of comparisons for which a method detects a significant preference towards one of the rankers. For an unbiased method, this number should be close to the number expected due to noise. For example, a significance test with a p-value of 0.05 should detect statistically significant differences between rankers under random clicks in 5% of the comparisons.

Table 5 shows the results: the percentage of detected significant differences for TDI and VA-TDI with dependent and independent block placement and for relevant and no relevant vertical documents at 100 to 500 impressions ($p = 0.05$). For both methods and all conditions we see that the number of significant differences detected is in line with the expected 5%. Using a one-tailed binomial confidence test (also with $p = 0.05$), we confirm that this number is only once significantly higher than 5%. The lowest number of significant differences is detected for VA-TDI under independent block placement after 400 impressions (13 differences or 2.6%), the highest value—and the only significantly higher value—is observed for TDI (38 differences or 7.6%), under dependent block placement and with relevant vertical documents, after 100 impressions. We also see that the number of detected significant differences does not increase with the number of impressions. This confirms that, like TDI, VA-TDI is unbiased under the RCM under all tested conditions.

To summarize, with our simulation experiments, we have shown that:

- VA-TDI does not degrade the user experience, as it does not break vertical blocks (Figure 4);
- VA-TDI can compare rankings, while preserving vertical blocks, as accurately as TDI (Figure 5); and
- VA-TDI, like TDI, is unbiased under random clicks (Table 5).

Based on these findings, we conclude that VA-TDI should be used instead of TDI for comparing two ranking systems in situations where there are vertical documents present.

6. DISCUSSION

We now discuss alternative design choices for vertical-aware interleaving and show how vertical-awareness can be integrated with interleaving algorithms other than TDI.

The design decisions for our VA-TDI method were made to preserve the user experience as much as possible. Two natural alternatives can be considered, but these impact the user experience in different ways. First, the interleaving algorithm could decompose the problem of block placement by first sampling the size and position of the vertical block, and then interleaving vertical and non-vertical documents separately (*Alternative 1*). Second, the algorithm could treat the vertical block as one pseudo-document during interleaving. Vertical documents would again be interleaved separately, and then inserted into the overall list at the place where the pseudo-document would naturally occur (*Alternative 2*).

A drawback of *Alternatives 1* and *2* is that they assume that the rankings of the vertical documents are independent of the placement of the vertical block. We may end up in a situation where we place a vertical block at a position advised by A and then start the block with a vertical document contributed by B which in turn may not be suitable for such a position. For example, if $A = [d_1^*, d_2, d_3, d_4]$, $B = [d_2, d_3, d_4, d_5^*]$ (where d_1^*, d_5^* are vertical documents) we may end up with the interleaved list $L = [d_5^*, d_2, d_3, d_4]$ which is a significant degradation of the user experience, assuming both A and B rankings are good and d_5^* should not be above d_2, d_3 and d_4. In contrast, as shown in Algorithm 1 VA-TDI makes sure that the first vertical document is contributed by the system that would contribute it in a regular TDI (e.g., the system that places the vertical block higher).

The OI approach by Radlinski and Craswell [18] specifies an interleaving algorithm as an optimization problem. OI starts from a set of constraints that a document list should fulfill, and formulates interleaving as an optimization problem that minimizes the number of samples required to make reliable comparisons while respecting the specified constraints. However, the most straightforward way of ensuring vertical document grouping has its issues. If we want both systems to contribute to the resulting vertical ranking we are forced to place the block lower in the interleaved list, because we first need to show the union of before-block documents from A and B. If we do so, we change the user experience and hence violate the requirement that interleaving should have a "Low Usability Impact," as formulated by Joachims. The same holds for the documents inside a vertical block. We cannot start adding non-vertical documents

from the ranking A until it yields all of its vertical documents. This may lead to a bigger vertical block.

To extend OI to account for vertical blocks, we need to relax its constraints on permissible ranking prefixes and allow the vertical block to begin anywhere between its placement in ranking A or B. The constraints need to be extended to account for maintaining one contiguous vertical block, and to determine the size of the vertical block. After formulating these constraints, OI can correctly interleave and compare result lists with vertical blocks. Due to space restrictions we leave a detailed formulation of this extension to future work.

7. CONCLUSION

In this paper we have proposed the first vertical-aware interleaved comparison method, VA-TDI. In contrast to previous interleaved comparison methods, VA-TDI is designed to account for the placement of vertical result lists as one contiguous block, thus preserving this important aspect of the user experience.

We validated this method in two sets of experiments, first using real-life click log data, and second using simulations. This combination enabled us to validate the method both in a specific realistic search setting, and in a broader simulation setup. Limitations of the log approach include that only one specific type of vertical could be tested. Future work should validate the approach in additional search settings, possibly including results with several vertical blocks from a variety of vertical search engines. The simulation approach is based on a state of the art federated click model, but as new insights are gained into users' click behavior with and without vertical results, the simulations should be further refined. Nevertheless, we found no qualitative differences between our experiments on log data and using the simulation setup. This suggests that the obtained results are reliable.

VA-TDI preserves the quality of the user experience. Our experiments on click log data showed that the user behavior (as captured by click metrics) on vertical-aware interleaved lists falls between that on the original rankings A and B. Our simulations confirmed that, in contrast to non vertical-aware interleaving, VA-TDI consistently produces one coherent block of vertical results. In addition, VA-TDI is able to reliably detect preferences in the quality of web-only, vertical-only, or overall result list quality. On click log data, we observed good correlation with commonly-used click metrics. In our simulations, we found that VA-TDI achieves the same accuracy as TDI, while preserving the quality of the user experience. Finally, our simulation experiments showed that VA-TDI preserves unbiasedness under random clicks. Our results confirm that VA-TDI opens up the way for applying interleaved comparison methods to search engine results with vertical or aggregated results, removing a major limitation of previous methods.

Acknowledgements. We thank Filip Radlinski for his detailed feedback on a draft of this paper. This research was partially supported by the European Community's Seventh Framework Program (FP7/2007-2013) under grant agreements nr 258191 (PROMISE Network of Excellence) and 288024 (LiMoSINe project), the Netherlands Organisation for Scientific Research (NWO) under project nrs 640.004.802, 727.011.005, 612.001.116, HOR-11-10, the Center for Creation, Content and Technology (CCCT), the QuaMerdes project funded by the CLARIN-nl program, the TROVe project funded by the CLARIAH program, the Dutch national program COMMIT, the ESF Research Network Program ELIAS, the Elite Network Shifts project funded by the Royal Dutch Academy of Sciences (KNAW), the Netherlands eScience Center under project number 027.012.105 and the Yahoo! Faculty Research and Engagement Program.

REFERENCES

[1] R. Agrawal, S. Gollapudi, A. Halverson, and S. Ieong. Diversifying search results. In *WSDM*. ACM, 2009.
[2] J. Arguello, F. Diaz, J. Callan, and B. Carterette. A methodology for evaluating aggregated search results. In *ECIR*. Springer, 2011.
[3] O. Chapelle, D. Metzler, Y. Zhang, and P. Grinspan. Expected reciprocal rank for graded relevance. In *CIKM*. ACM, 2009.
[4] O. Chapelle, T. Joachims, F. Radlinski, and Y. Yue. Large-scale validation and analysis of interleaved search evaluation. *ACM TOIS*, 2012.
[5] D. Chen, W. Chen, and H. Wang. Beyond ten blue links: enabling user click modeling in federated web search. In *WSDM*. ACM, 2012.
[6] A. Chuklin, P. Serdyukov, and M. de Rijke. Using intent information to model user behavior in diversified search. In *ECIR*, 2013.
[7] C. Clarke, M. Kolla, and G. Cormack. Novelty and diversity in information retrieval evaluation. In *SIGIR*. ACM, 2008.
[8] C. W. Cleverdon, J. Mills, and M. Keen. Factors determining the performance of indexing systems. Techn. report, ASLIB Cranfield project, 1966.
[9] S. Dumais, E. Cutrell, and H. Chen. Optimizing search by showing results in context. In *CHI*, 2001.
[10] J. He, C. Zhai, and X. Li. Evaluation of methods for relative comparison of retrieval systems based on clickthroughs. In *CIKM*. ACM, 2009.
[11] K. Hofmann, S. Whiteson, and M. de Rijke. A probabilistic method for inferring preferences from clicks. In *CIKM*. ACM, 2011.
[12] K. Hofmann, S. Whiteson, and M. Rijke. Balancing exploration and exploitation in listwise and pairwise online learning to rank for information retrieval. *Information Retrieval*, 16(1), Apr. 2012.
[13] K. Hofmann, S. Whiteson, and M. de Rijke. Fidelity, soundness, and efficiency of interleaved comparison methods. *ACM Trans. Inf. Syst.*, 31(4), Oct. 2013.
[14] T. Joachims. Optimizing search engines using clickthrough data. In *KDD*. ACM, 2002.
[15] T. Joachims. Evaluating retrieval performance using clickthrough data. *Text Mining*, 2003.
[16] T. Joachims, L. Granka, B. Pan, H. Hembrooke, and G. Gay. Accurately interpreting clickthrough data as implicit feedback. In *SIGIR*. ACM, 2005.
[17] A. K. Ponnuswami, K. Pattabiraman, Q. Wu, R. Gilad-Bachrach, and T. Kanungo. On composition of a federated web search result page: using online users to provide pairwise preference for heterogeneous verticals. In *WSDM*. ACM, 2011.
[18] F. Radlinski and N. Craswell. Optimized interleaving for online retrieval evaluation. In *WSDM*, 2013.
[19] F. Radlinski, M. Kurup, and T. Joachims. How does clickthrough data reflect retrieval quality? In *CIKM*. ACM, 2008.
[20] A. Schuth, K. Hofmann, S. Whiteson, and M. de Rijke. Lerot: an Online Learning to Rank Framework. In *Living Labs workshop at CIKM*. ACM, 2013.
[21] J. Seo, W. B. Croft, K. H. Kim, and J. H. Lee. Smoothing click counts for aggregated vertical search. *Advances in Information Retrieval*, 2011.
[22] K. Zhou, R. Cummins, M. Lalmas, and J. M. Jose. Evaluating Aggregated Search Pages. In *SIGIR*, 2012.